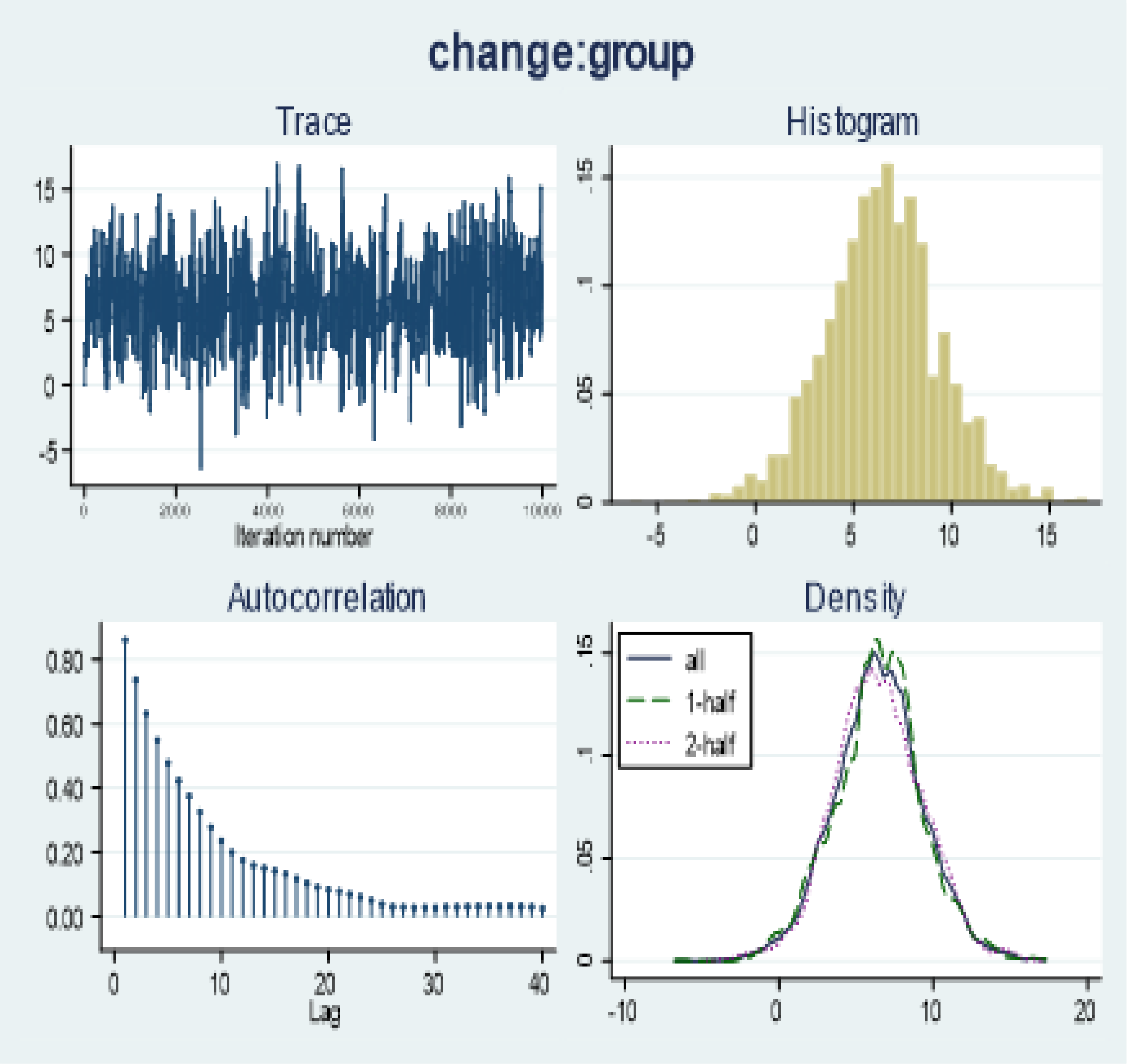
First draft, please do not quote.

Applied Econometric Analysis

Feridoon Koohi-Kamali



PREFACE

This text contains the first draft of my lecture notes on applied econometrics most of which were prepared for my Advanced Econometrics classes at the New School for Social Research during. The emphasis of the text is on the needs of general gradute students and policy analyists wishing to employ econometrics for research; those who wish to delve more deeply into the econometric theories of the topics examined should consult some of the works cited in the text. For that reason, the covegae of the text broad, though I cannot claim the text examines all key areas of applied econometrics. Still, it does cover a wide range: microeconometrics, including duration models; short and long panel data analysis, including dynamic and heterogenous panels, and macroeconometric time-series to which I have added a new chapter on volatility analysis. The text is divided equally into two sections that broadly follow my classes of advanced econometrics I and II at New School, though I have added a section on Machine Learning econometrics. However, I have allocated longer treatments and more exercises to three areas: models of duration analysis in view of their relevance to Covid-19 related pandemic research in health and economics, and Bayesian econometrics, as the approach requires background in non-simulation, prior to simulating applications. Moreover, with fast expanding use of Machine Learning models in econometrics, more space is also allocated to chapters 19 and 20 on Machine Learning models. I have prepared a separate supplementary text of solutions manual for end-of-chapter exercises of the full text to non-empirical and computer-based questions. All the empirical exercises are with Stata except in chapters 19 and 20 on that have a mixture of empirical exercises in both Stata and R to take advantages of wider avaiablility of Machine Learning empirical examples in R. A folder of all data files used in every chapter is also available. Reference list contains each key chapter references rathe than the extensive sources cited in the test.

Finally, I would like to express my appreciation for the helpful comments I received from Duncan Foley on the Bayesian portions of this text, and to thank the students at New School who attended my classes; their curiosity, questions and discussions have brought more clarity and accuracy to the text, though undoubtedly there is much room for improvement.

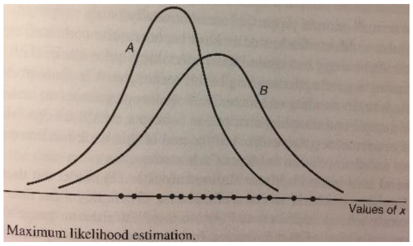
Feridoon Koohi, August 2022

Chapter 1 Maximum Likelihood, Other Nonlinear & Bayesian Estimation

Introduction

This chapter briefly discusses some basic features of nonlinear estimation. Nonlinear function of the dependent variable can assume different forms; nonlinearity can consist of parameter nonlinearity, or result from censoring/truncation data, or from the loss function even if the conditional mean is linear in parameters. The two leading estimators we employ in all such cases are the maximum likelihood (***ML***) estimator and the non-linear (***NL***) estimator.

The maximum likelihood is the most efficient estimator among the class of consistent asymptotically normal estimators. The Likelihood Principle chooses as estimator θ for the true parameter vector $\theta_0$that maximizes the likelihood of observing the actual sample; the likelihood of the probability density for the continuous case, and the probability of mass function for the discrete case. For example, if there are two values of θ for the probability of the observed data occurring of 0.005 and 0.008, on the likelihood principle, the second θ is preferred, or B density over A shown below.



Maximum likelihood estimation.

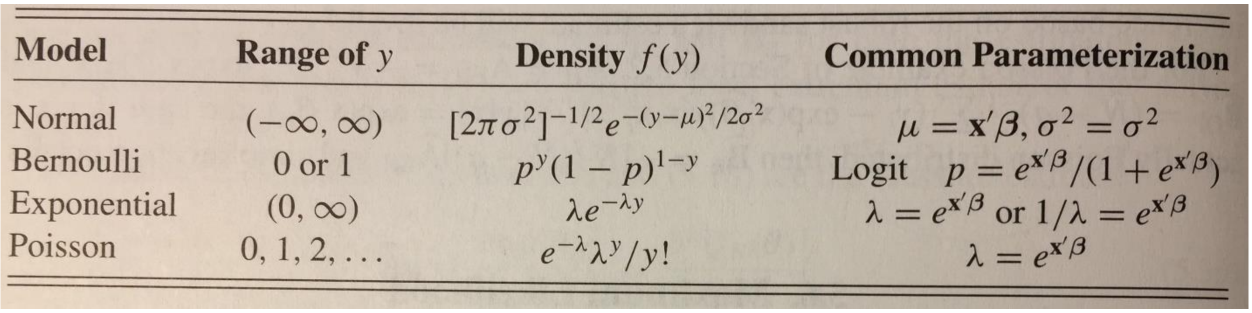
The **likelihood function** is denoted as $L_N = (\theta|y, X)$ based on the sample $[(y_i, X_i)$, (i=1, 2, . . . , *N)*]; given independent observations, it is the joint function defined by the product of the individual densities $\prod_i f(y|x_i, \theta)$, equivalently defined by the **log-likelihood function** as the sum of logs of this product $\sum_i lnf(y|x_i, \theta)$. Hence, maximizing $\mathbb{L}_N(\theta)$ is equivalent to maximizing $\mathbb{L}_N(\theta)$= $ln\, L_N(\theta)$. For cross-sectional data, the observation (y_i, X_i) are independent over *i*, resulting to the log-likelihood function

$$N^{-1}\mathbb{L}_N(\theta) = \frac{1}{N}\sum_{i\,=\,1}^{N} lnf(y|x_i, \theta) \qquad (1.1)$$

where division by N makes the function an average one.

Some commonly employed distributions and parametric specifications are shown in table 1.1. For *continuous data* on (-∞, ∞), the normal is the standard distribution; the classical regression model with μ=x$'y$ and constant σ^2. For *discrete binary data* with (0, 1) values, is the Bernoulli, a special case of binomial density. The parameterization of the Bernoulli density leads to logit and probit models with *p*=*Φ*(x$'y$) with Φ(.) the standard normal *cdf*. For *positive continuous data* on (0, ∞), esp. duration data, the exponential density is common, though the more flexible Weibull, gamma, and log-normal are often employed too. For *integer-valued count data* taking values 1, 2, …., the Poisson and negative binomial ae the usual densities setting λ=exp(x$'y$) to ensure a positive conditional mean. For *incomplete observed data*, censored or truncated distributions are used, the most common being the censored normal that leads to the Tobin model. The standard likelihood-based models are specified in terms of the distribution of the dependent variable rather than that of the error term.

Table 1.1-Commonly Used LM Densities



Model	Range of y	Density $f(y)$	Common Parameterization
Normal	$(-\infty, \infty)$	$[2\pi\sigma^2]^{-1/2}e^{-(y-\mu)^2/2\sigma^2}$	$\mu = \mathbf{x}'\beta, \sigma^2 = \sigma^2$
Bernoulli	0 or 1	$p^y(1-p)^{1-y}$	Logit $p = e^{\mathbf{x}'\beta}/(1+e^{\mathbf{x}'\beta})$
Exponential	$(0, \infty)$	$\lambda e^{-\lambda y}$	$\lambda = e^{\mathbf{x}'\beta}$ or $1/\lambda = e^{\mathbf{x}'\beta}$
Poisson	$0, 1, 2, \ldots$	$e^{-\lambda}\lambda^y/y!$	$\lambda = e^{\mathbf{x}'\beta}$

1.1 *Maximum Likelihood Estimator*

The MLE maximizes the log-likelihood function for y conditional **x** that solves the first-order conditions

$$\frac{1}{N}\frac{\partial \mathbb{L}_N(\theta)}{\partial \theta} = \frac{1}{N}\frac{\partial \ln f(y_i|x_i,\theta)}{\partial \theta} = 0 \qquad (1.1.1)$$

The gradient vector $\frac{\partial \mathbb{L}_N(\theta)}{\partial \theta}$ is called the **score vector**, and when evaluated at the *true parameter values* θ_0, it is called the **efficient score**. The **Regularity Conditions** are

$$E_f \frac{\partial \ln f(y|x,\theta)}{\partial \theta} = \int \frac{\partial \ln f(y|x,\theta)}{\partial \theta} f(y|x,\theta) = 0 \qquad (1.1.2)$$

$$-E_f \frac{\partial^2 \ln f(y|x,\theta)}{\partial\theta\partial\theta'} = E_f \frac{\partial \ln f(y|x,\theta)}{\partial\theta} \frac{\partial \ln f(y|x,\theta)}{\partial\theta'} f(y|x,\theta) \qquad (1.1.3)$$

The expectation of the score vector by (1.1.2) is equal to zero; note that we take the expectation, $E_f(.)$, with respect to $f(y|x,\theta)$density. The expectation of the **outer product of the score vector** is the matrix

$$\mathfrak{I} = E[\frac{\partial\mathbb{L}_N(\theta)}{\partial\theta}\frac{\partial\mathbb{L}_N(\theta)}{\partial\theta'}] \qquad (1.1.4)$$

The variance of the score vector $\frac{\partial\mathbb{L}_N(\theta)}{\partial\theta}$ is (1.1.4) since by (1.1.2), the vector has zero expectation. Large values of $\mathfrak{I}$ mean that small changes in θ result in large changes in the log-likelihood, hence the function contains a great deal of information about θ. Defined by (1.1.4), $\mathfrak{I}$is called **Fisher information matrix.** The regularity condition (1.1.3) for the log-likelihood function (1.1.1) implies

$$-E_f\left[\frac{\partial^2\mathbb{L}_N(\theta)}{\partial\theta\partial\theta'}|_{\theta_0}\right] = -E_f\left[\frac{\partial\mathbb{L}_N(\theta)}{\partial\theta}\frac{\partial\mathbb{L}_N(\theta)}{\partial\theta'}|_{\theta_0}\right] \qquad (1.1.5)^1$$

Provided the expectation is with respect to θ_0. (1.1.5) is called the **information matrix equality** and implies that the right-hand side of the equation is also equal to the information matrix.

Distribution of the LME

Consistency requires $E[\frac{\partial \ln f(y|x,\theta)}{\partial\theta}|_{\theta_0}] = 0$; this condition is satisfied by (1.1.2) as long as the expectation is taken with respect to $f(y|\mathbf{x},\theta_0)$, that is *if the dgp is correctly specified* for $f(y|\mathbf{x},\theta_0)$, then the MLE is consistent for θ_0. Consistency of ML distribution is based on the following assumptions:

(i) The *dgp* is the conditional density $f(y|\mathbf{x},\theta_0)$ defining the likelihood function (correct specification)

(ii) The density function satisfies $f(y|\mathbf{x},\theta^{(1)} = \quad f(y|\mathbf{x},\theta^{(2)}\ iff \theta^{(1)} = \theta^{(2)}$ (for uniqueness)

[1] The outer product on LHS of (1.1.3) is equal to the (negative) inner product on its RLS, this is a scalar equal to the trace of the outer product matrix. The scalar is the product of the Euclidian 1-*by*-1 vectors generalized to the product of 1-*by*-*m* & *m*-*by*-1 vectors where one vector has components in the opposite direction of the other, for details, see Pesaran (2015, section 9.2), or Wooldridge (2010, section 13.5).

(iii) $A_0 = \left[plim\ \frac{1}{N}\frac{\partial^2\mathbb{L}_N(\theta)}{\partial\theta\partial\theta'}|_{\theta_0}\right]$ exists and is finite nonsingular (for the matrix of variance).

(iv) The order of differentiation and integration of the log-likelihood can be reversed (to ensue regulatory condition (1.2) for the expectation of the score vector.)[2]

Proposition on Distribution of *ML* Estimator: based on (i)-(iv), the ***ML* estimator $\boldsymbol{\hat{\theta}_{ML}}$,** defined to be the solution of the first-order conditions $\frac{1}{N}\frac{\partial\mathbb{L}_N(\theta)}{\partial\theta}$, is consistent for the true parameter value θ_0 and $\sqrt{N}\left(\hat{\theta}_{ML} - \theta_0\right) \rightarrow^d \mathcal{N}[0, -A_0]$[3].

The proposition results in the *asymptotic distribution* of the MLE given as

$$\hat{\theta}_{ML} \sim^a [\theta, (E[\frac{\partial^2\mathbb{L}_N(\theta)}{\partial\theta\partial\theta'}])^{-\mathbf{1}}] \qquad (1.1.6)$$

(1.1.6) is evaluated at θ_0, and we assume LLN applies to replace the plim operator by lime. The right-hand side of (1.1.6) is the Cramer-Roa lower bound (CRLB) of the variance matrix of consistent asymptotically normal estimators with convergence to normality of $\sqrt{N}\left(\hat{\theta} - \theta_0\right)$ uniform in the compact intervals θ_0; here replacing the basic lower bound CRLB of the variance of unbiased estimators in small samples.

Example: Poisson Regression

The Poisson distribution is suitable for a dependent variable that takes only nonnegative integer values 0, 1, 2, …; employed to model the number of occurrences of an event, e.g. number of doctor visits per year. The Poisson probability mass function, discussed in more detailed in chapter 3, is

$$\mathrm{F(y|x)}=e^{-\lambda}\lambda^y/y!$$

where $y^!$ stands for the factorial of y, the function is specified as λ=exp($x'\beta$) with E[y]=λ and Var[y]=λ, hence resulting in the density of the Poisson regression for a single observation

$$f\,(y|\mathbf{x},\, \beta) = e^{-\exp\,(x'\beta)}\exp\,(x'\beta)^y/y!$$

[2] As an example, the expectation and the exponential function cannot be interchanged, see Q-1 exercise.

[3] Multiplication by $\sqrt{N}$ re-scales $\hat{\theta}$ to obtain a random variable that has finite, nondegenerate distribution as $N \rightarrow \infty$.

The maximum likelihood of this function is the joint density over the sample observations $\prod_i f(y|x_i,\theta)$, equivalently defined by the log-likelihood function as the sum of logs of this product $\sum_i lnf(y|x_i,\theta)$. The log-density for the *i*th observation is

$$ln\, f(y_i|\mathbf{x}_i, \beta) =- \exp(x_i'\beta) + y_i x_i'\beta - ln\, y^!$$

Hence, the Poisson MLE estimator $\hat{\beta}$maximizes

$$Q_N(\beta) = \sqrt{N}\sum_{i=1}^{N}\{-\exp(x_i'\beta) + y_i x_i'\beta - ln\, y^!\}$$

Where the scale factor $\sqrt{N}$ is included to ensure$Q_N(\beta)$ remains finite as $N \to \infty$; the estimator solves the first-order conditions for $\frac{\partial Q_N(\beta)}{\partial \boldsymbol{\beta}|_{\hat{\boldsymbol{\beta}}}} = 0$, or

$$\sqrt{N}\sum_{i=1}^{N}(y_i - \exp(x_i'\beta))\,\mathbf{x}_i|_{\hat{\beta}} = 0$$

This is a nonlinear equation that must be solved by numerical iterative methods since it has no analytical solution.

1.2 *Quasi-Maximum Likelihood*

The **Quasi-*MLE*$\hat{\boldsymbol{\theta}}_{QML}$** maximizes a log-likelihood function with a mis-specified density that usually leads to inconsistent estimation. The density mis-specification results in inconsistency because the expectation is no longer evaluated with respect to the correct $f(y|\mathbf{x},\theta_0)$. However, the Quasi-*MLE*$\hat{\theta}_{QML}$ converges in probability to the pseudo-true value θ^*. If $E[y|x] \neq \boldsymbol{x}'\beta_0$, the OLS can still be unbiased, the *QMLE*has a similar interpretation. Let the joint density of $y_1, , \ldots, y_N$ be $f(y\,|\,\theta)$ and the unknown true density as $f(y)$ where dependence on regressors are left out to simplify notations. Then the **Kullback-Leibler information criterion** (KLIC) is defined as

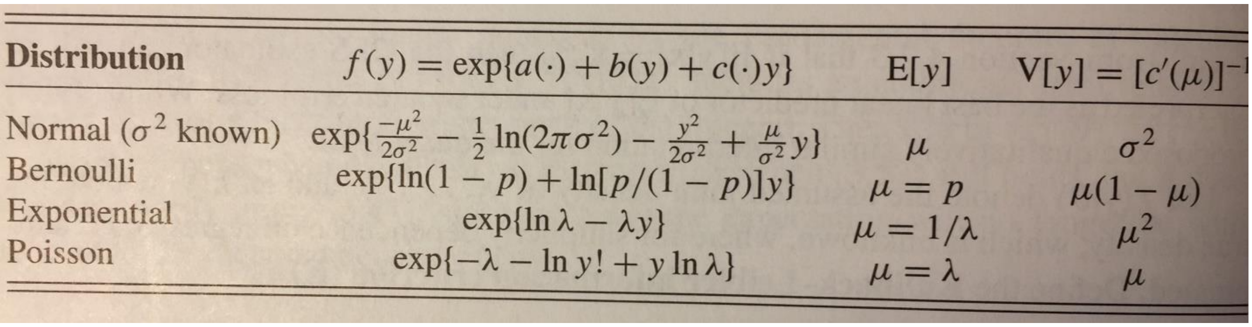
$$\text{KLIC}=E[\,\ln\,(\frac{f\,(y)}{f\,(y\,|\theta)})]$$

where the expectation is with respect to $f(y)$. LLIC has a minimum value of zero when $f\,(y)= f(y\,|\theta_0)$ when the density is correctly specified, and values > 0 indicate greater departure from the true density. Then the QLME minimizes the gap measured by KLIC between $f(y\,|\,\theta)$ & $f(y)$. In some special cases the QMLE is consistent with a partially mis-specified density; an example is the linear regression model with normality even if the errors are nonnormal as long as $E(y|\mathbf{x})=x'\beta_0$. Similarly, the **linear exponential family** (***LEF***)

$$f(y\,|\,\mu)=\exp[\alpha(\mu)+\mathrm{b}(y)+c(\mu)y]$$

where b is a normalizing constant to ensure the probability sum up or inyegrate to 1, and$[\alpha(\mu)+c(\mu)y]$ is linear in y. The *LEF* is robust to mis-specification; hence, *QMLE* with a *LEF* is consistent as long as the mean of y, conditional on x, is correctly specified, it is not necessary for the true *dgp* for y be *LEF*. The models based on the *LEF* are called **generalized linear models** (***GLM***s); this class includes nonlinear least squares, Poisson, geometric, probit, logit, binomial, gamma, and exponential regression models. Table 1.2 shows common examples of *LEF*.

Table 1.2-Common Examples of LEF



Distribution	$f(y) = \exp\{a(\cdot) + b(y) + c(\cdot)y\}$	$\mathrm{E}[y]$	$\mathrm{V}[y] = [c'(\mu)]^{-1}$
Normal (σ^2 known)	$\exp\{\frac{-\mu^2}{2\sigma^2} - \frac{1}{2}\ln(2\pi\sigma^2) - \frac{y^2}{2\sigma^2} + \frac{\mu}{\sigma^2}y\}$	μ	σ^2
Bernoulli	$\exp\{\ln(1-p) + \ln[p/(1-p)]y\}$	$\mu = p$	$\mu(1-\mu)$
Exponential	$\exp\{\ln\lambda - \lambda y\}$	$\mu = 1/\lambda$	μ^2
Poisson	$\exp\{-\lambda - \ln y! + y\ln\lambda\}$	$\mu = \lambda$	μ

1.3 *Nonlinear Least Squares (NLS)*

The ***NLS*** estimator E[y|**x**]=g(**x,** β) where g(.) is nonlinear in β; the linear least squares is a special case of the NLS with $\mathrm{g}(\mathbf{x}, \beta) = \mathbf{x}'\beta$. The typical reason for a nonlinear specification is to allow E[y|**x**] to include a range of restriction, for example E[y|**x**]>0 . The ***NLS*** applied to heteroskedastic models are less efficient than the ***MLE*** but extensively employed because they rely on weaker distributional assumptions. The ***NLS*** estimator minimizes the sum of squared errors

$$Q_N(\beta) = \sqrt{2N}\sum_{i=1}^{N}(y_i - g(x_i'\beta))^2$$

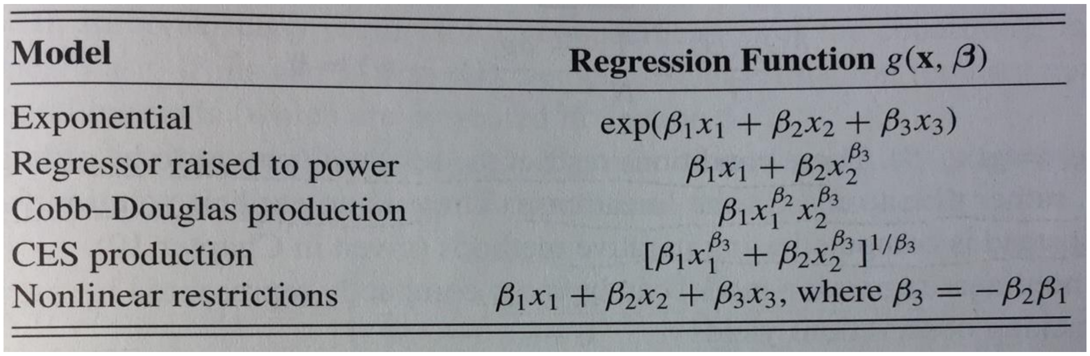
where the scale factor ½ intended to simplify the analysis. Differentiation for the ***NLS*** first-order conditions

$$\frac{\partial Q_N(\beta)}{\partial \beta} = \sqrt{N}\sum\nolimits_{i=1}^{N}\frac{\partial g_i}{\partial \beta}(y_i - g_i) = 0$$

These conditions restrict the (*y* – *g*) to be orthogonal to $\frac{\partial g}{\partial \beta}$ rather than to **x**, as in the linear case. Once again, there is no analytical solution for the ***NLS*** minimization; the iterative estimation methods are necessary. Table 1.3 shows the most common examples of NLE.

The *ML* and *NLS* are two leading examples of a general class of ***m*-estimators** that maximize an objective function defined over sum or average of *N* different subfunctions.

Table 1.3-common examples of NLE



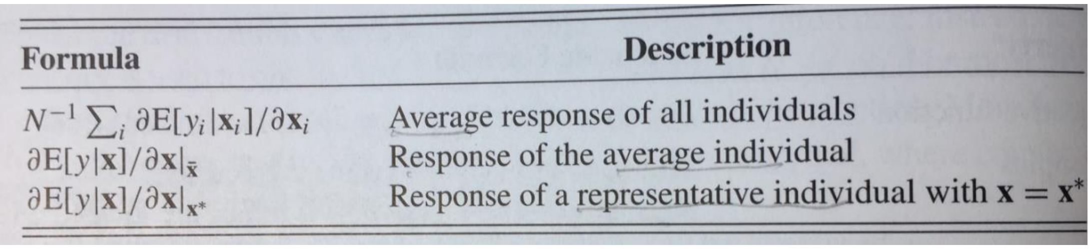
Model	**Regression Function** $g(\mathbf{x}, \boldsymbol{\beta})$
Exponential	$\exp(\beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3)$
Regressor raised to power	$\beta_1 x_1 + \beta_2 x_2^{\beta_3}$
Cobb–Douglas production	$\beta_1 x_1^{\beta_2} x_2^{\beta_3}$
CES production	$[\beta_1 x_1^{\beta_3} + \beta_2 x_2^{\beta_3}]^{1/\beta_3}$
Nonlinear restrictions	$\beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3$, where $\beta_3 = -\beta_2\beta_1$

1.4 *Marginal Effects in Nonlinear Regression*

We are often interested estimating the **marginal effect** of a unit change in a regressor $\mathbf{x_i}$ on the conditional mean of *y*. However, under nonlinearity this interpretation is not valid. For example, if $E[y|\text{x}]=\exp(\text{x}'\beta)$, then $\partial E[y|\text{x}]/\partial\text{x}=\exp(\text{x}'\beta)\beta$ which is a function of both parameters and regressors; the size of the marginal effects depend on β but also on changing **x**, hence vary with the evaluation value of **x**. In general, the marginal effects vary with the evaluation value of **x**.

There are three common measures of marginal effects; these measures are all equal in the linear case but they differ in nonlinear models, as shown in table 1.4. The first estimates the average the marginal effects for all individuals, the second evaluates the marginal effect at $\mathbf{x}=\bar{\mathbf{x}}$ for a representative individual, and the third evaluates the marginal effects for specific characteristics $\mathbf{x}=\mathbf{x}^*$, the marginal effect for a female with a college degree.

Table 1.4-Three Different Estimates of Marginal Effects



Formula	Description
$N^{-1}\sum_i \partial \mathrm{E}[y_i\|\mathbf{x}_i]/\partial \mathbf{x}_i$	Average response of all individuals
$\partial \mathrm{E}[y\|\mathbf{x}]/\partial \mathbf{x}\|_{\bar{\mathbf{x}}}$	Response of the average individual
$\partial \mathrm{E}[y\|\mathbf{x}]/\partial \mathbf{x}\|_{\mathbf{x}^*}$	Response of a representative individual with $\mathbf{x} = \mathbf{x}^*$

Direct interpretation of coefficient estimates is possible by using **single-index models** based on the following specification

$$E[y|\mathrm{x}]=\mathrm{g}(\mathrm{x}'\beta)$$

That is, the data and parameters enter the nonlinear mean function g(.) through the single index $\mathbf{x}'\beta$. Then, the mean is a nonlinear function of a *linear combination of the regressors and parameters*. This is the case of *mild nonlinearity* for which the marginal effects use the **calculous methods**, thus:

$$\partial E[y|\mathrm{x}]/\partial x_i = g'(\mathrm{x}'\beta)\beta_i$$

where $\mathrm{g}'(z) = \partial g(z)/\partial z$. Hence, the **relative effects** are given by

$$\frac{\partial E[y|\mathbf{x}]/\,\partial x_j}{\partial E[y|\mathbf{x}]/\,\partial x_k} = \frac{\beta_i}{\beta_k}$$

because the common component $g'(\mathbf{x}'\beta)$cancel out. This method averages the marginal effects of all individuals., and it tends to change relatively little across different functional form $g(.)$. Many standard nonlinear models such as logit, probit, and Tobit are of this single form.

An alternative, the **Finite-Difference Method** measures the marginal effects based on a comparison of the conditional mean when x_i is increased by one unit with the value before the change.

$$\frac{\Delta E[y|\mathbf{x}]}{\Delta \mathbf{x}_i} = \mathrm{g}(\mathrm{x} + e_j, \beta) - g(\mathbf{x}, \beta)$$

Where e_j is a vector with$e_i = 1$ for ith entry and zero for other entries. For the linear case, the calculus and finite-difference methods lead to identical results, but with nonlinear models, the methods produce different marginal effects except for a minute change in $\mathbf{x}_i$.

Calculus methods are often used for continuous regressors, while finite-difference methods are used for integer-valued regressors, for example the indicator (0, 1) variable.

Example: an exponential conditional mean yields $\partial E[y|\mathbf{x}]/\partial \boldsymbol{x_j} = E[y|\mathbf{x}].\beta_j$; a unit change in $x_i$has a *semi-elasticity* interpretation because the change in x_i results in the multiple of β_i, that is, if β_i=0.2 then a unit change leads to 0.2 times that amount, a 20% additional change by single-index/calculus method. On the other hand, the finite-difference method computes the marginal from

$$\partial E[y|\mathbf{x}]/\partial \boldsymbol{x_j} = E[y|\mathbf{x}].(e^{\beta_j} - 1)$$

Thus, except when β_i is very small, then there will be a difference, for example β_i =0.2, e^{β_i}=1.22, an increase of 22%.

1.5 *OLS & MLE Asymptotic Expectation and Asymptotic variance*

An estimator must have two desirable features. It should be asymptotically consistent and should be able to generate an asymptotic distribution for conducting statistical inference. Consistency is about what would happen to the moments of a distribution if the sample size becomes increasingly large while we also make numerous random resamples for each size. Since we work only with a fixed sample size, consistency is a thought experiment with a commonsense appeal: if you cannot estimate a good approximation for a parameter of interest by increasing the sample size, then you must have a defective estimator. Here we briefly examine the asymptotic estimation and distribution of the classical approach for the least squares and maximum likelihood models to emphasize the common principle around which both estimators are organized.

First, assume we have a unique true parameter value θ that generates the data. This condition requires the correct specification of the data generating process (*dgp*), and its unique representation, and it is estimated by $\hat{\theta}$. Even in a very large sample, θ and $\hat{\theta}$ will not be exactly equal because of the randomness of a sample but instead, we require $\hat{\theta}$ to *converge in probability* to θ, that is, $\hat{\theta} \rightarrow^{p} \theta$. Next, as N→∞, the distribution of $\hat{\theta}$ degenerates with all mass at θ; to prevent that result, we re-scale the estimator $\hat{\theta}$ by $\sqrt{N}$ to obtain a nondegenerative distribution and examine the behavior of $\sqrt{N}(\hat{\theta}$ - θ) as N→∞. For many estimators, $\sqrt{N}(\hat{\theta}$ - θ) *converges in distribution* to the multivariate normal, leading to the *limit distribution* of the estimator $\hat{\theta}$. The OLS case expresses this result by

$$\sqrt{N}(\hat{\theta} - \theta) = \left(\frac{1}{N}\sum_{i=1}^{N} x_i x_i'\right)^{-1} \frac{1}{\sqrt{N}}\sum_{i=1}^{N} x_i u_i$$

where we can define non-singular matrix *A*, such that $\left(\frac{1}{N}\sum_{i=1}^{N} x_i x_i'\right)^{-1} \rightarrow^{p} A^{-1}$. By the CLT, the second part involving $x_i u_i$ on the RHS has an asymptotic normal distribution with mean zero and variance-covariance matrix *B*. Then, $\sqrt{N}(\hat{\theta}$ - θ) has an asymptotic multivariate normal distribution with mean zero and variance-covariance matrix $A^{-1}BA^{-1}$. To obtain this result, we note that while the sample variance is not an unbiased estimate of the population σ^2, because $E(\sqrt{S_n}) \neq \sqrt{E(S_n)}$, by the *LLN*, $plimS_n^2 = \sqrt{plimS_n^2} = \sqrt{\sigma^2} = \sigma$, therefore $S_n$is consistent for σ. Using that result, it can be shown that B=$\sigma^2 A$, leading to

$$\sqrt{N}(\hat{\theta} - \theta) \sim^{a} N(0, \sigma^2 A^{-1}).$$

A limit distribution for an *m*-estimator can be similarly obtained by a first-order Taylor expansion approximation that leads to:

$$\sqrt{N}(\hat{\theta} - \theta) \sim^{a} N(0, A^{-1}BA^{-1})$$

where A=$plim\ N^{-1}\sum_{i=1}^{N} \partial^2 q_i(\theta)/\partial\theta\partial\theta'|_{\theta}$ and B= $plim\ N^{-1}\sum_{i=1}^{N} \partial q_i(\theta)/\partial\theta\partial\theta'|_{\theta}$.

We can then obtain the distribution of $\hat{\theta}$ by the division of the LHS of the above by $\sqrt{N}$ to have

$$V|\hat{\theta}| = N^{-1}(0, A^{-1}BA^{-1})$$

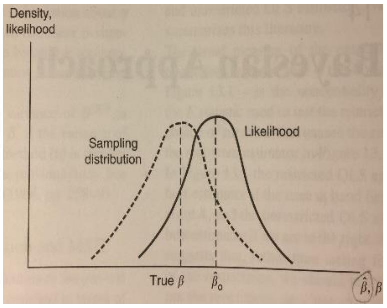
This relationship depends on the unknown true parameter θ, and is computed by the *estimated asymptotic variance* using consistent $\hat{A}$ & $\hat{B}$ of A & B.

$$\hat{V}|\hat{\theta}| = N^{-1}\hat{A}^{-1}\hat{B}\hat{A}^{-1}$$

1.6 *Basics of Bayesian Methods*

So far, we have presented the classical view of econometrics. We discussed how the classical approach employs the data to produce a "best" point estimate $\hat{\beta}$ for the true but unknown parameter β linearly by minimizing the squared sum of errors by the least squares estimator, or non-linearly maximizing the likelihood function by the maximum likelihood estimator. The parameter estimate, $\hat{\beta}$, is shown to be the outcome of each random sample drawing repeated a large number of times, the classical model's organizing principle, by appealing to the LLN and CLT, that has good properties such as unbiasedness. This view is summed up in Figure 1.2.

Figure 1.2 The Classical view of sampling Distribution



The alternative Bayesian approach we examine now has a sharply different organizing principle based on subjective probability, and the range of its applicability is increasing, as the advances in computer power have reduced the practical difficulties in its implementations. The Bayesian approach starts from the calculation of odds taken on the true value β to produce, not a point estimate, but the probability of event B occurring given that event A has occurred; this conditional probability is called the *posterior* density function. The odds reflect the *subjective probability* of the researcher or a "prior" density before seeing the data regarding the range of values that the true β can assume. The de Finetti *coherence principle* formalized the concept of subjective probability, according to which the individual should never assign probability odds to events that enables someone else to choose stakes that are a sure loss for the individual, regardless of the final outcome. This simple argument leads to the rule for conditional probability: $P(A\&B)= P(A)P(B|A)=P(B)P(A|B)$; from which follows the simplest form of Bayes' theorem

$$P(A\,|B)= \frac{P(B)P(A|B)}{P(A)}$$

That means if we observe the occurrence of the event A, then the probability of B occurring given that A has occurred is the posterior probability $P(A\,\&B)$; see section 17. 1 for the general form of Bayes' theorem. Being a distribution, the posterior is not restricted to a particular point estimate but can also produce other percentiles of interest; the choice of a point estimate of β as the mean of the posterior density depends on the loss function employed, the most common is the posterior mean obtained from a quadratic function proportional to the squared difference of $(\beta\ \text{-}\hat{\beta})^2$ as explained in figure 1.2.

Figure. 1.3 the Bayesian expected loss using $\hat{\beta}_2$

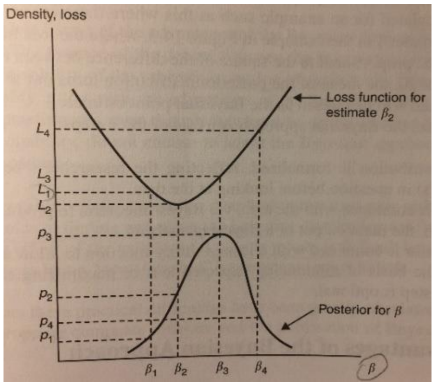


Figure 1.3 shows the loss involved in a specific point estimate β_0^* from the posterior distribution for every possible true value of β; the expected loss is then obtained from the expectation over all possible values of β; *not* from repeated sampling, as in the classical approach. Suppose β is estimated by β_2; different losses result from different unknown true values of β by being estimated with different point estimates; four such point estimates, i=1, 2, 3, 4, are shown in figure 1.2, each with probability p_i and the loss L_i. The expected loss from employing $\beta_2$is given as the weighted average of all L_i. *This is only the expected loss from a single estimate, β_2, and must be repeated infinitely for all possible values of the true β at different β_i points to determine the expected losses from employing each alternative point estimate β_i.* The height of the posterior function corresponding to each p_i gives the probability of each β_i being the true value of β. When all such losses are calculated, then the Bayesian point estimate is chosen from that loss function whose expected loss is the smallest; in practice the expected values are not all calculated but rather obtained by algebra. For figure 1.3, the mean of the posterior distribution results in minimum

expected loss selected as the Bayesian point estimate. Thus, the posterior function is combined with a loss function to produce a point estimate based on minimized expected loss. A Bayesian can employ uncontroversial non-informative prior reflecting complete ignorance, so the outcome is the result of the data alone. Unfortunately, this leads to estimation results identical to the classical answers, though with different interpretation. A complete ignorance prior would have the same shape and spread as the sampling distribution that is located over $\hat{\beta}$ in contrast to the sample distribution located over the true, unknown value of β. This critical difference is also emphasized when figures 1.2 and 1.3 are compared with respect to the horizontal axis: the classical sampling distribution has$\hat{\beta}$ on that axis while the Bayesian horizontal axis is drawn with respect to the values of β to highlight the fundamental difference on point estimation between the two approaches!

The conditional probability $P(A\,|B)$ is an *ex ante* belief on an event not yet occurred and captures well the Bayesian notion that probability is a feature of the individual's view of reality and has the paradoxical implication that econometric analysis should start with the data conditioned on beliefs. This requires relating Bayesian probability to observable quantities. This task is accomplished by de Finetti's fundamental concept of *exchangeability*, defined as: a finite sequence events (random variable), t=1, 2, . . . , T, is exchangeable *iff* the joint probability of the sequence is invariant under permutation of the subscripts:

$$\mathrm{P}(y_1, y_2, \ldots, y_T)=\mathrm{P}(y_{\pi 1}, y_{\pi 2}, \ldots, y_{\pi T})$$

where $\pi(t)$ is a permutation of the elements in the sequence). Further, an infinite sequence is exchangeable *iff* any finite subsequence is exchangeable. Exchangeability implies a sequence of alike random quantities can be operationalized by probabilities imputed to *observable* quantities. The de Finetti representation theorem provides the necessary correspondence between the parameters of subjective probability and the solely observable features of exchangeable sequences. For example, a sequence of Bernoulli trials is exchangeable *iff* the probability of a particular sequence is independent of the order of success (s) and failure (F); then the sequences are given the *same* probability. More specifically, if all we know about a coin is that it was tossed three times and two heads appeared, then exchangeability requires that we assign *equal* probability to all three sequences *HHT*, *HTH*, and *THH*.

Formally, define the average number of occurrences as

$$\bar{Y}_T = \frac{1}{T}\sum_{t=1}^{T} Y_t$$

and let a probability mass function (*pmp*) $h(y_1, y_2, y_3, \ldots) = Pr\ (Y_1=y_1, Y_2=y_2, \ldots, y_T=y_T)$ represent the exchangeable beliefs fo a along sequence Y_t (t=1, 2, 3, . . . , T) with its corresponding cumulative distribution finction (*cdf*) for a particular value of Y, y, given bt $H(y)$=Pr($Y \leq y$). Then $H(y)$ has the representation

$$h(y_1, y_2, \ldots, y_T) = \int_0^1 L(\theta)dF(\theta) \qquad (1.5.1)$$

where θ stands for the probability of success assigned to a probability distribution with *cdf* F(.), interpreted as a belief on the relative frequency of $\bar{Y}_T \leq \theta$ as $T \rightarrow \infty$; with:

$$L(\theta) = \prod_{t=1}^{T} \theta^{y_t}(1-\theta)^{1-y_t} \qquad (1.5.2)$$

$$F(\theta) = lim_{T\rightarrow\infty} P_H(\bar{Y}_T \leq \theta) \qquad (1.5.3)$$

Thus, given θ, F(.) may be interpreted as belief about the long run frequency of $\bar{Y}_T \leq \theta$ as $T \rightarrow \infty$. Exchangeability generalizes the concept of independence by demonstrating that identically distributed sequences are not necessarily independent unless exchangeable sequences all have equal probability assigned to them, that is iff the probability given to specific sequences of events does not depend on the order of "successes" and "failures".

Often, however, one has some idea about the likely probability of a parameter of interest to formulate an "informative" prior to reflect the odds about the hypothetical bets taken on the value of the true unknown parameter β. The employment of an informative prior is the main bone of contention between the classical and Bayesian schools, for example, how reliable the estimated posterior density would be if the informative density function has an incorrect prior distribution? The Bayesian approach provides protection against functional form mis-specification by employing a flexible framework that allow the posterior to belong to the same family distributional function as the prior, making the two distributions *conjugates* of each other. However, such flexibility may not always be enough to detect mis-specification, especially with large micro-data where variables may relate to each other in ways that are not easily identified by assumed, though flexible functional forms. In such cases, functional-form-free distribution is an attractive alternative option.

The algebraic choice of informative prior is more difficult, and often unavailable. For instance, to find the point estimate with a quadratic loss function requires estimating an integral for the mean of the posterior function; analytical solutions for integrals are frequently unavailable, and solutions by numerical methods are computationally demanding in high dimension integrals (those with a vector of parameters). However, improved computer power has overcome such problems by making possible the estimation of an integral by simulation. Simulation relies on the mathematical feature that an integral has an expected value interpretation and obtainable from $\int h(x)f(x)$ where x is a random variable with density $f(x)$. We draw an x value from $f(x)$ to calculate each $h(x)$ a large number of times by Monte Carlo simulation methods and then average all $h(x)$ values.

The *Markov Chain Monte Carlo* (***MCMC***) method has long sequentially drawn simulated values that converge to a stationary invariant distribution corresponding to the target posterior density. The two most common *MCMC* simulation methods to find such invariant distribution are the **Gibbs sampler** and the *Metropolis-Hastings* (***MH***) algorithms; the former is a special case of the latter. The Gibbs sampler employs a large number of random draws sequentially from alternating blocks of conditional densities that converge on the invariant posterior distribution; to start the simulation, a portion of the initial sample is disregarded as the *burn-in* sample. For example, the simulation chain for the posterior distribution of the normal linear homoscedastic model with normal-gamma priors consists of repeated draws from the normal distribution conditional on the precision parameter (inverse of variance) $\sigma^{-2}$and from the gamma distribution conditional on the normal β. The MH algorithm simulates from a *proposal* density that covers the range of values for the posterior *target* density, it is more general than the Gibbs method and used when sampling from the blocks of conditionals is unavailable. We discuss Bayesian simulation methods in some detail in chapter 18, see also exercise questions 1.6, for Bayesian linear model with the MH simulation, and 4.3, for the Bayesian random effects with the Gibbs simulation.

Several advantages of the Bayesian approach briefly outlined above become evident when contrasted with the classical approach. The main contrast is that a classical approach justifies a good estimator by appealing to its asymptotic behavior as sample size $N \rightarrow \infty$ in hypothetical repeated sampling, while a Bayesian approach obtains point estimation conditional on the actual data modified by the investigator's subjective priors. The classical school is called the *frequentist* in the Bayesian literature in view of its definition of probability based on the relative frequency with which an event occurs in repeated sampling. However, as the sample increases in size, the Bayesian estimator collapses on the MLE estimator because the importance of the prior diminishes and the actual data dominate the outcome; the likelihood mean and mode become identical.

Another important contrasting aspect of the Bayesian method is its approach to hypothesis testing. Since the Bayesian mothed is not producing a single estimate to test against the true value but a posterior distribution; its hypothesis tests are based not on point estimates but on comparing entire estimated posterior distributions. For example, suppose model M_1 has parameter $\beta \leq 1$ while model M_2 has parameter $\beta \geq 1$, then the integral of the posterior of $\int_{-\infty}^{1} \beta_p$=*prob*($M_1$), with *prob*($M_2$) =1 - *prob*($M_1$), and their ratio, called the *posterior odds ratio*, summarizes this information, therefore, only model comparison rather than significant tests are relevant in the Bayesian approach[4].

The algebra of Bayesian methods is usually considerably more difficult than that of the classical approach. For example, the classical analysis of a multivariant regression requires the regression errors to be normally distributed, while the Bayesian linear multivariant regression requires a multivariate normal-gamma prior, combined with a multivariate normal likelihood for the data and results in a multivariate normal-gamma posterior. However, the applied econometrician can rely on computer software to carry out the necessary simulations to produce the required posterior distribution.

Table 1.5. provides an applied example for the regression of wage on age for a sample of women aged 18-45 in the US labor market.

Table 1.5 linear regression of womens wage on age by least squares and linear Bayesian estimators.

Wage	linear least squares regression	linear Bayesian regression*
Age	0.3994 (0.0605)	0.4009 (0.0596)
Constant	6.0331 (1.7915)	5.9691 (1,7372)

*Prior distributions: slope parameters normal (0, 10000); variance inverse Gamma (0.01, 0.01) simulated by MCM-*MH* method.

[4] There is a subtle difference between selecting an optimal model from among a number of models regardless of its true status, and testing for a parameter in relation to its true value. The former always leads to definite outcome and therefore most relevant for making a decision, while the latter need not have a definite outcome because the rejection of a hypothesis does not necessary suggests accepting any of the alternatives.

We note that the priors employed in table 1.5.are normal for parameters and inverse-gamma for variance. The *OLS* and Bayesian means and standard errors are very similar, suggesting the priors are fairly uninformative. The exercise 1.6 implements the regressions for this example; further example is exercise 4.3 with applications based on the classical and Bayesian panel data random effects logit estimators.

We end this chapter by drawing attention to two different principles around which all the topics discussed in the rest of this text are organized. These two principles are rooted in two very different view of probability, one based on estimation of the unknown true parameters asymptotically by appeal to the *LLN* and *CLT*, the other by combining subjective priors with the data to produce estimates with the smallest predictive loss (due to error). These are two different outlooks about the interpretation of reality; since they essentially constitute different readings of the same evidence, the superiority of one over the other can never be empirically decided; they are two different “paradigms”. If the history of science is any guide, the place of each will be decided by the comparative success of each econometric approach to explain new development’, invent new tools to explore them, and the comparative simplicity of the solutions they offer. In any case, it is not the aim of this applied text to convince; instead, we draw attention to the areas where each approach can provide a relatively more effective solution in order demonstrate the usefulness of learning both approaches. To cite two examples, the classical approach to non-nested model selection involves formulating a third all-encompassing model; a procedure that is sometime hard to implement, see Pesaran (2015, chapter 11); by contrast, the Bayesian approach to model selection, discussed in section 17.1, requires a simpler method to implement testing for the optimal model selection. Even in the classical approach, the Bayesian model selection is the preferred alternative to the classical method of combining models by polling their point forecasts, see (Granger and Pesaran , 2000). On the other hand, the classical econometric approach has a well-developed body of distribution-free, though computationally intensive, estimators to protect against functional form mis-specification, while the Bayesian approach cannot easily preform such estimation. What is called distribution-free Bayesian estimators on a closer examination turns out to consist of flexible distributions models rather than distribution-free ones, see Greenberg (2013, chapter 9). Those interested in applied econometrics would likely produce more effective research if they identified and learned about such differences. It would be helpful to bear in mind that a range of topics to which each approach applies are in fact organized around two fundamentally different principles: one based on asymptotic estimation and testing, the other on posterior estimation obtained from combining subjective probability with data.

Readings

For textbook discussion, Cameron and Trivedi (2005, chapter 5); Wooldridge (2010, chapters 12 & 13).

Chapter 1 MLE, Nonlinear & Bayesian Exercises

Q1.1 If $f(y|\mathbf{x};\theta)$ is a correctly specified model for the density of y_i, does θ_0 solve by maximization of the conditional expectation of y_i, $\text{Max}_{\theta\in\Theta} E[f(y_i|\text{x}_i;\ \theta)]$?

Q1.2 Consider a general binary response model $\text{P}(y_i = 1|\text{x}_i\text{=G(x}_i, \theta_0)$, where $0< \text{G}(\text{x},\theta)<1$ for all x and θ; x and θ need not have the same dimension: let **x** be a *K*-vector and θ a *P*-vector.

a. Write down the log likelihood for observation *i*,
b. Find the score for each *i*; show directly that $E[\boldsymbol{s}_i(\theta_0)|\mathbf{x}_i] = 0$.

Q1.3 Suppose a before making a decision to publish a text book in hard-cover, a publisher surveyed 45 readers and found 15 preferred hardcopy edition while 30 preferred paperback edition.

(a) What is the maximum likelihood estimate of θ, the probability that a customer will buy a hard copy volume?
(b) Using a uniform prior what is your posterior distribution for θ?
(c) What is the mean of this distribution? *Hint*: The beta distribution given by $f(x) \propto x^{\theta-1}(1-x)^{\phi-1}$ has mean $\frac{\theta}{\theta+\phi}$.

Q1.4 Suppose you program a computer to do the following:

i. Draw 50 *x* values from a distribution uniform between 2 and 22
ii. Draw 50 *e* values from a standard normal distribution
iii. Create 50 *y* values using the formula *y*=2+3*x*+4*e*
iv. Regress *y* on *x* obtaining the sum of squared residuals *SSE*1.
v. Regress *y* on *x* for the first 20 observations, obtaining *SSE*2.
vi. Regress *y* on *x* for the last 30 observations, obtaining *SSE*3.
vii. Add *SSE*2 and *SSE*3 to get *SSE*4.
viii. Calculate *w*1=(*SSE*1-*SSE*4)/*SSE*4

ix. Repeat the process described beginning with step (ii) until 3000*w* values have been created, *w*1 through *w*3000.

x. Order the 3000*w* values from smallest to largest.

What is your best guess of the 2970th of these values? Explain your reasoning.

Q1.5 Download *mus10data.dta* and select year02; the data set contains the number visits to a physician’s office, and individual characteristics are: private insurance, chronic condition, gender and income.

a. Apply MLE for *docvis* Poisson regression with robust standard error estimates, and comment on the results.

b. Apply the NLE for the same regression, and compare the outcome with that in a.

c. Interpretation of the coefficients is an issue in non-linear estimation. Obtain the marginal effects for a. first by finite-differences, then by calculus methods; comment on the differences

d. Compare the marginal effects in c. at mean with the estimates at representative value, average value

e. Using the mean value of regressors, compute first elasticity, then semi-elasticity for the impact of a unit change in income on the probability of *docvis*.

Q1.6 Download *womenwage.dta*, containing wages (in $1000,s), age, years of completed schooling and experience (tenure) for women over 18 years of age and in the fertility cycle.

a. Fit a least squares model of wage income on age first, then fit a corresponding Bayesian regression. Since the second regression is based on simulation, set a random-number seed to 15 start the reproducible results.

b. Compare the estimates from the least squares with the Bayesian regressions, comment on their difference and other features of the Bayesian outcome.

c. Predict the expected wage of a 40-year-old woman conditional on the above Bayesian posterior model.

Chapter 2 Generalized Method of Moments

Introduction

Many applications of Instrumental Variables require estimating a *system* of IV equations rather than a single-equation estimation. The modern approach to **system instrumental variables** (***SIV***) estimation employs the **generalized method of moments** (***GMM***) to estimate a system of IV equations. This section examines some of the properties of a system IV and estimation procedures that have applications beyond a system of simultaneous equations such as the analysis of panel data, see chapter 5.

2.1-*Method of Moments Approach*

The fundamental assumption of least squares consistency is that the error term must be uncorrelated with the mode's regressors, i.e. $E(u|\mathrm{x})=0$. Then the conditional mean is $E(y|\mathrm{x})=\mathrm{x}'\beta$, and estimated β provides a consistent measure of the causal effect of the regressors on y. When this assumption is violated, the OLS coefficient estimates are no longer the measure of the marginal effect of the regressor x_j on the dependent variable y, that is $E(y|\mathrm{x})\neq\mathrm{x}'\beta$. However, consistent estimation is still possible on the strong assumption that there exits an instrument vector z_j highly correlated with x_j but uncorrelated with u, i.e. $E(u|\mathrm{z})=0$. Then differentiating the expected loss

$$E\ (u)^2=E[(y - \mathrm{x}'\beta)^2] \qquad (2.1.1)$$

Solving for β yields the optimal linear predictor

$$\beta = E\ ([\mathrm{xx}'])^{-1} - E[\mathrm{x}y])=0 \qquad (2.1.2)$$

If the residual in (2.1.2) is obtained conditional on the vector of z, then (2.1.2) becomes the **IV** sample analogue of the OLS estimator; this is an example of the **method of moments** (**MM**) estimator, and provides the basis for more complicated **IV** estimation models examined below. The instruments are obtained from the moment conditions implied by $\boldsymbol{E(u|\mathrm{z})}=0$, namely

$$E(u\,|\mathrm{z})= E[(y - \mathrm{x}'\beta)|z] =0 \qquad (2.1.3)$$

The main interest in going beyond single-equation IV application is in estimating a *system* of equations with endogenous explanatory variables, typically a set of structural equations, their reduced form equations, but also, a single-equation panel data over *T* different time periods.

Consider the following general linear model: Y_i=x$_i\beta$+ u_i

where $\boldsymbol{Y_i}$ is a (G x1) vector, $\mathbf{x_i}$ is a (G x K) matrix, and $\boldsymbol{u_I}$ is the (G x 1) vector of errors.

A typical example of system ***MM*** application is to use the sample mean as an estimate of the population mean. In general, the ***MM*** estimator solves the sample moments that correspond to the population moments. ***MM*** solves for the corresponding sample moments by

$$\frac{1}{N}\sum_{i=1}^{N} z_i{}'(y_i - x'\beta) = 0 \qquad (2.1.4)$$

in order to obtain the linear IV estimator

$$\hat{\beta}_{MM=(\sum_{i=1}^{N} z_i x_i')^{-1}\ \sum_{i=1}^{N} z_i y_i} \qquad (2.1.5)$$

More generally, let Z and X stand for the vectors of instruments and variables, then (2.1.5) in matrix form is

$$\hat{\beta}_{MM=(Z'X)^{-1}\,Z'y} \qquad (2.1.6)$$

Sometimes, MM estimation may be impossible even with plausible instruments available if there are more moment conditions, hence more equations to solve, than there are parameters. In such cases, the **MM** estimator can be extended into an alternative estimator by a method due to Hansen (1982) and known as the **generalized method of moments** (**GMM**) that can accommodate the case of over-supplied instruments.

2.2 *MM, 2SLS and GMM estimators*

GMM defines a class of estimators based on different choices of moment condition, and different weighting ***w*** for variance. Four assumptions are required to establish parameter identification, consistency and efficiency of GMM estimators, though not all are needed for MM estimators; the main focus of GMM estimation is on efficiency. Define the existence of ***r*** moment conditions for ***q*** parameters more generally by

$$E\ [\mathrm{h}\ (w_i, \theta_0)] = 0 \qquad (2.2.1)$$

Where $w_i$contains all the system's variables (y, x, z), θ_0 stands for the value of θ in the *dgp*, and **h**(.) defines an (r x 1) vector function that determines the relationship between the variables, namely, it specifies their functional forms. The following assumptions are required for consistency and efficiency:

I-*orthogonality condition:* the *dgp* imposes the moment conditions $\boldsymbol{E}$ [$\boldsymbol{h}$ ($\boldsymbol{w_i}, \boldsymbol{\theta_0}$)] =0, with $\mathbf{z_i}$ as a matrix of observable instruments. This assumption however, is not enough for identification.
A *sufficient* assumption for identification is the **rank condition**
II-*Rank condition*: rank of $\boldsymbol{E}$ ($\mathbf{z_i'x}$)=$\boldsymbol{K}$. This assumption requires that the columns of $\boldsymbol{E}$ ($\mathbf{z_i'x}$) be linearly independent of each other[5]. Since minimization of (2.1.2) requires an invertible *W* matrix, we also need to assume W has a nonsingular probability limit:

III-*Probability limit of W* : $\widehat{W}_{\text{p}} \rightarrow W$ as $N \rightarrow \infty$ where *W* is a symmetric positive definite matrix. Convergence follows from the law of large numbers because $\widehat{W}$ is a function of sample averages.

2.3 ***Distributional condition***: $N^{-1/2} \sum_{i=1}^{N} h_i |\, \theta_0 \;\rightarrow^{d}\; N\,(0,\; S_{\beta 0})$ where

$$S_{\beta 0} = plim\ N^{-1} \sum_{i=1}^{N} \sum_{j=1}^{N} [h_i h_j' \ |\theta_0] \qquad (2.3.1)$$

where h_i(.) is the *ith* component of *h*(.).

There are three possible relations between the number of parameters and the number of instruments. The first case is if *r* = *q*, the model is said to be **just-identified** since we have as many unknown parameters as we have instruments, one instrument for each parameter, or each equation (with exogenous variables trivially acting as their own instruments). Then the application of the law of large numbers shows that the MM estimator leads to consistently identified parameter estimation solved by (2.1.5) and (2.1.6) for $\boldsymbol{\widehat{\beta}_{MM}}$. Second, if ***r*** < ***q***, this is the **under-identified** case with fewer instruments than unknowns; then no consistent ***MM*** exists, quite a common situation in practice. Third, when ***r*** > ***q*** with if more instruments than unknown parameters, known as the **over-identified** case. Then (2.1.3), *Z*(*y* - Xβ)=0, has no solution there is no unique relationship . One possibility is to drop as many of the surplus instruments as necessary to reduce the case to a just-identified one. However, this means abandoning useful information that, if employed, can lead to more efficiency. Instead, we can settle on choosing $\hat{\beta}$ so as to make the loss

[5] A necessary condition for this assumption in a system of equations requires the Order condition for parameter identification; the rule is that the number of excluded exogenous variables must be greater or equal to the number of predetermined (endogenous or lagged) variables for each equation at hand.

function (2.1.1) to be, not equal to zero but *as "small"* (*close to zero*) *as possible*. In this case, we have (after dropping 1/*N* to simplify)

$$\hat{\beta}_{MM}\text{=}[\textstyle\sum_{i=1}^{N} z_i'(y_i - x_i\,\hat{\beta})]'[\sum_{i=1}^{N} z_i'(y_i - x_i\,\hat{\beta})] \qquad (2.3.2)$$

and the MM estimator chooses $\hat{\beta}_{MM}$ so as to make (2.3.2) as small as possible. Although the results are consistent based on the assumptions I & II above, the method very often fails to produce the best estimator. However, GMM is a more general estimator that uses a weighing matrix to solve for a vector of $\hat{\beta}$ by a quadratic loss function in β by

$$min.\hat{\beta}\ [\textstyle\sum_{i=1}^{N} z_i'(y_i - x_i\,\hat{\beta})]'W_N[\sum_{i=1}^{N} z_i'(y_i - x_i\,\hat{\beta})] \qquad (2.3.3)$$

where *r* x *r* weighing matrix W_N is symmetric positive definite and independent of β; the subscript *N* indicates its value dependents on the sample. Note that the dimension *r* (number of moment conditions) of W_N is fixed as *N* → ∞. Different choices for weighing matrix W_N lead to different estimators, all consistent but with different variances. In the case of overidentified models, it can be shown that (2.3.3) leads to the unique solution of $\hat{\beta}$ (see below).

First, consistency of GMM is based on the first three assumptions above:

Theorem 1 (*consistency of GMM*)*: under assumptions* I-III, $\hat{\beta}_{\mathrm{p}} \rightarrow \beta$ as N → ∞ (see Appendix for a proof); $\hat{\beta}_{GMM}$ is also asymptotically normally distributed with

$$\mathcal{N}\ [0,\ \Lambda \equiv E(\mathrm{z}_i'\mathrm{u}_i\mathrm{u}_i'\mathrm{z}_i) = Var(\mathrm{z}_i'\mathrm{u}_i)] \qquad (2.3.4)$$

When *r* =*q*, the GMM estimator is (2.1.5) and no matter how $\widehat{W}$is chosen, x'z is a *K* x *K* nonsingular matrix. While all GMM estimators are consistent, they differ in their different variances $S_{\beta 0}$ given by (2.1.1). However, by appropriate choices of $\widehat{W}$, (2.1.1) can be greatly simplified. Let us examine the choices. First, if the ***i*** and ***j*** observations are independent of each other, cross products in assumption IV disappear

$$S_{\beta 0}\text{=}\ plim\ N^{-1}\textstyle\sum_{i=1}^{N}[h_i h_i']\ \theta_0] \qquad (2.3.5)$$

In this case, substitute $\widehat{W} = (N^{-1}\sum_{i=1}^{N}[h_i h_i']\ |\theta_0]$ in (2.3.2) for $\hat{\beta}$equation results in the estimator for a *system of 2SLS equations*; because it extends the single-equation 2SLS to a system of equations, it also called the **generalized instrumental Variable estimator (*GIVE*)**. Second, when

r = ***q***, the just-identified GMM results simplify to those already given above and obtainable from (2.3.4). In this case, MM and GMM estimators and the estimates are invariant to the choice of the weighing matrix. Third, with ***r*** > ***q***, the best choice is a weighting method that simplifies the asymptotic normality of ***GMM*** by setting **W=** $\boldsymbol{\Lambda^{-1}}$ where, that is

$$\Lambda^{-1} \equiv [E(\mathrm{z}_i'\mathrm{u}_i\mathrm{u}_i'\mathrm{z}_i)]^{-1} = Var(\mathrm{z}_i'\mathrm{u}_i)]^{-1} \qquad (2.3.6)$$

that is the *optimal weights matrix are set equal to the inverse of the variance matrix*.

Assumption IV: Optimal weights condition: W= Λ^{-1} where Λ is defined as above.

Addition of assumption *IV* to those already stated leads to the theorem due to Hansen (1982):

Theorem 2 (*Optimal weighing matrix*)*: under assumptions I-IV, the GMM estimator is the most efficient among the class of GMM estimators.* This estimator is also called the ***two-step GMM*** estimator because it estimates the predicted values in a *system* of equations from several 1st stage regressions and then employs them in place of actual values in the 2nd stage.

Procedure for the application of GMM optimal weighting matrix:

a. Start with an initial $\hat{\beta}$ estimator of β, this is usually the system 2SLS estimator.
b. Obtain the vector of the residual $\widehat{\mathrm{u}_\iota} = y_i - x_i\hat{\beta}$ for *i*=1, 2, . . . *N*
c. Consistent estimator of (2.3.5) is $\widehat{\Lambda}$ = $[N^{-1}\sum_{i=1}^{N} z_i'\widehat{\mathrm{u}_\iota}\,\widehat{\mathrm{u}_\iota}z_i]$
d. Set $\widehat{W}$=$\widehat{\Lambda}$ $^{-1}$=$[N^{-1}\sum_{i=1}^{N} z_i'\widehat{\mathrm{u}_\iota}\,\widehat{\mathrm{u}_\iota}z_i]^{-1}$ and use this matrix to obtain the asymptotically optimal GMM estimator (2.3.6) estimated by

$$\{(\mathrm{x'z})\ [\textstyle\sum_{i=1}^{N} z_i'\widehat{\mathrm{u}_\iota}\,\widehat{\mathrm{u}_\iota}z_i]^{-1}(\mathrm{z'y})\} \qquad (2.3.7)$$

(2.3.7) is similar to (2.1.6) for the linear case, except that GMM estimator is weighted by (2.3.6). The square roots of the diagonal elements of this matrix are the asymptotic standard errors of the optimal GMM estimator and it is called the **minimum chi-square estimator**.

As a MM estimator, the OLS estimator of the sample mean can be an inefficient estimate of the population mean if the data are not from a normally distributed random sample; and remains so even with homoscedastic errors, if the errors are not normal. Then the sample median provides consistent estimates that may be more efficient than the sample mean based on the assumption that the errors are conditionally *symmetric*. Hence, instead of an OLS MM estimator based on E[xμ]=0, we can make the additional moment assumption that E[u^3|x]=0 that implies E[u^3.x]=0. The estimator generates moment condition based on

$$\begin{bmatrix} E[x(y - x'\beta)] \\ E[x(y - x'\beta)^3] \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

This solves for the sample moment conditions with 2K equations but only for K unknown parameters. However, the GMM estimator can be employed to set the variance as small as possible based on a quadratic loss function so $\boldsymbol{\hat{\beta}_{GMM}}$ minimizes

$$\beta_{\mathrm{Q}_N} = \begin{bmatrix} \frac{1}{N}\sum_i x_i x_i^{/} u_i] \\ \frac{1}{N}\sum_i x_i\; x_i^{/} \mathrm{u}_i{}^{3}\ ] \end{bmatrix}' W_N \begin{bmatrix} \frac{1}{N}\sum_i x_i x_i^{/} u_i] \\ \frac{1}{N}\sum_i x_i x_i^{/} u_i{}^{3}\ ] \end{bmatrix}$$

Where $u_i = (y - x'\beta)$ and $W_N$is a (2*K* x 2*K*) weighting matrix. Choices for W_N can lead to more efficient estimators than the OLS. The sample median can be shown to have asymptotic variance 1/*N* whereas the sample mean is inefficient with variance *V* [*y*]/*n*=2/*N*. This optimum *GMM* estimator with two moment conditions gives much lower weights to the second moments if it has high variance.

The *IV* approach has traditionally been applied in the context of a system of simultaneous equation with right-hand endogenous variables, first to estimate a system of reduced form equations, and then to recover the structural parameters from these estimates. Consistency requires the rank condition. Then, the equation-by-equation application of 2SLS renders consistent estimation of the structural model. However, a more efficient system estimator is the ***3SLS*** that assumes *homoscedastic errors* for each equation but correlated across equation errors terms, and exploit this correlation to improve estimation efficiency. First, we obtain the reduced form OLS estimates and the predicted values, then we use the predicted in place of actual values in second regressions, and finally, since the errors are correlated across equations, we estimate a system of structural equations with correlated errors by an estimator similar to the *SURE*, see chapter 5.

An important task with the application of the *GMM*, based on more moments (r) than parameters (q), is testing for an over-identification restriction for instrumental validity. Assume we have moment conditions with $r > q$ and $E[h\ (w_i, \theta_0)]$ =0 as defined by (2.2.1). The overidentification test is based on

$$\frac{1}{N}\sum_i \hat{h}_i = h(w_i, \hat{\theta})$$

For the over-identification case, $\sum_i \hat{h}_i \neq 0$ since $r > q$. Given θ estimated by $\hat{\theta}_{GMM}$; Hansen (1982) shows that the over-identification restriction (*OIR*) test statistic is

$$OIR = (\tfrac{1}{N}\sum_i \hat{h}_i\ )'\ \hat{S}^{-1}(\tfrac{1}{N}\sum_i \hat{h}_i)$$

where $\widehat{W} = \hat{S}^{-1}$ is distributed as $\chi^2\,(r-q)$ under H_0 : E[h (w_i, θ_0)] =0. Large *OIR* suggests rejection of the population moments and inconsistent *GMM* estimator.

The weak instruments test is another related test based on the first stage reduced-form $\boldsymbol{R^2}$ result and the $\boldsymbol{F}$ statistic for the joint significance of the main instruments. Typically, instruments are weak when the fit for the first-stage regression is poor, or the number of instruments is very large relative to the sample size, as will be discussed in the dynamic panel data models in chapter 5. A common statistic employed is a partial $\boldsymbol{R^2}$ of a regression on one error term obtained from regression of *y* on exogenous *x* variables; another error term obtained from a regression of the instruments *z* on the exogenous variables *x*. This method, generalized to structural equations with more than one endogenous variable, produces the *Shea's partial* $\boldsymbol{R^2}$. With more than one endogenous variable, there will be more than one first-stage regression and more than one $\boldsymbol{F}$ test. Then, the *minimum eigenvalue* of a matrix analog of the $\boldsymbol{F}$ statistic is used to test weak instruments. We reject the null hypothesis of weak instruments if the $\boldsymbol{F}$ statistic is greater than 13.9. Two types of formal tests of weak instruments tests are available. The first pre-selects the largest bias of 2SLS relative to that of OLS; the test requires at least three instruments with one endogenous variable. The second tests for significance of the endogenous regressors in the structural model pre-selecting a tolerance level for the test size distortion of the Wald statistic. The test has its own table of critical values, see empirical exercise 2.2_c-h.

Another important application of the *GMM* is dynamic panel data analysis where an MM estimator is based on the moment condition E [x$_{it}u_{it}$]=0. If we also assume that u_{it} is uncorrelated with regressors in periods other than the current time, then E [x$_{it}u_{it}$]=0 for $s \neq t$ supply additional instruments employed for more efficient estimators. Note that the lagged instruments themselves are likely to be correlated, hence the application of two- step GMM is commonly employed in dynamic panel data, see chapter 6.

Appendix

Proof of Theorem 1: We can re-write (2.3.3) as $\hat{\beta}_{GMM}$=(X$'$Z$\widehat{W}Z'$X)$^{-1}$(X$'$Z$\widehat{W}Z'$X), or

$$\hat{\beta}_{GMM}= [(N^{-1}\textstyle\sum_{i=1}^{N}\mathrm{x}_i'\,z_i)\;\widehat{W}(N^{-1}\textstyle\sum_{i=1}^{N}\mathrm{z}_i'\,x_i)]^{-1}(N^{-1}\textstyle\sum_{i=1}^{N}\mathrm{x}_i'\,z_i)\;\widehat{W}(N^{-1}\textstyle\sum_{i=1}^{N}\mathrm{z}_i'\,y_i)$$

The last term in brackets in this expression is in fact (2.1.4); substituting for y_i by (x$_i\beta + u_i$) , and canceling out positive and negative x$_i\beta$, leaves u_i in (2.1.4). Writing the above with u_i , and, since inconsistency implies $\hat{\beta}_{GMM} \neq$ β, we have

$$\hat{\beta}_{GMM}= [(N^{-1}\textstyle\sum_{i=1}^{N}\mathrm{x}_i'\,z_i)\;\widehat{W}(N^{-1}\textstyle\sum_{i=1}^{N}\mathrm{z}_i'\,x_i)]^{-1}(N^{-1}\textstyle\sum_{i=1}^{N}\mathrm{x}_i'\,z_i)\;\widehat{W}(N^{-1}\textstyle\sum_{i=1}^{N}\mathrm{z}_i'\,\mathrm{u}_i)$$

Under assumption II, E (z$_\mathrm{i}'$x)≡C has rank K ; and implies that ($C^/WC$) has also has rank K by assumption III, and is therefore nonsingular matrix with its inverse as ($C^/W$C)$^{-1}$. As N→∞, by the law of large numbers

$$plim\;\hat{\beta}_{GMM}=\beta+(C^/W\mathrm{C}\;)^{-1}\;C^/W\;(plim N^{-1}\textstyle\sum_{i=1}^{N}\mathrm{z}_i'\,\mathrm{u}_i)=\beta+(C^/W\mathrm{C}\;)^{-1}\;C^/W.\;0 = \beta. \qquad \text{QED}$$

Readings

For textbook discussion, see Cameron and Trivedi (2005, chapter 6), and Wooldridge (2010, chapters 8 and 14). Hansen (1986) developed the GMM approach.

Chapter 2 GMM Exercises

Q2.1 Employ the GMM estimator that solves the minimization problem (2.3) to show that

a. the solution for this GMM estimator satisfies the 1st-order condition $\sum_{i=1}^{N}(z_i'x_i)\,\widehat{W}(\,\sum_{i=1}^{N} z_i'(y_i - x'\beta))$=0

b. use this expression to obtain a more general weighted solution of (1.7) for $\widehat{\boldsymbol{\beta}}_{\boldsymbol{GMM}}$.

Q2.2 Download *mus06data.dta*, the MEPS data set on log of drug expenditure, *ldrugexp*, for > 65 with 4 instruments.

a. Just-identified case: create a macro for the list of covariates (x2list), then use just one instrument, *ssiratio*, apply 2sls with robust standard errors and 1st stage requested; comment on the instrument.

b. Compare 2sls robust, the *gmm* with heteroskedasticity, the *gmm* clustered on age, with simple 2sls. Any notable change?

c. Apply the Hausman test for *hi_empunion* endogeneity in 2sls regression; what is the result?

d. Test for over-identification by the Hansen/Sargan method for the *gmm* estimator, test result?

e. Use all 4 instruments for the *gmm* & test over-identification, and explain the outcome

f. Use Stock-Yogo to test for weak instruments from the *gmm* regression; carefully state the result.

g. Generate an asymmetric restriction on a two-equation structural model and apply **3*sls*** estimator, with the following features. 1st : just-identified equation *ldrugexp* regresses *hi_empunion* and all exog. var.s; 2nd: over-identified equation regresses *hi_empunion* on *ldrugexp* and *ssiratio*, but exclude *age & linc*.

h. Explain how the model's variables meet the 3*sls* assumptions for consistency

Chapter 3 Discrete Dependent Variables Models

Introduction

Most economic variables are constrained in their range of variation, usually to positive values, for example the rate of interest, but many among them are discrete variables that assume only a small number of values. Such models fall broadly into two categories: first are those that are a binary indicator taking just two values, (0, 1), or categorical indicators with more than two outcomes, for example a decision to apply for a graduate program, or a decision to take a bus, train or private car to work; the second are the models that are a different mixture of a discrete indicator variable and a continuous variable, for example a decision to buy a car, and if so how much to spend. The former model is known as a *binary, or categorical dependent models*; the second as the **limited dependent (LD) variable Models**. We first examine the dummy dependent models. Count data models are employed to analyze discrete data that takes a limited number of positive values. These play an important role in many areas of microeconometrics examined in this text, and for that reason we employ the Poisson regression with count data.

3.1 *Analysis of Count data*

Count data: non-negative dependent variable when the occurrences of an event have relatively few values, including zeros, over a specified interval of time or space, namely, the *number* of visits to a physician over the course of a year, the number of Covid19 patients on a ventilator during a day in a local hospital, the number of small business bankruptcies during a week in a city, the number of car accidents over a specified segment of a motorway, etc.; the model is extensively employed in health economics.

Poisson regression model

The ***Poisson probability distribution*** provides the foundation of the count data models. If a Y is a Poisson random variable, then its probability density function is

$$f(y)=Pr(Y=y)=\frac{e^{-\lambda}\lambda^{y}}{y!} \qquad (3.1.1)$$

where the *factorial* term $y!=y.(y\text{-}1).(y\text{-}2)\ldots1$; and λ is the mean of Y. This probability distribution has only parameter λ ; a key property of this distribution is that its *mean and variance are equal*:

$E(Y)=Var(Y)=\lambda$ (3.1.2)

The model assumes that the probability of occurrences in different time intervals are independent of each other. We parametrize the Poisson function by the exponential mean function as

$E(Y|\boldsymbol{x})=\lambda=exp(\boldsymbol{x}\beta)$ (3.1.3)

The Poisson equality of the mean and the variance makes the function inherently *heteroskedastic*. The log-likelihood for the Poisson function is maximized numerically by

$ln\ L(\beta)= \sum_{i=1}^{N}\{y.x_i'\beta - exp(x_i'\beta) - ln(y!)\}$ (3.1.4)

Since log of the nominator of (3.1.1) consists of sum of two log terms, and by (3.1.3), log of $e^{-\lambda} = -\lambda$; usually, $ln(y!)$ is dropped since it does not dependent onβ. The prediction of the conditional mean of y, given $\widehat{\beta_\iota}$ and a selected a value of x_0, can then be obtained from

$$E(\widehat{y_0}) = \widehat{\lambda_0} = \exp\ (\widehat{\beta_\iota} x_0)$$

Moreover, the probability of a particular number of occurrences can be estimated by inserting the estimated conditional mean into the probability function

$$Pr(Y = y) = \frac{\exp(-\hat{\lambda}0).\hat{\lambda}0^y}{y!}\ , y\text{=0, 1, 2, …}$$

The Poisson model is a *deterministic* function of the explanatory variables, that is, the model produces the same outcome for otherwise identical individuals.

Marginal effects and interpretation

The marginal effect of a change by one unit in a continuous explanatory effect ***x*** is obtained from the conditional mean: $E(y_i)=\lambda_i=exp(x_i\beta)$, and the derivative of exponential $\frac{\partial E(y_i|\boldsymbol{x})}{\partial x_i} = \beta_i \exp(\boldsymbol{x}'\beta) = \beta_i\lambda_i$. The coefficient β_i measures the relative change in $E(y_i|\boldsymbol{x})$ caused by a one-unit change in x_i. However, if x_i is measured on a log scale, then β_i is an elasticity estimate: $\frac{\%\Delta E(y|\boldsymbol{X})}{\partial x_i} = 100\frac{\partial E(y_i)/E(y_i)}{\partial x_i}$= $100\beta_i\%\Delta x_i$. That is $100\beta_i$ is approximately the percentage change in $E(y_i|\boldsymbol{x})$ induced from a unit increase in x_i. For a comparison of the exponential with the OLS slope estimate of x_i, compare the estimate $\hat{y}_{iols}$, $\hat{y}_{iexp}$ with $(\widehat{\beta_1}.\ \bar{y})$.

Over-dispersion and quasi-maximum likelihood.

The Poisson regression is too restrictive because its distribution is in terms of a single parameter λ. One consequence, the Poisson regression predicts the probability of zero count very much smaller than the actual sample zero observations; known as the *excess zero problem*. A more important deficiency: the Poisson estimated variance is often larger than the mean. This is known as the **overdispersion** problem; overdispersion has a consequence similar to heteroscedasticity in the linear regression in that the Poisson MLE remains consistent if the conditional mean is correctly specified. However, large over-dispersion results in hugely understated standard errors, and a very large overstated *t*-ratio, hence robust variance estimation is important. The presence of over-dispersion can also be evidence of misspecification; over-dispersion leads to inconsistency, not just to inefficiency. Estimation by the ML when the density function is misspecified (but the mean is correctly specified) is called quasi-LME (***QLME***); the Poisson *MLE* and *QMLE* are identical but have different variances; the *QMLE* uses either heteroskedasticity-robust standard errors, or corrects for them. With the latter, the standard errors of the Poisson *QMLE* must be adjusted. A simple method is to assume the unknown variance is proportional to the mean:

$$Var(y|\boldsymbol{x})= \sigma^2 E(y|\boldsymbol{x}) \; ; \sigma^2 > 0 \qquad (3.1.5)$$

When $\sigma^2 =1$, the variance is that of the Poisson model; but when $\sigma^2 >1$, then the variance is larger than the mean and hence larger than the Poisson variance, a common outcome in many Poisson applications.

We test over-dispersion by *LR* statistic, comparing it to the less restricted variance; the statistic is known as the *quasi-likelihood ratio statistic* obtained by dividing $LR=2(\mathscr{L}_{ur} - \mathscr{L}_r)$ by unrestricted $\widehat{\sigma^2}$. More specifically, specify overdispersion of the form, given $\boldsymbol{\mu}$ as the Poisson density mean, by

$\mathbf{V}[\boldsymbol{y_i}|\mathbf{x}_i]= \boldsymbol{\mu}_i + \alpha g(\boldsymbol{\mu}_i)$ with $g(\boldsymbol{\mu})= \boldsymbol{\mu^2}$ or $\boldsymbol{\mu}$ (see below). Assuming $\boldsymbol{\mu=exp(\mathbf{x}'\beta)}$ is correctly specified, the null hypothesis is $\mathbf{H_0}$: $\alpha=0$, so that $\mathbf{V}[\boldsymbol{y_i}|\mathbf{x}_i]= \boldsymbol{\mu}_i$, implying no overdispersion, versus $\mathbf{H_1}$: $\alpha\neq0$ or $\alpha>0$. Then construct fitted values $\widehat{\mu_i}=\exp(\mathrm{x}_i'\hat{\beta})$ and use that to define the auxiliary OLS without a constant with α as the sole independent variable with u_i random error

$$\frac{(y_i-\widehat{\mu_i})^2-y_i}{\widehat{\mu_i}} = \alpha . \frac{g(\widehat{\mu_i})^2}{\widehat{\mu_i}} + \mathrm{u}_i \qquad (3.1.6)$$

The reported *t*-statistic for α is asymptotically normal under **H_0**: α=0 for no overdispersion; the same test is valid for **underdispersion H_1**: α<0, that is if variance < mean.

Alternative Model of Count Data

There are several important reasons for the desirability of employing a more flexible alternative model of count data. First, overdispersion may be due to **unobserved heterogeneity** which can be accounted for by adding a random error term to the Poisson model; such a mixture approach leads to a more flexible and widely used negative binominal model discussed below. Second, under and overdispersion may arise because the process generating the first event may differ from that determining later events, for instance, an initial visit to the doctor's surgery is the individual's choice, but subsequent visits are the physician's choice. Third, the Poisson model of independent events may be invalid; for instance, the occurrence of one doctor's visit may make subsequent visits more likely. The last two cases violate the independence assumption of the Poisson distribution. However, the model is good for the purpose of estimating the mean of the Poisson distribution for a certain event even with over-dispersion, but if the research interest requires going beyond the first moment, then an alternative model must be employed.

Negative Binominal Model

The Poisson model is a deterministic function of the explanatory variables without a stochastic error term. Allowing for **unobserved heterogeneity** by including an error term in the model to ensure the parameters are random, causes the variance of the number of occurrences to exceed their expectation, consistent with the tendency of count data to display overdispersion. A popular way to introduce unobserved heterogeneity into the Poisson model is to specify to have $\boldsymbol{\lambda}$= $\boldsymbol{\mu.v}$; then, $\boldsymbol{\lambda}$ changes randomly rather than deterministically as in the Poisson model, with a stochastic error term $\boldsymbol{v}$ >0 independently distributed as a *Gamma* distribution, and specified to have a mean of one and a variance of $\boldsymbol{\alpha}$. However, $\boldsymbol{\mu}$ is a deterministic function of **x**. By integrating $\boldsymbol{v}$ out of this specification, we obtain the **Negative Binominal** (**NB**) distribution for the number of occurrences, with a randomly determined mean $\boldsymbol{\lambda}$, and but a different variance. Assuming $\boldsymbol{\alpha}$ to be different functions of $\boldsymbol{\lambda}$ generates different types of negative binominal models.

Let the Poisson model λ to consist of two components as $\lambda = \mu . v$ where μ is a completely deterministic function of x, typically $\mu = \exp(x\beta)$, but $v > 0$ is *iid* with density $g(v \mid \alpha.)$. With this formulation, different observations may have different heterogenous λ, partly due to a random unobserved component. The expected values in this model conditional on the deterministic component is $E[\lambda|\mu] = \mu$, that is, the interpretation of the slope parameters remains the same as in the Poisson model above. However, the marginal density of y, unconditional on the random parameter v but conditional on the deterministic parameters μ and α, is obtained by integrating out v (diffentiating the integral with respect to v), resulting in

$$h(y| \mu,\ \alpha) = \int f(y| \mu,\ v).\ g\ (v \mid \alpha)\, dv \qquad (3.1.7)$$

where $g\ (v \mid \alpha)$ is called the *mixing distribution* and α the unknown parameter that defines the distribution. The integration produces an "average" distribution consisting of a mixture of two distributions. If $g\ (v)$ is specified to have the Gamma density

$\mathrm{g}\ (\mathrm{v}\,) = \frac{\mathrm{v}^{\delta-1}\mathrm{e}^{\mathrm{v}\delta}\delta^{\delta}}{\Gamma(\delta)}, \quad v, \delta > 0$ where $\Gamma(.)$ Stands for the Gamma function[6]; with $E[v]=1$ and $\mathrm{Var}[v]=1/\delta$, and $f\ (y| \lambda)$ to have the Poisson density, then we would specify (3.1.7) as the negative binominal with the mixture density given by

$$h(y| \mu,\ \alpha) = \int_0^{\infty} \frac{e^{-\mu v}\ (\mu v)^y}{y!} . \frac{v^{\delta-1} e^{v\delta} \delta^{\delta}}{\Gamma(\delta)}\, dv \qquad (3.1.8)$$

(3.1.8) can be shown to be the **NB** probability mass function obtained from a *Poisson-Gamma mixture* represented by

$$h(y| \mu,\ \alpha) = \frac{\Gamma(\alpha^{-1}+y)}{\Gamma\ (\alpha^{-1}\ \Gamma(y+1))} . (\frac{\alpha^{-1}}{\alpha^{-1}+\mu})^{\alpha^{-1}} (\frac{\mu}{\alpha^{-1}+\mu})^{y} \qquad (3.1.9)$$

Expressed in logarithmic terms, the log-likelihood of (3.1.9) can jointly estimate the parameter vector of the Poisson and the heterogeneity parameter of Gamma distribution using the standard *ML* estimator; for details see Cameron and Trivedi (2005, p.675), or Wooldridge (2010, p. 737). The first two moments of NB distribution are

[6] In general, $x \sim Ga(\alpha, \beta)$ is defined by $\Gamma(\alpha, \beta) = \frac{\alpha^{\beta}}{\Gamma(\alpha)}\ x^{\alpha-1} e^{-\beta x}$ with $E(x) = \alpha / \beta$ and $Var(x) = \alpha / \beta^{2}$.

$$\mathrm{E}(y|\,\mu,\ \alpha) = \mu \qquad (3.1.10)$$

$$\mathrm{Var}(y|\,\mu,\ \alpha) = (\mu + \alpha\,\mu^2) \qquad (3.1.11)$$

The NB variance depends on the specification of α as a function of λ, the more popular version given here is as a function quadratic in μ is known as the NB2 because that model provides very good approximation for many different count data sets, while the version with linear variance=

(λ + α λ) is known as the ***NB1*** (the *NB1* has the drawback of excluding underdispersion, see Wooldridge p. 671)**.** Hence, both types of NB allow for overdispersion. Note that with both types of NB, the density reduces to that of the Poisson as $\alpha \to 0$, demonstrating that the NB is a generalization of which the Poisson is a special case[7]. In exercise Q3.2, we examine both the Poisson and $NB2$ models applied to **a** count data set for annually visits to a physician's office by US individuals 65+; the data lacks a race indicator but we have included a gender indicator as illustration.

3.2 *Limited Dependent Variable Models*

Models with discrete dependent variables are typically binary or assume limited categories, for instance choices, between two-year, four-year public and four-year private colleges. Such models are not adequate to deal with samples that have zero dependent variable for a substantial proportion of the sample, but otherwise non-discrete, continuous dependent variable, for example, expenditure on tobacco. The models developed for this type of data are called *limited dependent variable* models and they fall into two groups. First are the models for missing data on the dependent variables for a subset of the sample, but full observations on all explanatory variables; for example, in a tobacco expenditure sample, we may have full observations of relevant explanatory variables such as age, earnings, local tax on tobacco of all the individuals in the

[7] More Generally, y is the number of random trials, is fixed, and r =r, r+1, . . . ; n is the number of trials required for a "successful" outcome, for example, the number of visits to a physician's office in a year, with probability 0 <P<1 and r>0, then n ~ NB(r, p) if r trials are required to achieve r successes. With the NB2 probability mas function is $P[y_2 = y|p, r] = \binom{y-1}{r-1} p^r (1-p)^{n-r}$, for which $r = \frac{\mu^2}{\sigma^2 - \mu}$ and $p = \frac{r}{r+\mu}$, (the 1st bracket stands for the number of combinations) of (r – 1) out of (y – 1) combinations. Driving μ and σ from r and p shows the 2nd moment for a model of count data is $\sigma^2 = \mu + \frac{1}{r}\mu^2$. Now if $r \to \infty$, then mean and variance become equal and we are back to the Poisson model that rules out overdispersion (above α=1/r).

sample; only the dependent variable is missing for non-smoking, so we have a *censored dependent variable*. Second are the models of limited dependent variables with missing data on *both* the dependent and the explanatory variables, for example, using a survey of negative income tax, consisting of individuals with income below the poverty line, to estimate an earnings equation for the entire population. This is an example of a *truncated dependent variable*. The models of truncated data are harder to implement and perform less effectively than those of censored data. Here we examine models of censured data only.

Censoring may be from above as with *top-coded* income surveys where to avoid measurement error in top income, or to preserve for anonymity, income surveys typically set income above a certain level equal to zero. This is an example of *censoring from above* (or from the right). However, a more important kind of censoring is when the dependent variable remains unchanged at zero with changes in the explanatory variables for a nontrivial portion of the population as with non-purchase in a survey of expenditure on motorcars when some with positive expenditure may report zero at the time of the survey. The zero observations in this case may be simply the result of data collection, but also the outcome of behavior; for example, in a labor supply survey, the zeros can be due to desired hours of work, actual hours of work performed, or actual work hours for employed and non-employed. Consider

$$y=x\beta + \varepsilon$$

where x is a vector of explanatory variables for the unit individual h, and ε a normally distributed error term and we only observe $y_h > 0$, in this case its values *below* zero are unknown. This is an example, in a survey of hours worked often contains a substantial proportion of zeroes for some of the survey participants. This is an example of *censored from below* (or from the left). Suppose now we drop y=0 observations and apply the OLS to the censored sample. The estimates would be biased and inconsistent because if $y > 0$, then $\varepsilon > - x\beta$ and so $E(\varepsilon \mid \varepsilon > - x\beta) \neq 0$, that is the OLS condition of orthogonality $\boldsymbol{x}$ from ε is violated; they are correlated. Moreover, inclusion of zero observations in an *OLS* regression as though they are all genuine zeros are also biased and inconsistent because the zero/nonzero events do not have equal probability of occurrence with non-trivial zero proportion; the OLS estimation does not account for the probabilities for zero and positive expenditure in the sample. To resolve the problem, in a classic paper, Tobin (1958) proposed maximization of a likelihood function consisting of the product of the purchase and non-purchase observations based on a standard normal, homoscedastic error term. For a positive purchase, the probability of purchase for each h is given by the height of the standard normal density function; for each non-purchase, the probability is given by the integral above the censored zero of the standard normal density function; that is, the area under the standard normal cumulative distribution. The distinctive feature of the resulting likelihood function is that it is a mixture of density and cumulative normal distribution functions. More specifically, the **Tobit model** (Tobin's probit) is derived from a *latent variable* model linear in regressors with a normally distributed and homoscedastic, additive error term:

$$y^{*}= x\beta + \varepsilon\,; \varepsilon \sim N(0, \sigma^{2}) \text{ and } y^{*} \sim N(x\beta, \sigma^{2}) \qquad (3.2.1)$$

the observed y is then defined with the limit observations of the likelihood function $L(.)=0$ by

$$y= \begin{Bmatrix} y^{*} & \text{if } y^{*} > 0 \\ - & \text{if } y^{*} \leq 0 \end{Bmatrix} \qquad (3.2.2)$$

where **–** stands for missing observation; no particular value is observed when $y^{*} \leq 0$. More general censoring schemes from above or both above and below, the *two-limit Tobin,* are also possible. However, normalizing the limit at zero with $L(.)=0$ is necessary for identification in a linear model with an intercept. If $L\neq0$, then $\beta_1+ \beta_2 x_2 + \varepsilon > L$ or $(\beta_1 – L)+ \beta_2 x_2 + \varepsilon = 0$, so only the difference $(\beta_1 – L)$ is identified. The application of (3.2.2) to the $f^{*}(y) \sim N(x\beta, \sigma^{2})$ censored density shows the cumulative distribution function of the latent variable y* is

$$Pr(y=0)=P\ (y^{*} \leq 0)=Pr[x\beta + \varepsilon \leq 0]=\Phi(x\beta|\sigma)=1 - \Phi(x\beta|\sigma)$$

Where $\Phi(.)$ stands for the standard cumulative distribution function (*cdf*), the last two equalities are scaled by σ for conversion to standard normal distribution, and make use of the symmetry property of the standard normal distribution. Hence, for positive observations, the term that enters the likelihood function is the normal probability density with $E(x\beta)$, and variance of σ^{2}; the full likelihood function is the product of the probabilities that the limit observations occur times the probability density functions of all non-limit observations:

$$L(\beta, \sigma^{2})=\prod \left\{1 - \Phi(\frac{x\beta}{\sigma}) \right\}^{1-d} \text{x} \prod \left\{\frac{1}{\sqrt{2}\pi\sigma}\ e^{-\,(y-x\beta\,)^{2}/\,2\sigma^{2}}\right\}^{d}$$

where the indicator d=1 if y > 0; the Tobin estimator maximizes the corresponding censored log-likelihood function correspondingly by

$$ln\mathrm{L}_N(\beta,\sigma^2)=\left\{d_i \textstyle\sum_{i=1}^{N} \left\{d_i \left(-\frac{1}{2}\ln 2\pi - \frac{1}{2}\ln\sigma^2 - \frac{1}{2\,\sigma^2}\,(y - x\beta)^2\right)\right\} + (1 - d_i)(1 - \Phi\left(\frac{x\beta}{\sigma}\right))\right\} \quad (3.2.3)$$

The estimates by (3.2.3) are a consistent and efficient; however, the results are crucially dependent on the normality and homoskedasticity assumptions of the model; violations of either assumption would lead to inconsistent estimates. For example, with heteroskedastic errors, the Tobit estimates are inconsistent because E(d_i)= $\Phi\left(\frac{x_j'\beta}{\sigma_j}\right)$; d_i=1, 2, only when $\sigma_j^2 = \sigma^2$. It is possible to correct for heteroskedasticity by weighted least squares if we know its form. Dependence of the consistency of the Tobin estimator on the absence of minor deviation from homoskedasticity can be better understood in comparison with the OLS. For the latter, the mean and the variance of the error term are independent of each other. Therefore, heteroskedastic errors affect only the efficiency of the efficiency of the estimates, not its unbiasedness. By contract, the mean and the variance of the Tobit model are no longer independent of each other. Heteroskedasticity has the further consequence that the estimates are inconsistent. In addition, evidence suggest that bias due to non-normality in censored data can be substantial. In any case, given the fragility of the Tobit model, we should test for its distributional specification. We test by nesting the Tobit within a richer parametric model and apply a Wald, LR, or LM test to the restrictions. The LM test is particularly simple for testing against heteroskedasticity of the form $\sigma_j^2 = exp(\boldsymbol{x\beta})$ in the censored regression. The LM test is based on the unadjusted R^2 from an auxiliary regression, second stage regression; when multiplied by the sample size N, [N* R^2], the test has a χ^2 distribution. The LR is based on the likelihood estimates from both the restricted and unrestricted models and works like an F-test.

Interpreting the Tobit estimates

The slopes of the Tobit latent variable model are $\widehat{\beta}$ but we are also interested in the marginal effect of a change in x on either the regression function of the observed data $E(y|\mathbf{x})$, inclusive of zeros, or the regression function conditional on positive observations y>0 , E(y|$\boldsymbol{x}$, y>0). The slope is relatively simple, expressed as the product of the parameter estimate by a scale factor:

$$\frac{\partial(y|x)}{\partial x} = \beta.\Phi\left(\frac{\beta X}{\sigma}\right) \qquad (3.2.4)$$

(3.2.4) suggests the means for a rough comparison of OLS and Tobit estimates because the LH of it is equal to the OLS slope. Therefore, to turn this into an approximate Tobit, we must multiply it by $\widehat{\beta}$. Because Φ(.) values are positive, the sign of the estimated slope identifies the direction of the marginal effect but its magnitude depends on both the coefficient and the *cdf*. If β > 0, then as x increases, the *cdf* approaches one, and the slope of the regression approaches that of the latent variable model. There are two methods for obtaining the adjustment factor; both based on (3.2.4). First method involves computing the marginal effect at the average (PEA), that is evaluating $\Phi(.)$ at $\Phi(\beta'\bar{X})/\sigma)$, and then multiply $\widehat{\beta}$ by this scaling factor. However, the average may be of little value if the interest is obtaining some other percentile value or another central value. The second is to use the average partial estimate $n^{-1}\sum_{i=1}^{n}\Phi\left(\frac{\widehat{\beta}'X}{\hat{\sigma}}\right)$, a factor with values that always fall between zero and one. In fact, if y=0 observations are few, APE and AEP will both be close to one; and for y>0, the Tobit and OLS estimates will be identical. However, there is also a simple, OLS-based alternative to the Tobit estimator that offers approximation for the Tobit *MLE* based on a remarkable result of Greene (1981), and valid for many classes of Limdep models; namely, that all OLS slope estimates of censored survey (except the intercept) are biased downward in the *same proportion*. Consequently, the division of the OLS estimates by this factor of proportionality corrects the OLS estimates of censored data; the scale factor is approximated by the proportion of *nonlimit* (continuous) observations in the sample. Let θ = n_1/n by such a nonlimit proportion, then $\widehat{\beta}$= β_{OLS}/θ are consistent.

The Tobit marginal effect can be broken down into a part due to a change in x for the proportion of the population whose *y*-data is already observed, and that from changes in the proportion of the population who switch from *y*=0 to *y*>0 as *x* changes, for example changes in labor supply. Formally the two components of a marginal change can be written as

$$\frac{\partial E(y|x)}{\partial x} = Pr(Y>0)\,.\frac{\partial E(y|x,y>0)}{\partial x} + \ Pr\ (E(y|x,y>0).\frac{\partial \Pr\ (y>0)}{\partial x}$$

This is known as the **McDonald-Moffit decomposition** of the Tobit marginal effect.

Two-step Tobit

(3.2.3) restricts the censoring mechanism to be from the same model that generates the positive outcome variable. The case for separating the two processes is strong when we judge that certain values occur in frequencies inconsistent with a simpler, one-step Tobit model. A Tobit model that allows for the zero and nonzero to be generated by different densities provides more flexibility; for instance, we may have one equation for probability of hospitalization, and another for care expenses once admitted. Define a dummy variable d=1 for participants who have observed y > 0 , and d=0 for nonparticipants who have y=0. Then a two-part Tobit model is given by

$$f\,(y|\,X)=\left\{\begin{matrix}\Pr\,[d=0|\boldsymbol{x}) & \text{if } y=0\\ \Pr[d=1|x]\,f(y)d=1,\boldsymbol{x}) & \text{if } y>0\end{matrix}\right\}$$

This is a generalization of a one-step Tobit with a probit (or logit) model, which is an obvious choice for the decision *d* participation, using a positive value random variable such as log-normal. The same explanatory variables may appear in both equations (on the merits of this approach see the sample selectivity model examined below).

3.3 Sample selectivity Models

The OLS model of consistent estimation is based on the sample being randomly selected; whenever a sample is in part determined by the values of the dependent variable, the estimates will be inconsistent since the sample is no longer randomly selected. As with the Tobit estimator, the key issue is whether selection is based on endogenous variables; the application of OLS to a sample determined by an exogenous variable are consistent since if we start with a random sample and randomly drop observations, OLS will still be consistent. Selection based on an endogenous variable may be **self-selection** as participants may choose not to participate in the sample activity such as supplying labor to the market, or the sample may over-represent those chosen to participate. In either case, such samples are not random and consistent estimation requires regression models capable of correcting for the sample selection bias. Consider for example an earning equation as a function of the individual's age, education, experience, etc. There are, however, different possible mechanisms that transmit the effects of market entry to the wage equation. For instance, more capable workers are also more likely to be observed in the labor market *and* earn higher wages; if ability is unobservable, then its exclusion from estimation, as a repressor leads to biased estimates. This bias will be more pronounced if there are more zero observations for entry than is consistent with the wage model; in a typical labor supply survey, zero wage observations are many times larger than observations with positive wage values. The sample *selectivity bias* is an econometric explanation initially proposed for the implausible finding that women with small children appeared to have higher wages than those without. The former group had a higher shadow price of time and so a higher reservation wage, thus requiring a higher observed wage rate before they are observed entering the market. In a regression of wage equation with a sample of working women, the error term, for such households will be higher, incorporating unobserved higher reservation wages. The inclusion of this endogenous influence in the regression results in misspecification error, making the parameter estimates biased. The solution proposed to correct for the bias is: to compute the expected values of the error term, and use it as an additional explanatory variable in the wage equation. If one employs two equations for observed and shadow wages, together with an equation for an endogenous 'amount of work' variable which equates the two wage rates, then this amount is positive when the person is observed to trade in the labour market, and non-positive when she is not. Heckman (1974) estimated a system of simultaneous equations for this model by the ML method to obtain the expected value of the error term and correct for estimation bias. However, in a well-known paper, Heckman (1976) suggested a simpler two-stage estimation when the maximization of the likelihood function for selectivity bias proves cumbersome. This is the most common selectivity model used and discussed below.

In the generalized version of this two-stage approach, behaviour is described by two regression equations, plus an equation for an endogenous dummy dependent variable describing the selection of the households into one of the two regimes, the error term of the latter assumed correlated with those of the two regimes.

$$y_h = \beta_1' X_{1h} + u_{1h} \; if \; \gamma' Z_h \geq u_h \qquad (3.3.1)$$

$$y_h = \beta_2' X_{1h} + u_{2h} \; if \; \gamma' Z_h < u_h \qquad (3.3.2)$$

such that

$$d_h = 1 \; if \; \gamma' Z_h + u_h > 0 \qquad (3.3.3)$$

$$d_h = 0 \; otherwise \qquad (3.3.4)$$

It is further assumed that the three error terms are jointly normal, and hence expectation of each, conditional on the other, is linear, and that $\sigma = 1$ (since otherwise γ is estimable only up to a scalar factor of σ). Subscripts 1 and 2 indicate which of the two regimes or equations an individual observation on y_h belongs to. The expectations $y_h$in (3.3.1) and (3.3.2) are given by

$$E(y_h | \gamma' Z_h \geq u_h) = \beta_1' X_{1h} + \sigma_{1h} \lambda_{1h} \qquad (3.3.5)$$

$$E(y_h | \gamma' Z_h < u_h) = \beta_2' X_{2h} + \sigma_{2h} \lambda_{2h} \qquad (3.3.6)$$

where σ_{1h} and σ_{2h} are covariances between u_{1h} and u_{2h} with u_h. Given normality, Heckman (1976) shows that the last variables in (3.3.5) and (3.3.6) assume a particular form (depending on whether the truncation is from below or from above). This is,

$\lambda_{1h} = -\dfrac{\phi(\gamma' Z_h)}{\Phi(\gamma' Z_h)}$, $\lambda_{2h} = \dfrac{\phi(\gamma' Z_h)}{1-\Phi(\gamma' Z_h)}$; $\phi(.), \Phi(.)$ are the standard normal density and cumulative distribution functions (where use has been made of the conditional expectation for a truncated normal distribution[8]). λ_{1h} is known as the *inverse of the Mills ratio* and λ_{2h} as its complement. The parameters of interest for the selectivity test are σ_{1h} and σ_{2h}. It is, however, more common to estimate (3.3.5) and (3.3.6) as a single equation using all the observations on y_h.

Noting $p(\gamma' Z_h \geq u_h) = \Phi(\gamma' Z_h)$; $p(\gamma' Z_h < u_h) = 1 - \Phi(\gamma' Z_h)$, we have

$$E(y_h) = E(y_h|\gamma' Z_h \geq u_h).p(\gamma' Z_h \geq u_h) + E(y_h|\gamma' Z_h < u_h).p(\gamma' Z_h < u_h)$$

$$= \beta_1' X_{1h}\Phi_h + \beta_2' X_{2h}(1-\Phi_h) + (\sigma_{2u} - \sigma_{1u})\phi_h + \varepsilon_{1h} \qquad (3.3.7)$$

where conditional expectations given in (3.3.5) and (3.3.6) are substituted to obtain (3.3.7) whose error term now has the property that $E(\varepsilon_{1h}) = 0$. The interest in the selectivity bias test centres on the combined parameter $(\sigma_{2u} - \sigma_{1u})$, conveniently avoiding separate estimates for σ_{1u} and σ_{2u}. A probit analysis over the entire sample in the first stage provides an estimate of γ, allowing computation of the Mills ratio $\dfrac{\phi(\,.\,)}{\Phi(\,.\,)}$. This is then employed as an additional regressor in the second stage for a single equation, once again combining the observations on both sub-samples, estimated by OLS. Selectivity bias exists if $(\sigma_{2u} - \sigma_{1u}) \neq 0$; otherwise, we fail to reject the hypothesis that selectivity bias exists. Note that in all versions of the selectivity bias models, the computation of Mills' ratio requires observations on the explanatory variables for which the corresponding dependent variable has a zero limit, so the procedure is only applicable with censored data; not relevant when the sample is truncated.

In the (3.3.5)-(3.3.6) model, the vectors of X_{1h} and X_{2h} are not necessarily identical and can contain variables exclusive to each. If this is not the case, then $X_h = X_{1h} = X_{2h}$ & (3.3.7) simplifies to

$$E(y_h) = \beta_2' X_h + (\beta_1' - \beta_2') X_h \Phi(\Upsilon' Z_h) + \beta_2' X_{2h}(1-\Phi_h) + (\sigma_{2u} - \sigma_{1u})\phi_h + \varepsilon_h \qquad (3.3.8)$$

Here the only change in the function is through the scaling effect of the intercept, see Heckman (1990) on such varieties of selectivity bias.

[8] For $z \sim N(0, 1)$, $E[z|z>c] = \phi(c) / 1 - \Phi(c)$; $\Phi(c)$ measures probability by the area in the standard normal Φ to the left of c, $\Pr[z \leq c]$, hence $\Pr[z>c]=1-\Phi(c)$ for a constant c. In this case, $\phi(-c) = \phi(c)$, and $1-\Phi(-c) = \Phi(c)$, hence $E[z|z> -c] = \phi(c) / \Phi(c)$; see Maddala (1988, Appendix)for details.

If one were to test selectivity with $y_h > 0$ observations only, but still retain the binary function $d_h(.)$, then a probit analysis on $d_h(.)$ provides estimation of γ and hence computation of inverse Mills' ratio, λ_h, which is in turn used in the second stage as an additional variable in

$$y_h = \beta' X_h + \sigma \lambda_h + v_h \; ; \; y_h > 0 \,\&\, E(v_h) = 0 \qquad (3.3.9)$$

Identification

The above mode will be poorly identified if an identical vector of variables is employed in both entry and in wage equations; therefore, the simplifying assumption that $x_1 = x_2$ is far more critical for the selectivity model than for the two-step model with all independent variables fully observable. This results from $x_1 = x_2$, $\lambda(.)$ in (3.3.9) which would be highly collinear with another set of explanatory variables in that equation, namely x_2. Given $x_1 = x_2$, it would be very likely that point estimates obtained for (3.3.1)-(3.3.3) will have larger standard errors. This highlights the need for additional variation *unique* to $x_1\beta_1$ in the probit equation (3.3.1) for more accurate estimates. There is therefore, a critical role for employment of one or more *exclusion restrictions* on variables that are included in the probit equation but excluded from the wage equation. With $x_1 \neq x_2$, exclusion restrictions will introduce greater variability across $x_1\beta_1$ observations to enable the model to tell apart the effects of participants and nonparticipants.

Of course, even with $x_1 = x_2$, we still have some variation unique to the probit model of entry by the nature of its nonlinear functional form and in contrast to the log linearity of the wage equation. With this type of exclusion restriction, parameter identification relies solely on functional form restriction. If, however, non-linearity is limited rather than pronounced (both equations are similarly close to linearity), then identification by functional form would be rather poor; therefore, effective selectivity bias test and control often rely on the critical role played by at least one exclusion variable blocked from the wage equation and unique to the entry probit equation, for example, fixed costs of entering the labor market such as time or distance to and from work, or age and age squared.

Readings

For textbook discussion, see Cameron and Trivedi (2005, chapters 14, 16, and 20); Wooldridge (2010, chapters 17, 18, and 19). Tobin (1958) and Heckman (1976) are the classics on the Tobin and Heckman models; for an application of count data models, see Cameron *et. al.* (1988); for that of the Heckman model, Koohi-Kamali (2021).

Chapter 3 Discrete Dependent Variables Exercises

Q 3.1

a. For estimating the mean of a nonnegative random variable y, the Poisson quasi-log likelihood for a random draw is

$$\ell_i(\mu) = y_i \log(\mu) - \mu, \qquad \mu>0$$

$$\mathrm{E}[\ell_i(\mu)] = \mu_0 \log(\mu) - \mu.$$

Show that this function is uniquely maximized at μ=μ_0.

b. The gamma (exponential) quasi-log likelihood is

$$\ell_i(\mu) = -\frac{y_i}{\mu} - log(\mu) \qquad \mu>0$$

Show that E[$\ell_i(\mu)$] is uniquely maximized at μ=μ_0.

Q 3.2 Download *mus17data.dta* on the annual number of visits to physician's office*;* define a global variable for the covariates of *docvis*: *private medicaid age age2 educyr actlim totchr female*.

a. Estimate a Poisson model of *docvis* regressed on the global list, and obtain the squared correlation coefficient between the fitted and observed values of the dependent variable.

b. Estimate the model in a. by *QMLE* and test for overdispersion.

c. Obtain the marginal effects of the explanatory variables in a.

d. Estimate the equation in **a.** by *NB2*, employing *ML* and *QML* estimators.

Q 3.3 Download *tobin.dta*, regress the Tobin model of *aptitude* on the variables *read* and *math* with lower (left) censuring; comment on the outcome.

Q 3.4 Download *mroz.dta*. Regress the Hackman model with an indicator for women's choice to work or not as a function of *age, education, married and children,* and for earnings as function of *age & education*. What variable in the output file controls for the selectivity bias?

Chapter 4 *Analysis of Panel Data*

Introduction

Suppose we want to estimate a consumption function, with N=4000 individuals in each of four time periods, T=4. If we regress consumption y on income x by OLS by drawing a line through the data points, a slope estimate is shown by the line *AA* in the Figure 4.1 below. Now we can also identify data by the cross-section unit for each individual over the four-year period; in this note I use "individuals" as a generic term also to cover families, firms, countries, or regions, depending on the unit of observation. The figure below identifies four such units, out of the 4000 observations, each observed four times, by drawing an ellipse around each individual. The sample contains many more ellipses, roughly divided in equal number above and below *AA*. These individuals have different intercepts (the point at which their consumption function curve meets the y-axis). Note, however, that the lines through the four data points for each individual are parallel to *AA* and to each other, that is, the slope remains unchanged. These different intercepts reflect the effects of many influences that account for individual uniqueness, or cross-sectional ***heterogeneity***. The *OLS* estimation by the line *AA* is biased since it ignores heterogeneity unless the omitted influences are uncorrelated with the included independent variables. Note that this also implies that the error term no longer has a constant variance since it varies **within individual clusters** over time.

4.1 *Robust time-invariant estimators*

There are two ways to deal with the effects of heterogeneity. The first is to create a separate dummy variable for each cross-sectional unit, and estimate the equation with OLS. This **dummy *fixed effects*** estimator, however, is only possible with a small number of cross-sectional units; with 4000 such units, we need 3999 dummies, a massive loss of degree of freedom! An alternative method that gives the same estimate is to transform the equation so that ***time-invariant*** individual characteristics are eliminated. This outcome can be achieved by two different estimators, and both give unbiased estimates but both also remove all independent variables that remain unchanged over-time within the unit, namely, gender or race; not helpful if we wish to estimate their impact on the dependent variable. As illustrated by Figure 4.1, within each ellipse, such values remain unchanged so their differences from their averages are all zero.

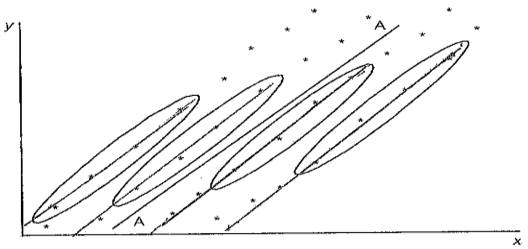


Figure 4.1 *fixed effects estimator with intercept control excluded variables uncorrelated with x*

The second method, called the ***random effects*** (***RE***) model, is designed to avoid the above shortcoming of the fixed effects model while allowing for changes in the intercepts across units. However, the random effects by different intercepts have a new meaning as though they are drawn by random sampling; different intercepts are now random and therefore, treated as a part of a composite error term, so that term has two parts, a time-invariant error and the usual error changing with time. Since there may well be correlation between the error terms due to their common time-invariant component, we have to transform the original model so as to eliminate the error correlation, similar to the strategy we followed to deal with serial correlation. Just like the solution to serial correlation, the transform equation will be nonlinear in parameter and must be estimated by *GLS* to produce efficient estimates. Hence, the *RE* estimator retains time-invariants and is more efficient than the *FE* or *FD* estimators, see below.

So far, we assumed that problem with the compound error term is the correlation between the errors. If, however, the unobservable time-invariant error component is also correlated with included explanatory variables, for example, unobserved intelligence may well be correlated with completed years of education, then the estimates will be biased even if the transformed equation estimates are efficient. The following Figure 4.2 explains this important drawback of the *RE* estimator; the graph similar to the previous one for the *FE* model but with a notable difference in that the common slope, line *AA*, is no long the same as the individual slopes. The main reason for this outcome is the larger intercept for an individual, the larger is the individual's x, namely, higher x cut the y-asis at higher values; hence the *OLS* overestimates the common slope (more steep than individual slopes). The increase in y values is caused for two reasons. First, because of higher x values; second because of higher intercept. The *OLS* attributes both these changes to x. As a result, there may be correlation between the composite error term and the explanatory variables, leading to an upward bias in slope estimates. It is therefore critical to employ the *RE* estimator only when the evidence suggests its error term is not corelated to the equation regressors. In that case, the random effects model should not be used. The ***Hausman test*** is designed to determine if we are justified employing the random model or whether we should employ the safer unbiased fixed effect method without separate estimates for observable time-invariants such as religion.

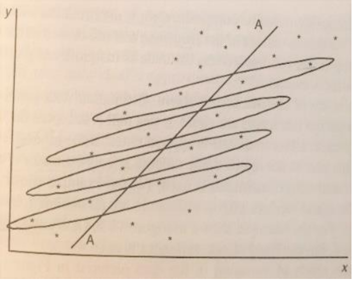


Figure 4. 2 *Random effects estimator with a positive correlation between x and the intercept*.

4.2 *Fixed Effects Model*

The first model that remove the time-invariant unobservables is called the *Fixed effects* model. Start with the constant coefficient assumption of the OLS model, namely, line *AA* above. For individuals *i=1,2,…N*, and periods *t=1,2, …T*, we have

$y_{it} = \beta_1 + \beta_2 x_{2it} + \beta_3 x_{3it} + e_{it}$, *iid* e_{it} with E($e_{it} | \beta_1, x_{2it}, x_{3it}$)=0, for *t*=1, 2 (4.2.1)

Suppose now we abandon the assumption of constant coefficients across individuals and rewrite (4.2.1) with changing coefficients across individuals *i* as

$$y_{it} = \beta_{1i} + \beta_{2i} x_{2it} + \beta_{3i} x_{3it} + e_{it} \qquad (4.2.2)$$

(4.2.2) is hard to estimates when the panel is short and wide, namely, when we have large cross-section samples of individuals followed over a limited number of periods. This is because all the variation in (4.2.2) comes from changes over time, so with say five period data, we only have five observations to estimate the three parameters in (4.2.2), and the results are likely to be very imprecise; typically (4.2.2) requires more than five coefficients to estimate, making the task impossible. To proceed with the estimation of (4.2.2) we need to simplify, and a common simplification is to assume that differences between individuals result in changes in the intercept across individuals alone but their slop coefficient remain unchanged (as in the above Figure 4.1):

$$y_{it} = \beta_{1i} + \beta_2 x_{2it} + \beta_3 x_{3it} + e_{it} \qquad (4.2.3)$$

(4.2.3) is a **Fixed Effects model** because it controls for within-individual time-invariant effects, or **individual heterogeneity,** by allowing the **intercept fixed effects** for the same individuals in different time periods, for example the same county or the same household. This model can be estimated by two different methods: by **the Least Squares Dummy Estimator** if the number of individuals is small, for example, the number of research universities in Philadelphia, and by the **Fixed Effects Estimator** applicable when the number is large, for example, data on the UN recognized list of countries.

When number of cross-sectional individuals is not small, we must rely on the alternative fixed effects estimator. Start again with the data on individual *i* over t=1,…T periods as shown in (4.2.3) and obtain the average value for each variable *over time*, still assuming no change in coefficient across individuals, only those within-individuals. The "-" over each variable indicates its averaged value over time.

$$\bar{y}_i = \beta_{1i} + \beta_2 \bar{x}_{2i} + \beta_3 \bar{x}_{3i} + \bar{e}_i \qquad (4.2.4)$$

Then subtract (4.2.4) from (4.2.3), and write the new variables, defined as *deviations from the mean*, with a "~" hat

$$\tilde{y}_{it} = \beta_2 \tilde{x}_{2it} + \beta_3 \tilde{x}_{3it} + \tilde{e}_{it} \qquad (4.2.5)$$

Where $\tilde{y}_{it} = (y_{it} - \bar{y}_i)$; $\tilde{x}_{2it} = (x_{2it} - \bar{x}_{2i})$; $\tilde{x}_{3it} = (x_{3it} - \bar{x}_{3i})$; & $\tilde{e}_{it} = (e_{it} - \bar{e}_i)$.

(4.2.5) is the **Fixed Effects (FE) Estimator**, also called the **Within Estimator** because variation with each cross-sectional unit over time is controlled; the estimator has several notable features. First, the least squares estimates of (4.2.5) are unbiased and consistent in small and large samples, and produce identical estimates to the Least Squares Dummy Fixed Effects Estimator without the need to include all the dummy variables, resulting in a big gain in degree of freedom. Moreover, the two methods produce the same least squares residuals. The most attractive feature of (4.2.5) is that ***unobservable time-invariant variables***, those that are constant for each individual, **are eliminated**. Since the unobservable time-invariant effects are the source of the correlation of error terms over time, by definition, they become a part of the error term; their removal by (4.2.5) prevents them from affecting the included coefficient estimates. Finally, if we assume the error term across individuals are uncorrelated, reasonable if they are randomly selected, then we can correct for this type of non-constant error variance by the application of the White, or the Newey-West methods examined earlier to obtain **panel-robust standard errors**. However, this comes with a cost since taking deviation from the mean across variables eliminates not just the unobservable effects, but **all** time-invariant effects such as race or gender. This is because the deviations from the mean for variables constant over time would all be zero. If estimates are required for the effects of such observable time-invariant variables, then we cannot use the Fixed Effects model.

4.3 *First-Differenced Estimator*

The second alternative called the **First Differenced Estimator** (***FD***), removes the endogenous time-invariant unobservables in a panel data series by first differencing all the variables; since the time-invariant variables are the same in both period, differencing wipes out all cross-sectional endogeneity from the model.

$$y_{it} - y_{it-1} = \beta_2(x_{2it} - x_{2it-1}) + \beta_3(x_{3it} - x_{3it-1}) + (e_{it} - e_{it-1}) \qquad (4.3.1)$$

Both *FE* and *FD* account for individual heterogeneity through an intercept that changes cross-sectionally; both estimators are valid based on the assumption of **strict, or strong exogeneity**:

$$E(e_{it} \mid x_{2it}, x_{3it}) = 0,\ t = \text{I}, 2, \ldots T$$

For the *FE* estimator. We can also state this assumption for the differenced estimator as

$$E[(x_{jit} - x_{jit-1})(e_{it} - e_{it-1})] = 0,\ j = 1$$

To understand the implications of strict exogeneity, simplify the subscripts by highlighting the current as 2 and lagged as 1, hence

$$E[(x_2 - x_1)(e_2 - e_1)] = \text{E}(x_2{}'e_2) + \text{E}(x_1{}'e_1) - \text{E}(x_1{}'e_2) - \text{E}(x_2{}'e_1) = 0$$

The first two terms in the above are zero by the orthogonality condition of the explanatory variables and the error term from the *same* period secured by the iid error assumption. However, that is not enough to ensure consistency in this panel context; we must also assume that the orthogonality condition between the explanatory variables and the error terms also holds when they are from *different* periods, that is, between (x_1 *&* e_2); (x_2 *&* e_1). While it is reasonable to expect the latter outcome to hold, it is important to be aware that it does not follow from the standard orthogonal error assumption, also Cameron &Trivedi (2005, pp749-50).

Since both models have consistent estimates, the question is how we should choose between them? The choice depends on the number of time period data availability and on the presence of lagged dependent explanatory variables. If T=2, the *FD* and *FE* produce identical estimates and test statistics regardless of which one we employ (see exercise question 4.1). However, if $T > 2$ and large N (narrow and wide panels), the *FE* and *FD* estimators differ; since both are unbiased, the choice between them is made on the grounds of efficiency. Since unobserved cross-sectional effects are typically serially uncorrelated, the *FE* estimator is more efficient without serial correlation and more often employed. However, no serial correlation could be a false assumption; for instance, if e_{it} follows a random walk path, there will be a substantial positive serial correlation in e_{it}. Then differencing will remove any first-order serial correlation and the *FD* estimator is a more efficient choice. Moreover, if the model contains lagged dependent variables, that is if the model is *dynami*c, then the assumption of orthogonality in different time periods, that is strict exogeneity, is violated, and although both estimators are unbiased, the *FE* estimator has much less bias than the *FD* estimator because the bias in the former does not depend on T but that of the latter tends to zero at the rate of $1/T$. We shall examine both estimators in the context of dynamic panels later. Another difference between the two estimator emerges when panel of T is large relative to N. Panels with large T may display spurious co-movement, or be non-cointegrated. The *FD* estimator has the advantage of converting an non-integrated series into a weakly dependent process by the application of first differences based on the central limit theorem when $T > N$. We shall discuss this approach in the context of long and narrow panels later in chapter 14. Generally, if the estimators give very different results, it is best to report two sets of estimates and test statistics.

4.4 *Random Effects (RE) model*.

The Fixed Effects Estimator captures individual differences by including individual-specific intercepts by β_{1i} that are fixed over time. The random effects model extends that notion by acknowledging that sample individuals are randomly selected, so such individual time-invariant differences are random rather than fixed. The model implements this notion by a break-down of the fixed intercepts with specification that β_{1i} in (4.2.3) consists of a fixed component representing the population average, $\overline{\beta}_1$, and random individual differences from that average, u_i.

$$\beta_{1i} = \overline{\beta}_1 + u_i \qquad (4.4.1)$$

We make the usual assumptions about the individual random effects: it has zero expectation and constant variance. Substitute (4.2.1) into (4.2.3)

$$y_{it} = \left(\overline{\beta}_1 + u_i\right) + \beta_2 x_{2it} + \beta_3 x_{3it} + e_{it} = \overline{\beta}_1 + \beta_2 x_{2it} + \beta_3 x_{3it} + v_{it};\ v_{it} = u_i + e_{it} \quad (4.4.2)$$

The combined error term in (4.4.2) is now composed of two components, the usual regression random effects, e_{it} , and the random individual effects u_i. (4.4.2) is also known as an **error components model**. The errors v_{it} are correlated over time for each individual *i*, but uncorrelated otherwise. The correlation is the result of the common component u_i to all time periods v_{it}. This correlation equals the proportion of the variance in the total error term v_{it} caused by the variance of the individual error component u_i , that is

$$\rho = corr\,(v_{it}, v_{is}) = \frac{\sigma_u^2}{\sigma_u^2 + \sigma_e^2} \quad t \neq s \qquad (4.4.3)$$

If the RE estimator is applied when the residual is heteroskedastic or serially correlated, typical with AR models), the a FGLS estimator can be employed in first differences after weighing the averages by an estimator $\widehat{\theta_i}$ that approaches unity as the RE estimator gets closer to FE estimator. $\widehat{\theta_i}$ provides the correction of inefficient errors by a key OLS transformation consistent estimate of

$$\theta_i = 1 - \sqrt{\sigma_\varepsilon^2}/(T_i\sigma_u^2 + \sigma_\epsilon^2) \qquad (4.4.4)$$

Testing for random effects. The size of estimated ρ determines the presence of the random effects in the sample. If there is no individual heterogeneity if u_i =0; then σ_u^2 = 0 also, so will the correlation from (4.4.4): ρ = 0. This suggests testing the null for the correlation as

$$H_O:\ \sigma_u^2 = 0 \text{ vs. } H_A:\ \sigma_u^2 > 0$$

Rejection of σ_u^2 = 0 means there are random effects among sample individuals. The Lagrange multiplier (*LM*) test statistic, based on the restricted model of assuming the null is true, tests for the presence of random effects. A positive *LM* test outcome suggests the presence random effects.

4.5 *Fixed Effects Vs. Random Effects*

If the random error $v_{it} = u_i + e_{it}$ is correlated with any right-hand variable in (4.4.2), then the random effects (GLS) estimates will be biased and inconsistent; for example a person's ability, an *unobservable*, correlated with an explanatory variable for skill in a wage equation, will be included in u_i . In this case the Fixed Effects model will remain unbiased and consistent even in presence of correlated error terms because the fixed effects transformation by (4.2.5) removes all time-invariant effects, including the random effects of u_i . The Hausman test is designed to decide between the fixed *vs.* random models by comparing their common coefficient estimates. The test is based on the idea that if there is no correlation between u_i and the explanatory variables, both models will be unbiased and consistent, thus both converge on the true parameter values. Therefore, in large samples, the two estimators should be similar. If, on the other hand, there is correlation between u_i and the explanatory variables in (4.4.2), the Fixed Effects model will converge on the true parameter values, but the Random Effects converge on some other values, and the two estimators will be different. This idea can be tested for each single parameter by the difference between the t-ratios obtained from each model. The Hausman test, however, is a joint parameter test comparing **all** the coefficient estimates from the two models, except the intercepts, to decide how close the joint differences between the two sets parameters are to zero. The rejection of the null hypothesis suggests using the fixed rather than random effects model.

4.6 *IV for Biased Random Effects Estimates*.

If separate estimates for time-invariant effects are required in presence of the correlation between u_i and the explanatory variables, then an instrumental variables estimator called the ***Hausman-Taylor estimator*** should be applied to the random model to overcome the problem of inconsistent parameter estimates if the hypothesis of equal coefficient estimates is rejected. This estimator works *if the number of exogenous time-varying variables is greater than or equal to the number of endogenous time-invariant variables*. To check this, divide the variables into four groups: endogenous time-varying and time-invariant; exogenous time-varying and time-invariant. Since the fixed effects transformation in (4.2.5), $\tilde{x}_{it,endo}$ = ($x_{it,endo} - \bar{x}_{i,endo}$), removes the correlation between u_i and the explanatory variables, $\tilde{x}_{it,endo}$ is a suitable instrument for time-invariants of $x_{it,endo}$. Another suitable instrument for time-invariant $x_{i,endo}$ is $\bar{x}_{i,exog}$. Note that the gain from having consistent IV estimators must be sufficiently large to justify the increased variance associated with the use the IV estimator; first stage t-ratio for the instrument as evidence of its strength is informative in this regard and should be regularly reported.

We conclude this chapter with an example that compares the performance of the classical and Bayesian panel data logit estimators as a further exercise about some defining aspects of the two approaches to econometrics raised in section 1.5. Table 4.1 shows the classical and Bayesian random effects panel data estimates from a Bangladesh panel data survey using a binary variable for womens' use of contraceptive as a function of age, urban-rural residency and the number of children. Although the Bayesian panel data estimator can be applied as both fixed and random effects estimators, we confine the comparison here to the panel data random effects model because that model can also account for intra-cluster correlation in the same cluster because of shared cluster-level random effects. The results are presented in table 4.1; the simulation method employed in this application is the Gibbs sampler MCMC, see chapter 18.

Table 4.1 Classical and Bayesian logistic regressions of contraceptive use in Bangladesh (1989)

C_use	Classical logistic regression	Bayesian logistic regression*
Urban	0.7323 (0.1195)	0.7364 (0.1121)
Age	-0.0265 (0.0079)	-0.0263 (0.0076)
1 child	1.1160 (0.1581)	1.1293 (1531)
2 children	1.3659 (0.1747)	1.3681 (0.1679)
3 children	1.3440 (0.1797)	1.3404 (0.1774)
Constant	-1.6893 (0.1478)	-1.6889 (0.1481)

*Prior distributions: slope parameters normal (0, 100); variance inverse Gamma (0.01, 0.01) simulated by *MCMC-Gibbs* sampler method; binary C_use=1 for a yes response.

We note that the priors employed for this example are zero-mean normal prior for the parameters, and a noninformative prior inverse gamma, using the MCMC-Gibbs sampler. The Bayesian parameters estimates and variance estimates very close, evidently the priors are non-informative. Exercise question 4.3 asks you to reproduce the outcome in table 4.1 and comment on the outcome.

Readings

For textbook discussion, see Pesaran (2015, chapter 26), Cameron and Trivedi (2005, chapter 21); Wooldridge (2010, chapter 10). Nerlove (2002) reviews the evolution of panel data analysis.

Chapter 4 Analysis of Panel data Exercises

Q 4.1 Let $\hat{\beta}_{FE}$ & $\hat{\beta}_{FD}$ denote the fixed effects and first-differenced estimators, respectively.

a. Show the FE and FD estimators are numerically identical.

b. Show that the variance matrix estimates from the FE and FD methods are numerically identical.

Q 4.2 Download *mus08psidextract.dta*.

a. Fit a Fixed Effects model for log of wage, *lwage*, as a function of *exp, exp2, ed, and wks* .

b. Fit the Random Effects model for log of wage, *lwage*, as a function of *exp, exp2, ed, and wks*.

c. Test the Fixed Effects model against the Random Effects model in *a. & b.* for endogeneity

d. Fit a differenced model to the above data set, compare the outcome with that in *a.*

e. Apply the Hausman-Taylor IV estimator to *lwage* panel to control for potential endogeneity.

Q4.3 Download *bangladesh.dta*, a subsample of data from the 1989 survey of polled 1,934 Bangladesh women on their use of contraception.

a. Fit a standard panel data random effects logistic regression for each district and comment on the outcome (random-effects estimator is more useful for modeling intracluster correlation, when observations in the same cluster correlate because they share common cluster-level random effects).

b. Fit a two-level random-intercept model *bayemh* with the corresponding random-effects parameters assigned a zero-mean normal prior distribution; apply a relatively weak normal (0, 100) prior for urban, age, children and the constant. Moreover, assign a noninformative prior *igamma* (0.01, 0.01) for the variance parameter, using Gibbs sampler.

c. Compare the outcomes in a & b, and commend on the Bayesian random effects convergence.

Chapter 5 Dynamic Panel Data Models

Introduction

Modeling variables that changes over time requires taking account of time lags that often includes the effect of their own past lags the drives the current values. A model of a time-series based, *inter alia*, on its own lagged values is called a **dynamic** model. The autoregressive distributed lag (ARDL) model, discussed in chapters 9 and 14, provides a general method for the inclusion of a lagged dependent variable that results in a more parsimonious model with improved forecasting performance because it includes lagged variables of both the dependent variable and other explanatory variables. However, the extension of ARDL to dynamic models of panel data analysis encounters an important problem that does not exist in the non-panel linear context. Here we examine this problem where a T panel series is fixed or "short" in T, that is when $N>T$.

5.1 *Time-variable panel endogeneity*

Consider

$$y_{it} = \alpha_i + \gamma y_{it-1} + x_{it}\beta + \varepsilon_{it} \;;\; i=1, 2, \ldots N, t=1, 2, \ldots, T \qquad (5.1.1)$$

where the autocorrelation coefficient $\gamma < 1$ *to* ensure integrated time-series, see chapter 6, and ε_{it} are independent across i units. The *OLS* in this context is inconsistent because the error term

$(\alpha_i + \varepsilon_{it})$ will be correlated with y_{it-1} since

$$y_{it-1} = \gamma y_{it-2} + x_{it-1}\beta + [\alpha_i + \varepsilon_{it-1}] \qquad (5.1.2)$$

and, therefore, correlated with α_i. To see this, difference (5.1.1) from (5.1.2)

$$y_{it} - y_{it-1} = (\gamma y_{it-1} - \gamma y_{it-2}) + (x_{it}\beta - x_{it-1}\beta) + (\alpha_i - \alpha_i) + (\varepsilon_{it} - \varepsilon_{it-1}) \text{ or}$$

$$\Delta y_{it} = \gamma \Delta y_{it-1} + \Delta x_{it}\beta + \Delta\varepsilon_{it} \qquad (5.1.3)$$

Then y_{it-1} in Δy_{it-1} is a function of ε_{it-1}; the latter becomes a component $\Delta\varepsilon_{it}$, resulting in the correlation of Δy_{it-1} and $\Delta\varepsilon_{it.}$ This inconsistent estimation problem is not confined to the least squares estimator. The demeaned fixed Effects estimator regresses $(y_{it} - \overline{y_\iota})$ on $(y_{it-1} - \overline{y_\iota})$ and has an error term $(\varepsilon_{it} - \overline{\varepsilon_\iota})$; y_{it} is correlated with ε_{it}; so y_{it-1} is correlated with ε_{it-1}, and therefore both correlated with, say $\widetilde{\varepsilon_t}$ (the average over *it* and *it*-1 periods), the component of the compound error term because of the common component $\overline{\varepsilon_\iota}$. Then if the regressor (y_{it-1} - $\overline{y_\iota}$) is correlated with (ε_{it} - $\overline{\varepsilon_\iota}$), the Fix Effects Estimator will also lead to inconsistent parameter estimates. Moreover, this is so even if α_i is modeled as a random effect, γy_{it-1} will be correlated with α_i , and, therefore, with the compound error (α_i+ ε_{it}).

5.2 *GMM generated instruments*

A solution is possible if there is at least a 3rd panel available; then although the first differences estimator is inconsistent, IV versions of this estimator result in consistent estimates because the additional panel provides a valid instrument to the endogenous lagged dependent variable. With T=3, we have

$$y_{it-3} - y_{it-2}=(\gamma y_{it-2} - \gamma y_{it-1})+(x_{it-3}\beta - x_{it-2}\beta)+(\varepsilon_{it-3} - \varepsilon_{it-2}) \qquad (5.2.1)$$

and then we can use y_{it-1} as an instrument for (γy_{it-2} - γy_{it-1})=Δ y_{it-2}.in the equation by the application of 2SLS since there is not component (5.2.1) error term from (t -1) period. In this case, there is one instrument for one endogenous variable and the IV leads to **just-identified** specification. However, this solution is inefficient because as T increases more instruments become available that are left unused by the 2*SlS* approach. For example, move (5.2.1) one period forward

$$y_{it-4} - y_{it-3}=(\gamma y_{it-3} - \gamma y_{it-2})+(x_{it-4}\beta - x_{it-3}\beta)+(\varepsilon_{it-4} - \varepsilon_{it-3}) \qquad (5.2.2)$$

Now y_{it-2} becomes an additional instrument, and, therefore, we have two instruments, y_{it-1} and y_{it-2}. However, the IV application using both instruments lead to **overidentified** specification because there is more than one instrument correlated with a single endogenous variable; in this case efficiency requires using all instruments for the differenced equation (5.2.1). Note both single and multiple instrument approaches *employ instruments in levels for differenced endogenous variables*. The result is that when the instruments are weak, we can obtain further instruments to improve consistency and efficiency by using *differenced instruments applied to endogenous variables in levels*.

However, identification is clear in a 2SLS with one instrument for each regressor but what if we have more instruments than the number of endogenous regressors? Then in a system of equations with a moment condition for each regressor we have more equations than the number of unknown parameters and the system cannot be perfectly solved because the specification is then overidentified. Such a system of equations can be modelled with a vector of regressors $\boldsymbol{X}$=(y_{it}, x_{it})

$$Y=X\beta + \varepsilon \qquad \& \qquad (\varepsilon \mid \boldsymbol{z})=0$$

where $\boldsymbol{z}$ is a column of j instruments, and $j \geq k$ regressors; the vector of empirical moment conditions $E_N(z\ \varepsilon) \neq 0$ when $j > k$. The problem is then to try to use all moment conditions at once so as to minimize the vector $E_N(z\ \varepsilon)$ as far as possible. A general solution is to de-emphasize the large variances by weighing the moments in *inverse proportions to their variances and covariances,* so instruments with high variances receive a lower weight and those with low variances a higher weight. We examined this approach in detail in chapter 2 as the **Generalized Method of Moments, or *GMM*** for short.

The choice of different weight schemes to lower high variance instruments and raise low variance instruments in order to minimize the variance -covariance matrix Ω of the error term leads to different *GMM* solutions to the identification problem in the presence of multiple instruments. If we assume homoscedastic errors, then $\Omega = \sigma^2 I$ where I is the identity matrix; the scheme in this case is the inverse of σ^2, called *one-step GMM,* that leads to consistent estimation. With heteroskedasticity, the GMM requires the application of robust sandwich or cluster-correction methods. This *two-step GMM* can also provide consistency. In this case, given consistent $\hat{E}$ estimates, we either have

$$\hat{\Omega} = \hat{E} = \begin{bmatrix} \widehat{e_1^2} & 0 & 0 & 0 \\ 0 & \widehat{e_2^2} & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \widehat{e_N^2} \end{bmatrix}$$

or, in the context of a wide panel (large N), with “clustered” individual patterns of covariance, we have

$$\widehat{\Omega_\imath} = \hat{E}_\imath \hat{E}_\imath{}' = \begin{bmatrix} \widehat{e_{\imath 1}^2} & \widehat{e_{\imath 1}^2}\widehat{e_{\imath 2}^2} & \cdots & \widehat{e_{\imath 1}^2}\widehat{e_{\imath T}^2} \\ \widehat{e_{\imath 2}^2}\widehat{e_{\imath 1}^2} & \widehat{e_{\imath 2}^2} & \cdots & \widehat{e_{\imath 2}^2}\widehat{e_{\imath T}^2} \\ \vdots & \vdots & \ddots & \vdots \\ \widehat{e_{\imath T}^2}\widehat{e_{\imath 1}^2} & \cdots & \cdots & \widehat{e_{\imath T}^2} \end{bmatrix}$$

The *inverse* of $\hat{\Omega}$ and $\widehat{\Omega_\imath}$ are then employed for the minimization of the variance-covariance of the instrument-conditional moments of the error terms.

Below we examine the dynamic models based on *IV* and different *GMM* weights for *Ω*. These models are all dynamic with time periods of data *T* small relative to cross-section units *N* (*N*>*T*), and unavailability of good outside instruments. Therefore, the only available instruments in the dynamic models are *internal*, that is lags of the instrumented variables.

5.3 *Anderson and Hsiao 2SLS estimator*

$$y_{it}=\alpha_i+\gamma y_{it-1}+\mathbf{x}_{it}\beta+v_{it}\ ;\ v_{it}=(\alpha_i+\varepsilon_{it})$$

where the error components are assumed independently distributed of each other as α_i ~ *iid* (0, σ_α^2) and where ε_{it} ~ *iid* (0, σ_α^2); hence σ_α^2 remain the same in both terms, that is the model assumes homoscedasticity. Taking first-differences will eliminate the unit-specific α_i ,

$$\Delta y_{it}=\gamma\Delta y_{it-1}+\beta\Delta x_{it}+\Delta v_{it} \qquad (5.3.1)$$

However, note that $E(\Delta y_{it-1}\Delta v_{it})\neq 0$; as a result, applying *OLS* to this model leads to inconsistent estimates. Note that even if ε_{it} is serially uncorrelated, its first difference will be correlated over time. Anderson and Hsiao (1981) suggested an IV approach to this problem by noting that since $E(y_{it-2}\Delta v_{it})=0$, y_{it-2} would be a valid instrument for Δy_{it-1} ; it is not correlated with ε_{it} *as long* the latter is not serially correlated, and yet likely to be highly correlated with Δy_{it-1}. Instrumenting by Δy_{it-1} leads to consistent but not necessarily efficient estimates because as the number of *T* increases, so do the available number of instruments, and yet the IV approach does not make full use of all instruments to improve efficiency. Moreover, the method ignores the autocorrelation in the first differenced errors Δv_{it} that can lead to inconsistent *IV* estimates for a large *T*. The Anderson and Hsiao estimators based on *2SLS* generate instruments in levels and in differences; the instruments in levels are usually employed to reduce the loss of degree of freedom since with, for example, Δy_{it-2} is not available until *t*=4 whereas with y_{it-2} becomes available with *t*=3.

5.4 *Panel GMM Estimators*

Two common *GMM* transformations are employed to improve efficiency by making use of the all available instruments. One model, known as the differenced *GMM*, is based on removing the fixed effects by differencing and using lag instruments in levels for the endogenous regressors in differences. Another, called the system *GMM*, employs regressors in levels, therefore, retaining the fixed effects, but employs lagged instruments in differences. The difference between the differenced and system *GMM* in their approach to endogenous fixed effects is somewhat akin to that between the fixed and random effects models, given that the former removes all time-invariants while the latter retains the time-invariants; except that the dynamic panel models depend on internal lagged instruments in levels or in first-differences. The time periods of the panel have to have at least t=3 to generate instruments in levels and t=4 for differenced instrument.

5.5 *Arellano-Bond Two-step Estimator with Instruments in Levels*

This model employs the orthogonality conditions between Δy_{it-1} and ε_{itt} to obtain additional instruments and employs the *GMM* approach using **all** moment conditions. Let us lag (5.3.1) as

$$(y_{i3} - y_{i2}) = \gamma\ (y_{i2} - y_{i1}) + \beta^{'}\Delta\ x_{i3} + \Delta v_{i3}$$
$$(y_{i4} - y_{i3}) = \gamma\ (y_{i3} - y_{i2}) + \beta^{'}\Delta\ x_{i4} + \Delta v_{i4}$$
$$\ldots. \qquad (5.5.1)$$
$$(y_{iT} - y_{iT\text{-}1}) = \gamma\ (y_{iT\text{-}1} - y_{iT\text{-}2}) + \beta^{'}\Delta x_{iT} + \Delta v_{iT}$$

In the first equation, the valid instrument for $(y_{i2} - y_{i1})$ is y_{i1}, since this instrument is correlated with $(y_{i2} - y_{i1})$ but uncorrelated with Δv_{i3}. In the second equation, the valid instruments for $(y_{i3} - y_{i2})$ are y_{i1}, and y_{i2}, and in the *Tth* equation, the valid instruments are y_{i1}, and $y_{i2}, \ldots, y_{iT\text{-}2}$. An additional instrument becomes available with each additional time period. Since the Δx_{ii} are assumed exogenous, they act as their own instruments. Thus, the number of available moment conditions is [*T.(T-1)*]/2 given by

$$E[y_{is}\ (\Delta y_{it} - \gamma\Delta y_{it\text{-}1} - \beta^{'}\Delta\ x_{it})]=0,\ \ s=0,\ 1,2,\ldots,\ t\text{-}2;\ t=2,\ 3,\ \ldots,\ T.$$

This method is known as the *two-step GMM*. The remaining problem is serial correlation in the transformed error terms $\Delta v_{it.}$ To deal with serial correlation, this approach applies *GMM* to observations stacked for *N* different groups. However, Blundell and Bond (1998) show that the *IV* and *GMM* estimators deteriorate as the variance of α_i increases relative to the variance of v_{it}, or when $\gamma \approx 1$; it can be shown that instruments in levels y_{it} are then weakly related to those in differences of Δy_{it} . Furthermore, when T is not small, as $T\rightarrow\infty$, the number of *GMM* orthogonality conditions *r*=[*T.(T-1)*]/2 tends to infinity. Finally, the consistency of this approach depends on serially uncorrelated errors ($\Delta v_{it}\ \Delta v_{it\text{-}2}$). With serially correlated errors, the *GMM* approach loses its consistency. For this reason, it is suggested that the application of this approach be accompanied by testing for higher second and higher-order autocovariances; note that the first-order negative serial correlation in (5.5.1) is expected and uninformative.

One weakness of the differenced transformation is that it magnifies data gaps in *unbalanced panels*. For example, if y_{it} is missing, then both Δy_{it} and Δy_{it+1} are also missing. However, there may be missing data for the entire time period for all variables when the panel is *irregularly spaced*, for example with a two-period interval, then a five-period interval, etc[9]. A second common transformation recommended by Arelleno and Bover (1995) and known as *forward orthogonal deviations*, is to subtract *all* available future observations of a variable from its contemporaneous one. In order to overcome the problem with unbalanced panels, this method minimizes loss of data by transforming a variable *w as*

$$w_{it+l} = c_{it}\left(w_{it} - \Sigma_{l>t}\, w_{it}\right)$$

where w_{it} stands for the unbalanced-panel-corrected y_{it}, and c_{it} is the scale factor equal to $\sqrt{T_{it}/T_{it}+1}$. That is, the method subtracts the average of all post observations of a variable beyond the missing period, no matter how many gaps in the data, thereby minimizing data loss.

5.6 *Blunder-Bond System GMM estimator with differenced instruments*.

We note that possible correlation in y_{it} is not only with y_{it-1} but also α_i; each gives a different interpretation of *correlation over time*. Assume for simplicity *β,* a pure *AR*(1) model

$$y_{it} = \alpha_i + \gamma y_{it-1} + \varepsilon_{it},$$

then

$$E(y_{it} \mid \alpha_i\, ,\, \gamma y_{it-1}) = \alpha_i + \gamma y_{it-1} \text{ and } \mathrm{Cor}(y_{it}\, ,\, \gamma y_{it-1} \mid \alpha_i) = \gamma\ ,$$

that is conditional on α_i. This is a standard *AR*(1) model of y_{it} solely determined by y_{it-1}. However, because α_i is not observed, we only observe $E(y_{it} \mid y_{it-1}) = \gamma\, y_{it-1} + E(\alpha_i \mid y_{it-1})$; therefore $\mathrm{Cor}(y_{it}\, ,\, \gamma y_{it-1}) \neq \gamma$ *,* and two possible reasons emerge for the correlation between y_{it} & y_{it-1}. When the causal relationship is from y_{it} to y_{it-1} is large, the *individual* effect $\alpha_i \approx 0$ and $\mathrm{Cor}(y_{it}\, ,\, \gamma y_{it-1}) = \gamma$ is the *true state dependence* outcome when σ^2_α is small relative to σ^2_ε. However, due to *unobserved heterogeneity,* there will be correlation $\mathrm{Cor}(y_{it},\, y_{it-1}) = \sigma^2_\alpha/(\,\sigma^2_\alpha + \sigma^2_\varepsilon\,) \neq 0$ even if $\gamma=0$. The first

[9] A recent study by Millimet & McDonough, (*J. A. Econometrics* 2017) examines correction for this type of missing panel data.

suggests a panel variable is driven by its own past values, while the second suggest important variables excluded from the model. Such circumstances lead to weak instruments. Blundell and Bond (1998) proposed an alternative model based on imposing restrictions on the distribution of the initial values y_{i0} that allow lagged *differences* of y_{it} as instruments in the levels equations. The restriction turns out to be important when instruments perform poorly, that is when $\gamma \approx 1$, or when $\sigma^2_\alpha/\sigma^2_\varepsilon$ is large since in such cases, lagged levels are weak instruments in the differenced equations. Consider the general $\gamma y_{i0} = \alpha_i /(1+\gamma)+ v_{it}$, $i=1, 2, \ldots, N$; based on the assumption that $E(\Delta y_{it}\, \alpha_i)=0$. The condition states that the deviations of the initial (independent of t) values from $\alpha_i /(1+\gamma)$ are uncorrelated with the level of $\alpha_i /(1+\gamma)$ itself. To secure this outcome, we further assume that

$$E\{[y_{i0} - \alpha_i /(1 +\gamma)]\, \alpha_i\}=0 \qquad (5.6.1)$$

Provided that this last condition holds, the following T-1 *additional* moment conditions also become available.

$$E[(y_{it} -\gamma\, y_{it-1})\, \Delta y_{it-1}]=0, \text{ for } t=2, 3, \ldots, T. \qquad (5.6.2)$$

The employment of this estimator, known as *System GMM*, when either γ is close to 1 or when $\sigma^2_\alpha/\sigma^2_\varepsilon$ is large, leads to substantial gains compared to the Two-Step *GMM*, especially when the instruments are weak.

Arellano and Bover (1995) suggested an alternative to the Arellano-Bond differenced instruments approach to dynamic panel bias; further developed by Blundell and Bond (1998). Instead of transforming the equation by differencing to remove the α_i fixed effects, this approach relies on instruments defined by transformed lag differences that are exogenous to the fixed effects. Using again w_{it} for the unbalanced panel correction y_{it} , if we can assume that $E(\Delta w_{it}\, \mu_{\mathrm{i}})=0$ for all i and t (if $E(w_{it}\, \mu_{\mathrm{i}})$ does not change over time), then Δw_{it-1} is a valid instrument for the variables in levels, that is, given the compound error $v_{\mathrm{it}}=(u_i\, \varepsilon_{it})$,

$$E(\Delta w_{it-1}\, \varepsilon_{\mathrm{it}})= E(\Delta w_{it-1}\, \mu_{\mathrm{i}}) + E(w_{it-1}\, v_{\mathrm{it}}) - E(w_{it-2}\, v_{\mathrm{it}})=0 + 0 - 0$$

Therefore, while the Arellano-Bond level instruments are orthogonal to the differenced variables, the Blundell-Bond differenced instruments are orthogonal to the level variables. More generally, if w have predetermined (exogenous and lagged endogenous) values, then Δw_{it} is also a valid instrument since $E(w_{it}\, v_{\mathrm{it}})=0$. Note that in unlike the differenced *GMM* that removes all time-invariant variables from the model, this alternative approach includes the time-invariants since the model is in levels; however, asymptotically the time-invariant variables are not affected by their inclusion since all variables are assumed to be orthogonal to the fixed effects.

5.6 *Specification tests for panel Serial Correlation and Overidentification*

In both approaches, however, the validity of the instruments depends on the absence of serial correlations because $w_{it\text{-}1}$ and $w_{\mathrm{it}\text{-}2}$ with past and contemporary errors, hence may correlate with future errors. The compound error term, v_{it}, is certainly autocorrelated due to its time-invariant component; the estimators are designed to remove the fixed effects. However, the error autocorrelation may be due to $\varepsilon_{\mathrm{it}}$, the time-varying component remains a potential problem that makes v_{it} autocorrelated of order 1, that is the differenced error term $\Delta\varepsilon_{\mathrm{it}} = v_{\mathrm{it}} - v_{\mathrm{it}\text{-}1}$, is serially correlated because $v_{\mathrm{it}\text{-}1}$ is a component of both Δv_{it} and $\Delta v_{it\text{-}1}$, etc. If this is the case, then the instrument set must be confined to lags 3 or more of y, assuming there is no 2nd order serial correlation. The fixed effect autocorrelation aside, we can test for serial correlation in error differences. We can ignore the order-1 negative serial correlation between Δv_{it} and $\Delta v_{it\text{-}1}$ because both share $v_{it\text{-}1}$ as uninformative. To check for serial correlation in levels, Arellano and Bond suggested testing for order-2 correlation in differences, that is, to test for serial correlation in the $v_{it\text{-}1}$ in the $\Delta v_{it\text{-}1}$ with the $v_{it\text{-}2}$ in the $\Delta v_{it\text{-}2}$; generally test for serial correlation of order k in levels by checking for correlation of order k+1 in differences. This solution for serial correlation would not work for the differenced errors since these are dependent on many forward lags. However, as long as none of the regressors depend on future disturbances, the test remains valid for *OLS*, *2SLS*, and any *GMM* panel regression, ruling out error correlation across individuals.

The other remaining issue is the test for valid instruments with the over-identification specification. When the number of moment conditions exceeds the number of parameters, $r>q$, the model is over-identified, since more orthogonality moment conditions are employed than required. Under the null hypothesis of the joint validity of moment conditions, the vector of empirical moments for instrument orthogonality is randomly distributed around 0. A Wald test can check the hypothesis. Sargan (1958) and Hansen (1982) suggested a frequently employed Wald overidentification restriction test, known as the *J-statistic* that is distributed as $\chi^2_{(r\text{-}q)}$ for the number of overidentified restrictions (r - q), for models estimated by the *GMM*. *J*-statistic larger than the corresponding critical values means rejection of the hypothesis that the moments are supported by the data; suggesting that at least some moment conditions are unsupported (invalid). The test can also be applied to investigate an additional vector of moments have 0 means, and therefore, can be included in the moment conditions to improve inference. It should be noted that when *T* is large, using too many moment conditions results in the Sargan-Hansen overidentification results in a test with a very low power.

5.7 *SURE panel Estimator: Long & Narrow*

Introduction

These are the panels with limited cross-sectional units on which there are long time-series sets of data are available. With Long and narrow panels with a narrow *G* cross-sectional units g=1, 2, …, G and time-series over *t=1, 2, …T* periods, we have a system of *g*-equations over *t* periods, usually with the same vector of explanatory variables, as

$$Y_{gt}= \alpha_g X_{\mathrm{gt}} + \mu_g$$

If there is contemporaneous correlation between μ_g s, then we must estimate this g-equation system of equations together to exploit the correlation to obtain improved estimation accuracy.

Suppose we have to estimate two different investment function for two different private banks over a 30-year period; there are several ways to estimate such a function as a system of two investment equations in order to obtain improved estimation efficiency:

1- If we assume the parameter coefficients are the same for all two equations, we could just pool the data and apply OLS to a single equation with the same coefficients for all three banks.

2- It might be more plausible to assume the coefficients to be different for each bank. Then we add two dummies and their interactives with the explanatory variables, and once again estimate the function with an application of the *OLS* to a single equation.

3- Further plausibility might come from assuming different slopes *and* different variances, estimates of the error terms for each of the two equations. Then we could apply *GLS* to a single equation for both banks that takes care of heteroeskadasticity of the variances, assuming no contemporaneous correlation between the two error terms.

4- Moreover, we could allow the two error terms to be autocorrelated (serial correlation) within each equation, but uncorrelated across equations, and apply *FGLS* or Newey estimators to a single equation.

5- Finally, we could relax the assumption further by allowing *contemporaneous correlation* among the two error terms, that is, allow the error term for the first bank equation to be correlated with that of the second bank ***in the same time period*.** In our example, for instance, both banks are similarly influenced by the change in the economic outlook of the country in a given year. Usually, the correlation between different time periods are assumed zero, namely, no serial correlation. It would then be necessary to estimate a system of two separate equations together by GLS, using an estimation of the covariance between the two variances: Cov(e_1 e_2)= $\sigma \neq 0$, i.e. no contemporaneous correlation. The estimator employed in this case is known as the ***SURE*** (seemingly unrelated estimation), the main topic of this section.

6- In each 1-5 cases, we must test the relevant assumption to justify the chosen estimation model.

i. Panel Exogeneity Assumption

What kind of exogeneity assumption we use is critical for the estimator choice with panel data sets. The simplest is to assume ***contemporaneous exogeneity of X_t***; that is a weak restriction that X_t and μ_t are uncorrelated in the *same time period t*:

$$E(\mu_t|X_t)=0, \qquad t=1, 2, \ldots T. \qquad (5.7.1)$$

A stronger assumption, called ***sequential exogeneity,*** is that *all* current and past explanatory variables are uncorrelated with μ_t:

$$E(\mu_t|X_{t,}, X_{t-1, \ldots} X_{t,})=0,\ t=1,\ 2,\ \ldots T. \qquad (5.7.2)$$

This implies that $E(y_i|X_{t,}, X_{t-1, \ldots} X_{t,})= E(y_i|X_t)= X_t\beta$, that is no lags of X_t are required to obtain the expected value of Y_t.

A much stronger exogeneity assumption is that μ_i has zero expectation conditional of all variables in *all* time periods, including future periods.

$$E(\mu_t|X_{1,},X_{2,}\ldots X_{t,})=0,\quad t=1,\ 2,\ \ldots T. \qquad (5.7.3)$$

Called **strict exogeneity**, this assumption excludes any correlation between the error term and all explanatory variables in all time periods, including future time periods. Strict exogeneity is false if the equation contains lagged dependent variables, therefore, assumption (5.7.3) fails. Such an equation is a common feature of *dynamic* models of panel data with lagged dependent variables as regressors. For example, suppose the vector of explanatory variables X_t=(1, y_{t-1}), so that

$E(\mu_t|X_{1,},X_{2,}\ldots X_{t,})= E(\mu_t|Y_{0,},Y_{1,}\ldots Y_{t-1,})= \mu_t \neq 0$ for $t=1,\ 2,\ \ldots T\text{-}1$, because $\mu_t=(Y_t-\beta_0 - \beta_1 Y_{t-1})$

Even without lagged dependent explanatory variables, strict exogeneity fails. For example, consider a finite distributed lag model of poverty (P_t) as a linear function of current and lagged welfare expenditure (W_{t-1}).

$$P_t=\theta_t + \delta_0 w_t+\delta_1 w_{t-1}+u_t$$

where θ_t represents a time effect. Then if welfare changes with P_{t-1} as

$$W_t=\eta_t + \rho_1 P_{t-1}+e_t \qquad (5.7.4)$$

(5.7.4) would violate strict exogeneity if $\rho_1\neq 0$ because W_{t+1} would then be affected by u_t. Assuming that X_t is fixed in repeated sampling with panel data is the same as the classical model assumption of strict exogeneity. However, panel data analysis by *SURE* does not contain lagged dependent explanatory variables, see chapter 14 for that kind of model.

The *SURE* Estimator employs different variances for each equation in order to estimate the error covariance in each period; then adds them up and corrects the sum for loss of degrees of freedom (*T* – the number of explanatory variables). Using the extra information provided by the estimated covariance, the SURE estimates a system of equations, each with different slopes and different correlated error variances for covariance, with lower standard errors to improve efficiency.

Following steps are involved in the *SURE* procedure:

1-Each equation is separately estimated by *OLS*

2-Estimated variances are employed to estimate the covariance. For a two-equation example above, that is

$$\sigma_{12} = \frac{1}{\sqrt{T-K_1}\sqrt{T-K_2}}\sum_{t=1}^{T} \widehat{e_1}\,\widehat{e_2}$$

where K_1 and K_2 are the number of parameters (excluding the intercept) and $\widehat{e_1}$ & $\widehat{e_2}$ ate the estimated standard errors.

3-*SURE* uses estimates from step 2 to estimate a system of equations jointly by the FGLS method. Prior to the employment of *SURE* estimator, it is necessary to carry out a test of *Ho:* $\sigma_i^2 = \sigma_j^2$ with $i \neq j$ for a significant differences between the variances. This can be decided by a two-tail Goldfeld-Quandt. The test assumes group-wise heteroscedasticity such that $\sigma_i^2 = \sigma^2 x_i^2$, e.g. for income as x_i, rank the observations by x, and then split them into low and high x_i and compute the F test statistic as

$$F_{(n1\text{-}k,\ n2\text{-}k)} = \frac{\widehat{e_1} e_1/n_1 - k}{\widehat{e_2} e_2/n_2 - k}.$$

If we cannot identify x_i, then an alternative test is the LM Breusch-Pagan tests based on

$$\sigma_i^2 = \sigma^2(a_0 + a'\boldsymbol{z})$$

with a vector $\boldsymbol{z}$ of independent variables; with this test statistic, the model is homoscedastic if a=0, Greene (2000, pp. 509-510).

ii. When the SURE has no efficiency advantage over the OLS

There are two situations under which the efficiency gain from the SURE is no more than that obtained from the least squares:

1-if there were no contemporaneous error correlation, then there would be no links between the separate equations, and hence no gain in efficiency by estimating a system of equations together by the *SURE*; the results would be identical to those obtained by the *OLS* or *FGLS* equation-by-equation estimation.

2-Somewhat less obvious, if the same explanatory variables with the same observations appeared in all separate equations, then the *OLS* and *SURE* would produce *identical* estimates even if each equation has a different error variance; again, no gain in efficiency by the *SURE* compared to the *OLS*., Greene (2000, pp. 616-17). If the explanatory in each equation were *different*, then a test would be required to see if the correlation between errors significantly differ from zero when estimated *jointly* by *SURE* compared to separately by *OLS*; if so, then *SURE* simultaneous system of equation estimation would improve efficiency over the *OLS* separate estimation. A variance-covariance ratio test statistic ratio must be computed to decide the outcome. For a simple two-equation system, for example, the test computes $r^2_{12}= \sigma^2_{12} / \sigma^2_1 * \sigma^2_2$ to test Ho: $\sigma_{12} =0$; this test has a $LM=(T^* r^2_{12})$ test statistic distributed as $\chi^2_{(1)}$; more generally with a *m*-equation system:

Ho: $\sigma_{12} =\sigma_{13} = \sigma_{23} =\ldots\sigma_{lm} =0$ & $\chi^2_{(M)}$ has test statistic

$$LM=T(r^2_{12} + r^2_{13} + r^2_{23} \ldots + r^2_{lm})=T\sum_{i=2}^{m}\sum_{j=1}^{i-1} r_{ij}^2.$$

3-One advantage of the *SURE* estimator is that it allows testing for *cross-equation coefficient restrictions in different equations.* in this case *SURE* is still useful even if the explanatory variables were the same because the standard *F*-test can only test for coefficient restrictions within the same equation, not across different equation. To test a joint hypothesis across different equations requires estimating a variance matrix of coefficients to obtain covariance estimates from different equations; the *SURE* preforms this task automatically, computing *F* or *Wald* statistics.

As an example of using the SURE to test for cross equation restriction, consider a two-equation model

4- $y_1=\gamma_{10}+\gamma_{11}x_{11}+ \gamma_{12}x_{12}+\alpha_1 x_{13}+\alpha_2 x_{14}+u_1$ (5.7.5)

5- $y_2=\gamma_{20}+\gamma_{21}x_{21}+\gamma_{22}x_{22}+\alpha_1 x_{23}+\alpha_2 x_{24}+u_2$ (5.7.6)

Therefore, α_1 & α_2 are restricted to be equal in both equations. We can redefine the vector of parameters β as $\boldsymbol{\beta}=(\gamma_{10}, \gamma_{11}, \gamma_{12}, \alpha_1, \alpha_2, \gamma_{20}, \gamma_{21}, \gamma_{22})'$; then we choose $\boldsymbol{X_i}$ as the (2 x 8) matrix corresponding to

$$\boldsymbol{X_i} = \begin{pmatrix} 1 & x_{i11} & x_{i12} & x_{i13} & x_{i14} & 0 & 0 & 0 \\ 0 & 0 & 0 & x_{i22} & x_{i23} & 1 & x_{i21} & x_{i24} \end{pmatrix},$$

wherex_{i11} and x_{i12} are set equal to zero in the second equation because they are specific to the first equation and each equation intercept is set equal to 1, etc. $\boldsymbol{X_i\beta}$ then leads to (5.7.5) and (5.7.6). In this case, the unrestricted model is one with its own parameters, and its estimated variance-covariance matrix employed to obtain the restricted estimates.

Readings

For textbook discussion, see Pesaran (2015, chapter 27), Cameron and Trivedi (2005, chapter 22). Arellano and Bond (1991), Arellano and Bover (1995), and Blundell and Bond (1998) are papers on the short-span panel data models; Zellner (1962) proposed the SURE model.

Chapter 5 Dynamic Panel Data Models Exercises

Q 5.1 Consider a dynamic panel set up in which lagged values of the dependent variable are included as covariates.

a. Explain why the OLS, within fixed effects and first-differenced estimations are all inconsistent.
b. Explain and write down in mathematical terms the solutions of Anderson-Hsiao and Arellano-Bond to this problem.
c. Explain the differences between these two methods in estimation results.

Q 5.2 Download *mus08psidextract.dat*.

a. Apply the Arellano-Bond *gmm* estimator to a pure time-series AR(2)of pure time-series model of *lwage*, employing robust standard errors; test the results for 2nd & 3rd order serial correlation and account for the number of instruments.
b. Apply the Arellano-Bond estimator to a AR(2) model of *lwage* as a function of pre-determined lagged *wks*, endogenous *ms & union*, and exogenous *occ south smsa & ind*. Keep the number of instruments for lagged dependent limited to 3, employ robust standard errors, account for the n umber of instruments used, and test for 2nd & 3rd order serial correlation and for overidentification.

Q5.3 Apply the Blundell-Bond/Arellano-Bover model to the same model as Q5.2, but this time with differenced instruments. Comment on the outcome

Q 5.4 *Download grunfeld2.dta.* This data set contains two firms, GE & WE, each with 20 time-series observations on the values of each firm investment, inv, stock market values, *v*, and capital stocks, *k*, of each firm.

a. Estimate a pooled regression equation for *inv* as a linear function of *v & k,* and test for coefficient equality using a dummy version of the equation.
b. Test the variances of each firm's equation for equality, then reshape the data set as a long panel data and estimate the model by *SURE*.
c. Obtain the estimated variance/covariance matrix of the residuals used by *SURE.*

Chapter 6 Forecasting with AR, MA Processes

Introduction

A substantial part of time-series analysis deals directly or indirectly with forecasting. Forecasting of a time variable relies on two models of forecasting: either we explain change in the variable in terms of its own lagged values (autoregression), or in terms of the lagged values of the error terms of an autoregressive equation (moving average), or a combination of both. Forecasting by these processes must meet the conditions for stability and select the optimal number of lags to be effective. In this chapter we examine such conditions, and define the moments of these two key processes, and compare their forecasting performances.

Basics of Forecasting

i. Loss Function

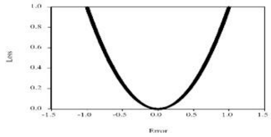
A loss function quantifies what is a “good” forecast. Example: given four demand outcomes, in two we make the correct inventory decision, and in the other two incorrect inventory decision. There are losses involved in such decision, for example if you decide to keep a low inventor but demand turns out to be high. Note that the loss can be symmetric as here, or asymmetric, if the cost of lost sales is bigger than the cost of unneeded inventory. The losses involved lead to similar losses in the forecasts on which inventory decisions are made.

Define loss as $e = y - y_{\_hat}$. Then the **quadratic loss** function

$$L(e) = e^2$$

This function is symmetric around the origin; increases at an *increasing* rate on either side of the origin. Thus, larger errors are penalized much more heavily than small errors.

Quadratic Loss

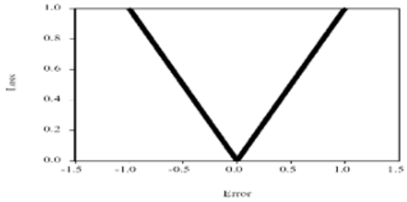


Another symmetric function is the **absolute error loss** is:

$$L(e) = |e|$$

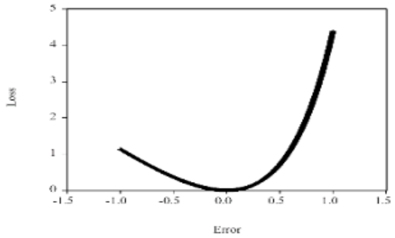
This is also a symmetric function with increasing absolute loss on either side of zero but the loss increases at a *constant* (linear) rate with the size of the error.

Absolute Loss



An asymmetric loss function may be more relevant to decisions. Here, negative forecast errors are less costly than positive errors:

Asymmetric Loss



In finance we may be interested in direction of change in forecast rather than its size, that is the forecast variable is discrete rather continuous (as defined by $y - y_{\_hat}$).

$$L(y,\hat{y}) = \begin{cases} 0, \text{ if sign } (\Delta y) = \text{sign}(\Delta \hat{y}) \\ 1, \text{ if sign } (\Delta y) \neq \text{sign}(\Delta \hat{y}) \end{cases}$$

We obtain optimal forecasts by minimization of expected loss of a loss function.

ii. The Forecast Statement

We most frequently work with point estimates because density forecasting involves the possibility of incorrect distributional assumption, and point forecasts are easy to understand and may be used to guide action. *Point forecast*s are those forecasting next year's GDP; *interval forecasts* are when we forecast the range of values with a certain probability. The *forecast horizon* is the number of periods between now and the forecast date and can change depending on the frequency of observations. A one-step forecast with monthly data is one month ahead; with quarterly data, one quarter ahead, and in general, a *h-step ahead forecast,* is a h-step ahead forecast at time *T* is a *single T+h* value, and h-step ahead **extrapolation** forecast at time *T* is a *set* of *T+h* values as shown here for *T*+4. In forecasting we can use univariate *information set*, that is the historical values of a series *y* up to and including the present; or, use a multivariate information set including an additional a set for *x* variables potentially related to *y*.

FIGURE 2.6 4-Step-Ahead Point Forecast

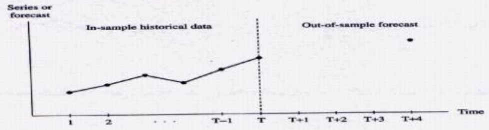
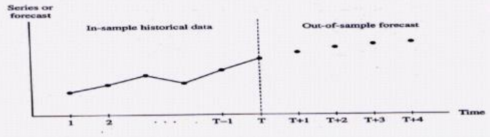


FIGURE 2.7 4-Step-Ahead Extrapolation Point Forecast



$$\Omega_T^{\text{uni var iate}} = \{y_T,\ y_{T-1},\ \ldots,\ y_1\}$$

$$\Omega_T^{\text{multi var iate}} = \{y_T,\ x_T,\ y_{T-1},\ x_{T-1},\ \ldots,\ y_1,\ x_1\}$$

The simplest, *parsimonious* models tend to preform best in out-of-sample forecasts in finance because: simple models have more precise parameter estimates, are easier to communicate, and reduce the scope for *data mining*, obtaining a model that fits the historical data very well, but preform very poorly in out-of-sample forecast because it is estimated with unusual features of past data with no relationship to the future forecast. The *Parsimony Principle* maintains that smaller is often better.

iii. Deterministic Trend

A *Trend* is a slow, long-run evolution in the forecast variable. A *Deterministic Trend* changes in a perfectly predictable manner while a *Stochastic Trend* changes randomly over time, the former displays a long period of increase followed by a long period of decrease. An example is the US unemployment rate. Some series have an obvious upward or downward trend, so the path should include a "drift" term for an improved forecast (if <0 then downward tendency, if >0 then upward tendency, such as the log of US GDP). This may require fitting different time-trend to different data sets, for example for a linear trend for unemployment as a function of time

$$U_t=\beta_0+\ \beta_1 TIME_t.\rightarrow (\ \hat{\beta}_0\ ,\ \hat{\beta}_1\ )=min\sum_{t=1}^{T}\ (U_t-\beta_0-\ \beta_1 TIME_t)^2.$$

iv. Selection of Forecasting Model

We select a forecast model based on minimizing its forecast error. Let us exclude the last period *t* from the sample. The model selection can be based on the smallest out-of-sample 1-step-ahead (predicted) ***mean squared error (MSE):*** $MSE=\frac{\sum_{t=1}^{T}e_t^2}{T}$, where T is the sample size and

e_t= (y_t - $\hat{y}_t$), *and* $\hat{y}_t=\hat{\beta}_0+\hat{\beta}_1 TIME_t$ Since we apply the same sample size to all trend models (linear or quadratic) above, minimization on the basis of MSE is the same as minimizing the sum of squared residuals (numerator of *MSE*); or similarly, as maximizing R^2.

Should we select the forecast model on the lowest *MSE*, or highest R^2? Note *MSE cannot increase* as we add more trend terms to the model, just as R^2 cannot fall, yet these additional trend terms may be irrelevant, resulting in *over-parameterization* (*data-mining*). A high R^2 may result in too complex models with very good in-sample fit, but this practice often results in poor out-of-sample fit. The consequences are overt underestimation of in-sample *MSE* and therefore biased out-of-sample forecast estimates because *MSE* or R^2do not penalized for the inclusion of additional trend terms, namely, for the loss of degree of freedom. There is thus a trade-off between good fit and the number of variables included. Good criteria for model selection should be based on such an optimal trade-off.

We know that adjusted R^2 does correct for loss of *df*, and that suggests a penalizing factor for the in-sample *MSE* for loss of *df* and obtaining s^2*,* the *MSE* corrected for *df* as the square of the standard error of the regression: $s^2=\frac{\sum_{t=1}^{T}e_t^2}{T-k}=\{\frac{1}{1-(k/T)}\}\frac{\sum_{t=1}^{T}e_t^2}{T}$ where *k* is the number of parameters in the trend model, and rewritten with a separate the term inside the last (curly) to make explicit the penalizing factor employed. Minimizing s^2 is then equivalent to maximizing $\bar{R}^2$=1 –

s^2 / [$\sum_{t=1}^{T}(y-\bar{y})/(T-1)$] obtained from unadjusted R^2 by division of the nominator by (T-k) and the denominator by (T-1). Note since the denominator of $\bar{R}^2$ depends only on the data, minimization of s^2 amounts to the maximization of $\bar{R}^2$.

v. Akaike & Schwarz Information Criteria

The penalizing factor for s^2 is a function of k/T. Two other model selection criteria also defined as functions of k/T are:

$$AIC=e^{(2k/T)}\frac{\sum_{t=1}^{T}e_t^2}{T} \text{ or } ln(AIC)=ln\ (\frac{\sum_{t=1}^{T}e_t^2}{T})+2(k/T);\text{ and}$$

$$SIC=T^{(k/T)}\frac{\sum_{t=1}^{T}e_t^2}{T} \text{ or } ln(SIC)=ln(\frac{\sum_{t=1}^{T}e_t^2}{T})+(k/T)ln(T).$$

Note that for both *AIC* (*Akaike information criterion*), and *BIC* or *SIC* (*Schwarz information criterion*), the first term (*MSE*) becomes smaller as extra variables are added, but the second term becomes larger to penalize for extra variables. In fact, all three have the general form of “MSE scaled by a penalty factor” that is a function **(*k/T*)**. We can thus compare model selection in terms of penalty severity. The difference between ln(*AIC*) and ln(*SIC*) hinges on the difference between 2 and *ln(T)*, *ln(T)*> 2 for sample sizes *T*>8 since ln8=2.08 *v.* 2 for *AIC*. Therefore, even with moderate sample size of 10 or more, *SIC* penalizes additional variables more severely than *AIC*. In any case, the second term in *AIC* is not large enough to ensure the correct number of polynomial terms even in large samples, so AIC (and s^2) is not consistent. This can be seen in Figure 6.1 as all three are functions of *k/T*, so the penalty rises with increase in *k/T* (from 0 to 0.25 for a given sample size of 100), but more slowly with s^2, somewhat faster with *AIC* but very sharply

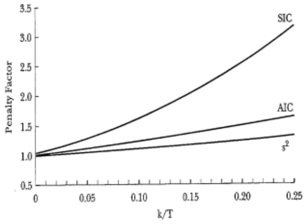
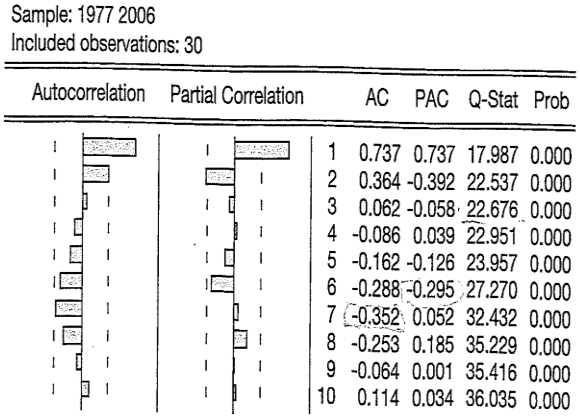


Figure 6.1 *Degrees-of-Freedom Penalties with Various Selection Criteria*

with *SIC as the sample size increases*; so s^2<*AIC*<*SIC*. Usually, *AIC* and *SIC* both select the same model, and if they lead to different models, the recommendation is to select the more parsimonious model based on the lowest *SIC*. However, note that such comparisons can only be made for models with the *same* dependent variable, e.g. either y or ln(y). Finally note that although we employ *AIC* and *SIC* here for deterministic models, these play a more important roles in stochastic forecasting models when we wish to decide the impact of lagged values of a variable on its current value. Including too few lags in the in-sample model means loss of valuable information, while including too many leads to complex models with poor out-of-sample performance. *AIC* and *SIC* are often employed to determine a balanced trade-off for the number of lags and select the optimal model.

Model selection can also be based on diagrammatic **correlogram** plots (Figure 6.2) showing the correlation between observations one or more periods apart indicating if they are significant at 5 % or 1%.; some econometrics texts do not bother with correlogram plots and rely almost entirely on *MSE* and its related criteria for model selection. An example is from US annual working hours per employee, showing plots of correlogram and *AIC* (here *AC*), and *SIC* (here *PAC*). The broken lines are the confidence intervals; values falling within the interval suggest the series is stable since it changes close to the mean. Note that there is a conflict here as *AC* picks the model with 7 lags but *PAC*, the one with two lags which is the preferred one due to its smaller number of parameters?



Sample: 1977 2006
Included observations: 30

Autocorrelation	Partial Correlation		AC	PAC	Q-Stat	Prob
		1	0.737	0.737	17.987	0.000
		2	0.364	-0.392	22.537	0.000
		3	0.062	-0.058	22.676	0.000
		4	-0.086	0.039	22.951	0.000
		5	-0.162	-0.126	23.957	0.000
		6	-0.288	-0.295	27.270	0.000
		7	-0.352	0.052	32.432	0.000
		8	-0.253	0.185	35.229	0.000
		9	-0.064	0.001	35.416	0.000
		10	0.114	0.034	36.035	0.000

Figure 6.2 *correlogram*

A cycle usually involves up and down movements. The distinction between cyclical movements that have a constant mean over time and those that do not is a central question in time series analysis. Let us see why. In cross sectional analysis we can estimate the population mean by the sample mean interpreted as the mean of the distribution of sample means obtained in *repeated sampling* from the same population. With time series data, we only have one observation per each time period and we cannot go back in time to the mean from repeated sampling. We can, however, interpret an observation for a particular time period as representing the mean of the distribution of all possible outcomes generated by a stochastic process. For example, we can think of a monthly DJ value per unit time as the mean of a stochastically generated distribution of all possible values at that time. But if the stochastic mean may be different for different time periods, which one of the population means as represented by sample values of the series is the sample mean of a time series over all time periods it represents? What is the population mean for the DJ index itself if it keeps changing over time; which particular population mean is the sample mean an estimator for? The answer is none if the population mean is not constant. That question would be *meaningless* if the population mean varies over time, as illustrated in Figure 6.3. We call such a time series **nonstationary**; what mean of the series in figure (a), the sample mean in figure (b) stands for? If, however, the population mean is constant over time, then we can employ the sample mean as an estimate of the stochastically generated population mean and test to find out if it is a good estimator. Such a time series we call **stationary**. In this case, we can say the sample mean in figure (d) represents an estimate of the (unique) stochastically generated population mean in figure (c). Note that a stationary state as shown here only requires the series to have constant mean, not necessarily that the mean is obtained from a normal distribution.

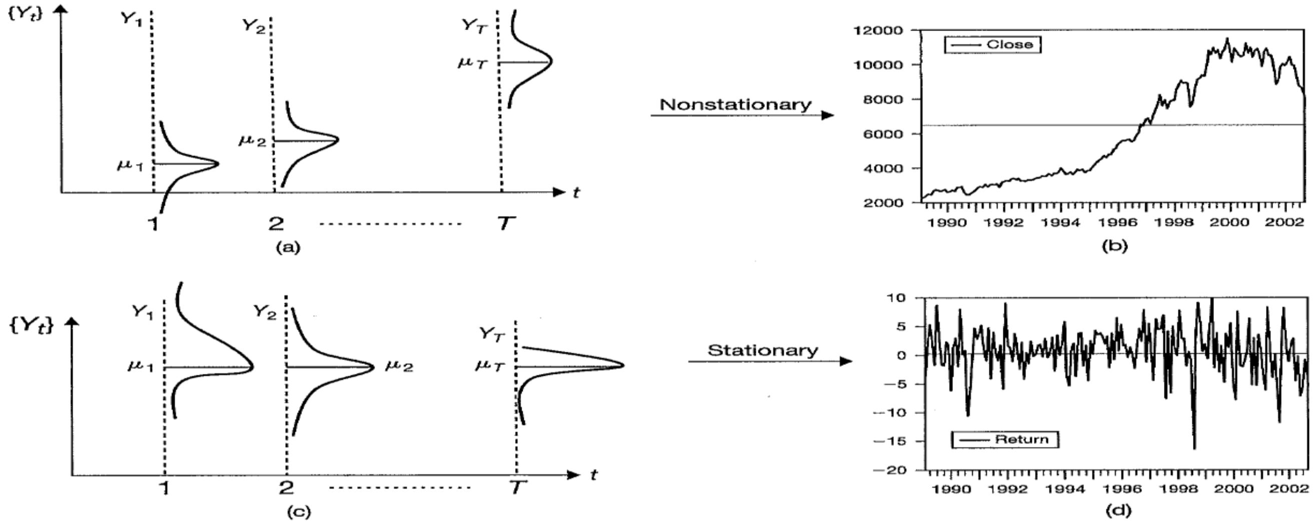


Figure 6.3 *Time-series: Non-stationary and Stationary*

The problem is the stochastic process that generated the series is unknown. We must make assumptions about the underlying stochastic process from which a time series sample is drawn in order to make inferences about the population using the sample moments. If the sample displays non-stationary properties, then we must first convert it into a stationary series before we can obtain sensible estimates and test results; and we shall discuss some methods to do so later.

6.2 *forecasting requirements*

The constancy of a series has two essential requirements: that the forecasting models of cycles of time series be *covariance stationary* and have a *white noise error* term.

i. Covariance Stationary

we define the **autocovariance function**γ based on the distance between y_t and y_{t-1} called the **displacement τ**. So,

$$\gamma(t,\tau)=\text{Con}(y_t, y_{t-\tau})=E(y_t - \mu).(y_{t-\tau} - \mu).$$

Covariance stationarity requires that $\gamma(.)$ be independent of time and depend only on displacement τ, that is $\gamma(t,\tau)=\gamma(\tau)$ for all t; that also implies the autocovariance function is symmetric: $\gamma(\tau)=\gamma(-\tau)$. Note that at $\tau=0$, $\gamma(0)=\text{Con}(y_t, y_t)=\text{Var}(y_t)$. Using $\gamma(0)$ as a bottom line, we finally require that no autocovariance be larger in absolute value than $\gamma(0)$; and this condition is met if $\gamma(0)$ is finite, that is if $\gamma(0)<\infty$ as a sort of upper bound for the autocovariance function.

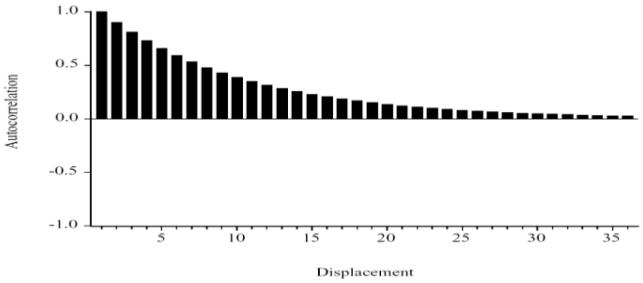
In practice, we often work with **autocorrelation function** instead of the autocovariance function; the autocorrelation function has a much easier meaning as $Corr(y_t, y_{t-\tau})\in(-1, 1)$ whereas $Cov(y_t, y_{t-\tau})$ can take any value and is not independent of the unit of measurement used.

$$\rho(\tau)=\frac{\text{cov}(y_t, y_{t-\tau})}{\sqrt{\text{var}(y_t)}.\sqrt{\text{var}(y_{t-\tau})}}=\frac{\gamma(\tau)}{\sqrt{\gamma(0)}.\sqrt{\gamma(0)}}=\frac{\gamma(\tau)}{\gamma(0)} \text{ for } \tau=1, 2, 3, \dots$$

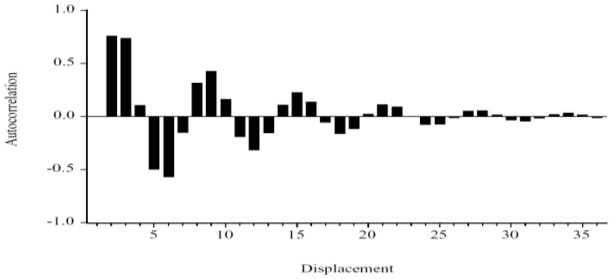
Note that since $\rho(0)=\gamma(\tau)/\gamma(0)=1$, a series' dynamic structure can only be examined by correlations beyond displacement at 0.

The autocorrelation function does not control for the influence of the periods *in-between* y_t & $y_{t-\tau}$ when $\tau>1$. A ***Partial*** autocorrelation function measures linear correlation after controlling for the effects of y_t & $y_{t+1-\tau}$. Both types of autocorrelations approach zero as τ becomes larger; the decline may be a gradual, one-sided decay, or a decay pattern that oscillates in sign.

Autocorrelation Function with one-sided Gradual Damping



Partial Autocorrelation Function with Gradual Damped Oscillation



ii. White Noise errors

Suppose $\boldsymbol{y_t=\varepsilon_t}$ where $\boldsymbol{\varepsilon_t}$, the **shocks**, are serially uncorrelated over time, and ε_t ~(0, σ^2). A process with zero-mean and constant variance and no serial correlation is a *zero-mean white noise* process and written as ε_t ~WN(0, σ^2). A (weakly) white noise ε_t is not necessarily normally distributed; if it is, then ε_t is a *normal* (also called *Gaussian* or *strong*) white noise written as ε_t ~ *iid*(0, σ^2). There are **no** patterns in a white noise process. The first two unconditional moments of a series y_t are E(y_t)=0 and Var(y_t)=σ^2. Because a white noise series is uncorrelated over time, all autocorrelation and partial autocorrelations beyond displacement, are zero. We usually contrast *conditional mean and variance* of a series with its unconditional mean and variance in order to discover the series dynamic patterns. The conditioning information set consists of either the past history of the series or the past history of the shocks to that series Ω_{t-1}={ y_t, y_{t-2}, ...} or {ε_t, ε_{t-2}, ...}. Unlike unconditional mean and variance, the conditional mean and variance are not necessarily constant. However, given an independent white noise process, the conditional moments are E(y_t |Ω_{t-1}|=0, and Var(y_t|Ω_{t-}

$_{1}$)=E[(y_t – E(y_t)|Ω_{t-1})]^2=σ^2. Therefore, the conditional and unconditional moments are *identical* for an independent white noise series.

iii. The Lag Operator

We use *lag operator* notation to express forecasting models with many lags and parameters in shorter, compact notations. The lag operator with one lag is written as L^1$_{\text{yt}}$= y_{t-1}; with two lags is then L^2$_{\text{yt}}$= L$_{(y t-1)}$= y_{t-2}. Other examples: first-differenced operator Δ is in fact a first-order lag operator since Δ$_{\text{yt}}$ =(y_t - y_{t-1})= (y_t - L$_{yt}$)=(1 – L)y_t; or a second-order lag operator, e.g. (y_t + 0.9y_{t-1}+ 0.6y_{t-2})=(1+0.9L+0.6L^2)y_t, is an example of a *distributed lag* model, a weighted sum of past and present lag values. More generally, the lag operator written with a *polynomial in the lag operator* of degree *m* is

$$B(L)=b_0+b_1L+b_2L^2+\ldots+b_mL^m=\sum_{i=0}^{m} b_i L^i \text{ for } i=0,\ 1,\ 2,\ \ldots,\ m.$$

6.3 *Wold Theorem and its approximation*

Many different dynamic structures are consistent with covariance stationary state, leaving open the question of model selection. The Wold theorem points to the appropriate model selection. The Wold Theorem claims that no matter how a process generated a time series, it can always be represented as a function that is **linear** in its unpredictable past $\varepsilon_{t.}$ as long as the series is covariance stationary and ε_t is a zero mean process. Such a unique linear presentation consists of an infinite polynomial of ε_t:

$$\text{B(L)}\varepsilon_t = b_0\varepsilon_t + b_1\varepsilon_{t-1} + b_2\varepsilon_{t-2} + \ldots = \textstyle\sum_0^\infty bi\ \varepsilon_{t-I}\ ;\ \sum b_i^2 < \infty\ ,\ b_0=1\ \&\ \varepsilon_t \sim WN(0,\ \sigma^2).$$

Such an infinite distributed lag of white noise shocks is called a **Wold representation**. ε_t is a sequence of one-step ahead random shocks called **innovations**. At time *t*, ε_t is a “surprise” to $y_{t.,}$ but all other shocks ε_{t-1}, ε_{t-2}, … are already known and processed in $y_{t.}$ Because there must be an *i* period after which the shock effects may become negligible, the theorem assumes$\sum b_i^2<\infty$. The Wold Theorem does not assume (Gaussian) normality of ε_t, but ε_t is a white noise process. The conditional mean and variance of Wold representation show the dynamics of series modeled in terms of its conditional first two moments. The conditional mean:

$$E(y_t\,|\Omega_{t-1})= E(\varepsilon_t\,|\Omega_{t-1})+b_1E(\varepsilon_{t-1}\,|\Omega_{t-1})+ b_2E(\varepsilon_{t-2}\,|\Omega_{t-1})+\ldots=0+b_1\varepsilon_{t-1}+b_2\varepsilon_{t-2}+\ldots = \textstyle\sum_0^\infty bi\ \varepsilon_{t-i}$$

i.e. mean changes over time *t* with availability of information. Conditional variance:

$$Var(y_t\,|\Omega_{t-1})= E(y_t - E(y_t\,|\Omega_{t-1}))^2= E(\varepsilon^2{}_t\,|\Omega_{t-1})= E(\varepsilon^2{}_t)=\sigma^2$$

The Wold theorem is a major step forward in providing the basis for forecasting and clarifying the conditions required to do so. However, it also has a major practical problem, namely it is based on an infinite series, and involves estimation of an infinite number of parameters! Therefore, solutions must be found to obtain sensible approximations to Wold representation. We shall demonstrate that such an approximation can be obtained by the ratio of two *finite* polynomials of order p and q, $L(B)\varepsilon_t \approx \frac{\theta q(L)}{\emptyset p(L)}$. We shall examine finite distributed lag models that provide the necessary approximations to Wold; typically, they are of very low order, namely, lags of 0, 1, or 2 will often prove quite effective to render a hopeless task of estimating a model with infinite parameters into a parsimonious forecasting model with just a few parameters!

6.4 *MA, AR & ARMA Processes*

Three common approximations to the Wold infinite series representation are moving average (*MA*), auto regression (*AR*) and their combination (*ARMA*). We examine each in turn.

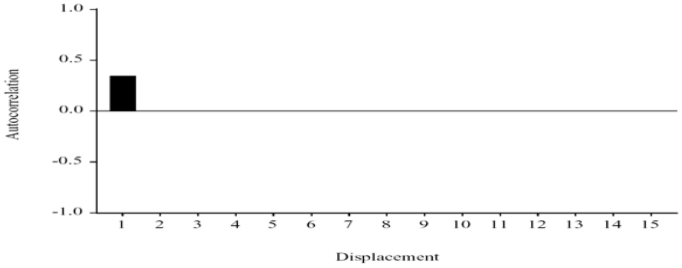
i. Moving Average Process

Since Wolds representation is based on shocks, the conditional mean of its infinite series provides a theoretical minimized loss forecast value. Let us start with an approximation for such an infinite series forecast in terms of its shocks or past lag values. Such a series is called a **Moving Average** or ***MA***. We wish to obtain an approximation to the conditional mean of Wold series by the simplest, first order *MA*:

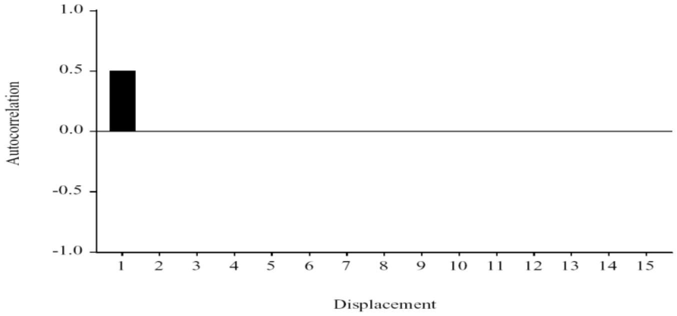
$$MA(1)\text{: } y_t = \varepsilon_t + \theta\, \varepsilon_{t-1} \;=(1+ \theta L)\, \varepsilon_t \;\;\&\; \varepsilon_\tau \sim WN(0, \sigma^2\,) \qquad (6.4.1)$$

(6.4.1) adds the moving average of the past error term observations to the mean of *y* to obtain a moving average of the past values of *y*. *MA*(1) series is a function of one-period unobservable shocks. θ measures the impact of a past shock on the current value of the series. One would expect small and large shocks to have very different Long Run impacts, but that is not so with the *MA*(1) model as evident from the graph below showing no LR effect..

Population Autocorrelation Function, MA(1) Process, θ=.4



Population Autocorrelation Function, MA(1) Process, θ=.95



We say *MA*(1) has **weak dynamics and short memory** regardless of the value of θ; this is expressed explicitly in an *abrupt cut off* in the *MA*(1) autocorrelation function in that all autocorrelations

$$\rho(\tau) = \frac{\gamma(\tau)}{\gamma(0)} \text{ beyond } \tau > 1 \text{ are zero.}$$

Moments of MA(1)

Unconditional mean: $E(y_t)=E(\varepsilon_t)+\theta E(\varepsilon_{t-1})=0$, using (6.4.1)

Unconditional variance: $Var(y_t)=Var(\varepsilon_t)+\theta^2 Var(\varepsilon_{t-1})=\sigma^2+\theta^2\sigma^2=\sigma^2(1+\theta^2)$, that is, for given$\sigma^2$, $Var(y_t)$ changes only with the size of the shock θ, e.g. 0.4 vs. 0.95.

Contrast these with the mean and variance conditional on past information set $\Omega_{t-1}=\{\varepsilon_{t-1}, \varepsilon_{t-2}, \ldots\}$.

Conditional mean: $\mathrm{E}(y_t|\Omega_{t-1})=E(\varepsilon_t|\Omega_{t-1})+\theta\, E(\varepsilon_{t-1}|\Omega_{t-1})= \theta\varepsilon_{t-1}$

where we have made use of the fact that Ω_{t-1} excludes ε_t, $E(\varepsilon_t|\Omega_{t-1})= E(\varepsilon_t)=0$. Moreover, $E(\varepsilon_{t-1})$ depends onΩ_{t-1} but because ε_t and ε_{t-1} are independent (serially uncorrelated), $\theta E(\varepsilon_{t-1}|\Omega_{t-1})=$

$\theta E(\varepsilon_{t-1}) = \theta\varepsilon_{t-1}$ where the last equality comes from that the fact that at time *t*, ε_{t-1} *is no longer random but a known constant*; hence, a constant conditional mean.

Conditional variance: $\mathrm{Var}(y_t|\Omega_{t-1}) = [(y_t - \mathrm{E}(y_t|\Omega_{t-1}))^2|\Omega_{t-1}] = \mathrm{E}(\varepsilon^2_t|\Omega_{t-1}) = \mathrm{E}(\varepsilon^2_t) = \sigma^2$

since ε is serially uncorrelated. Two points become clear from the contrast. The conditional mean adapts to the past information while the unconditional mean remains constant at zero; lags>1 do not affect the conditional mean, only the first lag does. The other point is that the conditional variance remains constant, unlike the unconditional one (affected by θ).

The *MA*(1) autocovaraince function at displacement $\boldsymbol{\tau}$ is

$$\gamma(\tau) = E[(\varepsilon_t + \theta\varepsilon_{t-1})(\varepsilon_{t-\tau} + \theta\varepsilon_{t-\tau-1})] = \begin{cases} \theta\sigma^2 & if \quad \tau = 1 \\ 0 & otherwise \end{cases} \tag{6.4.2}$$

(do the exercise question to see why).

The autocorrelation function is then the autocovrainace function scaled by the (unconditional) variance

$$\rho(\tau) = \frac{\gamma(\tau)}{\gamma(0)} = \begin{cases} \dfrac{\theta\sigma^2}{(1+\theta^2)\sigma^2} if\ \tau = 1 \\ 0 \qquad\quad otherwise \end{cases} \tag{6.4.3}$$

The main point is that all autocorrelations of *MA*(1) beyond τ=1 are zero, resulting in a sharp cut off in the function. This implies the *MA* process has limited ability to capture the impact of the past shocks on the current series, a feature that inhibits its potential as a forecasting model.

Invertiblility

MA(1) meets the covariance stationary conditions since it has constant unconditional mean and constant, finite unconditional variance, and its autocorrelation function depends on τ only. If in addition, **|θ|<1**, then *MA*(1) is also **invertible**. This means the series can be re-stated in terms of its current shock and *lagged values of the series*, and not of the lagged values of the shocks. It turns out that invertibility of *MA* enables us to obtain a more accurate and highly parsimonious model for dynamic effects. This leads to the *AR* presentation:

Given $\varepsilon_t = y_t - \theta\, \varepsilon_{t-1}$, we use successive lag periods of $\varepsilon_{t.}$expressed in terms of y_t *by backward substitution*:

Start with by inverting (6.4.1), so $\varepsilon_t = y_t - \theta\varepsilon_{t-1}$ and then substitute for $\varepsilon_{t-1}= y_{t-1} - \theta\, \varepsilon_{t-2}$:

$$\varepsilon_t = y_t - \theta[(y_{t-1} - \theta\, \varepsilon_{t-2})],\text{ so we now rewrite the series}$$

$$y_t = \varepsilon_t + \theta y_{t-1} - \theta^2 \varepsilon_{t-2},\text{ or } \varepsilon_t = y_t - \theta\{(y_{t-1} - [\theta\ (y_{t-2} - \theta\, \varepsilon_{t-3})]\}\rightarrow$$

$$y_t = \varepsilon_t + \theta y_{t-1} - \theta^2 y_{t-2} + \theta^3 \varepsilon_{t-3}\ldots$$

$$y_t = \varepsilon_t + \theta y_{t-1} - \theta^2 y_{t-2} + \theta^3 y_{t-3} - \theta^4 y_{t-4} + \theta^5 y_{t-5}\ldots\rightarrow$$

$$\varepsilon_t = y_t - \theta y_{t-1} + \theta^2 y_{t-2} - \theta^3 y_{t-3} + \theta^4 y_{t-4} - \theta^5 y_{t-5}\ldots) \rightarrow$$

$$\varepsilon_t =(1- \theta L + \theta^2 L^2 - \theta^3 L^3 + \theta^4 L^4 - \theta^5 L^5 \ldots)y_{t.=} (\theta)L.\ y_t \qquad (6.4.4)$$

Inside the (longer) bracket is an infinite geometric series; the sum of such a series is equal to $\frac{1}{1+\theta}$ if θ<1. Therefore, in lag operator notation

$$\varepsilon_t = \frac{1}{1+\theta L} y_t \qquad (6.4.5)$$

This is a ratio of two polynomial series in y_t, the numerator is a *degenerate* polynomial (because it is of degree 0- no lag) and the denominator is a polynomial of an infinite degree. Because θ raises to higher powers, this series converges only if $|\theta|$<1, or equivalently only if the *MA*(1) lag polynomial root L= $-\frac{1}{\theta}$ (obtained as the solution to $(1+\theta L)=0$) if θ<1 in absolute value).

MA of order q

The first-order MA is a special case of MA(q) of order q >1 with longer memory.

$y_t=\varepsilon_t+\theta_1\varepsilon_{t-1}+\theta_2\varepsilon_{t-2}+\ldots+\theta_q\varepsilon_{t-q}=\theta(L)\varepsilon t$ where $\theta(L)=1+Q_1L+\ldots+Q_qL^q$. The *MA*($q$) conditional mean depends on q lags rather than only the first lag as in *MA*(1), but still all autocorrelations beyond displacement $\tau > q$ are zero. Note that *MA* models are non-linear in parameters, see (6.4.4) & (6.4.5); the coefficient of the second lag of y_t is the squared of the coefficient of the first lag of y_t, etc. They must be estimated by iteration using some numerical estimation method.

ii. Autoregressive Model process

It is instructive to contrast the dynamics of the *MA* process with those of the *AR* process. *AR*(1) model explains the current series as a linear function of its past one-period lag and an additive shock

$y_t = \varphi y_{t-1} + \varepsilon_t$ and $\varepsilon_t \sim WN\,(0, \sigma^2)$, or in lag operator form

$$y_t - \varphi y_{t-1} = (1 - \varphi L) y_t = \varepsilon_t \qquad (6.4.6)$$

AR(1) meets invertibility conditions, but must satisfy covariance stationary state conditions in order to converge; The condition for autoregressive covariance stationary state is obtained by backward substitution similar to the above for MA:

$\varepsilon_t = y_t - \varphi\, y_{t-1}$ and by backward substitution as above we have:

$$y_t = \varepsilon_t + \varphi\, \varepsilon_{t-1} + \varphi^2\, \varepsilon_{t-2} + \dots \qquad (6.4.7) \text{ or}$$

$$y_t = \frac{1}{1 - \varphi L}\, \varepsilon_t \qquad (6.4.8)$$

This is a ratio of two polynomials, the numerator a degenerate of degree 0 inε_t, the denominator of degree *p* inε_t. This *MA* representation of *AR*(1) is a convergent series if $|\varphi|<1$ or equivalently if L= - $\frac{1}{\varphi}$. As with (6.4.5) for *MA*, (6.4.8) for *AR* requires $|\varphi|<1$ to converge; the parameters φ of its *AR* (∞) in (6.4.7) is said to be *absolutely summable*, that is $\sum_{j=0}^{\infty} |\varphi_j| < \infty$ for an *AR* polynominal of order *j*.

Moments of AR(1)

We employ *MA* representation of AR to work out unconditional moments.

Unconditional mean: $E(y_t) = E(\varepsilon_t) + \varphi E(\varepsilon_{t-1}) + \varphi^2 E(\varepsilon_{t-2}) + ... = 0$.

Unconditional variance:

$$Var(y_t) = Var(\varepsilon_t) + \varphi Var(\varepsilon_{t-1}) + \varphi^2 Var(\varepsilon_{t-2}) + \dots = \sigma^2 + \varphi^2 \sigma^2 + \varphi^4 \sigma^2 + \dots = \sigma^2(1 + \varphi^2 + \varphi^4 + \dots)$$

If $|\varphi|<1$ (using the rule for the sum of a geometric series) and

$$(1+\varphi^2+\varphi^4+\ldots) \approx \frac{1}{1-\varphi^2} \rightarrow Var(y_t)= \frac{\sigma^2}{1-\varphi^2}.$$

Conditional mean: $E(y_t|\Omega_{t-1})= \varphi E(y_{t-1}|y_{t-1})+\mathrm{E}(\varepsilon_t|y_{t-1})=\varphi y_{t-1}+0$

Conditional Variance: : $\mathrm{Var}(y_t|\Omega_{t-1})=\varphi^2\mathrm{Var}(y_{t-1}|\ y_{t-1})+\mathrm{Var}(\varepsilon_t|y_{t-1})=0+\sigma^2$.

That is, a conditional mean adapts to the changing information but the unconditional mean is constant (zero); conditional variance is constant while unconditional variance changes with the parameter φ depending on the number of lags, givenσ^2value.

To obtain the *AR*(1) autocorrelation function, we rely on the **Yule-Walker equation** according to which we can quickly work out γ(τ)if we know the initial period γ(τ-1):

$$\gamma(\tau\text{-}1).\ \varphi = \gamma(\tau) \qquad (6.4.9)$$

By this recursive method, all we have to do to obtain γ(τ) is to scale

γ(τ-1) by AR parameter φ. Start with $\gamma\ (0)= \frac{\sigma^2}{1-\varphi^2}$, then at τ=1→

$\gamma\ (1)= \varphi\frac{\sigma^2}{1-\varphi^2}$; at τ=2→ $\gamma\ (2)= \varphi^2\frac{\sigma^2}{1-\varphi^2}$, etc., so in general $\gamma\ (\tau)=\varphi^{\tau}\frac{\sigma^2}{1-\varphi^2}$ for τ=0, 1, 2, ….

The *AR*(1) autocorrelation function is then obtained by dividing through by $\gamma\ (0)= \frac{\sigma^2}{1-\varphi^2}$ to have

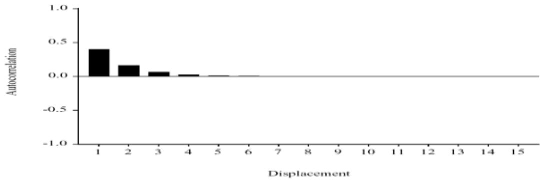
$$\rho(\tau) = \frac{\gamma(\tau)}{\gamma(0)} = \phi^{\tau},\ \tau=0,\ 1,\ 2,\ \ldots \qquad (6.4.10)$$

φ is called the **persistence parameter** since unlike *MA*(1), *AR*(1) autocorrelation approaches zero gradually (persists beyond displacement period) without an abrupt cut off and only at the limit). With $\varphi > 0$, the autocorrelation is one-sided and with $\varphi<0$, the decay oscillates. The *AR* of order one process displays a much bigger difference between 0.4 and 0.95 from *MA*(1) – Compare graphs, why? This is an indication of a longer memory of the *AR*(1) process, capturing more persistence dynamics. An *AR*(*p*) process owes its long-term memory to its relatively slow declining autocovariance function that decays *geometrically*.

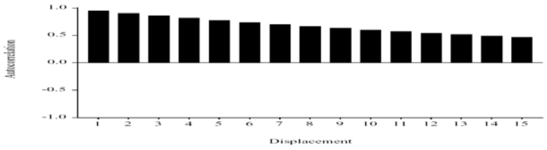
Key point: *forecasting requires linking the present series to its observable past. AR(1), where each lag of y incorporates the information on the previous lag, offers the necessary links using all information from the past, MA(1) does so only using the information from the last period because it excludes lagged periods of unobservable shocks>1. However, given invertibility, we can always generate a AR(1) process for MA(1)*. Compare (6.4.1) with (6.4.5) to see why; for more on the *MA* short and *AR* long memory and forecasting abilities, see section (6.5) forecasting with *MA & AR* processes.

Below shows lag effects with *AR*(1) for 0.4 *vs*. 0.95.

Population Autocorrelation Function AR(1) Process, φ=.4



Population Autocorrelation Function AR(1) Process,φ=.95



AR(p)

$$y_t=\varphi_1 y_{t-1}+\varphi_2 y_{t-2}+\ldots+\varphi_p y_{t-p} \text{ and } \varepsilon_t \sim \text{WN}(0, \sigma^2),$$

or in lag operator form

$$\varphi(L)y_t=1-\varphi_1 L+\varphi_2 L^2-\ldots-\varphi_p L^p)y_t=\varepsilon_t. \qquad (6.4.11)$$

A necessary (but not sufficient) condition for *AR*(*p*) to be covariance stationary is $\sum_{i=1}^{p}\varphi_i < 1$. The higher-order autocorrelation can oscillate with a greater variety of patterns. *AR*(*p*) is a generalization of *AR*(1) as

$$\theta(\text{L})\, y_t = (1 - \varphi_1 L - \varphi_2 L^2 - \ldots - \varphi_1 L^P)\, y_t = \varepsilon_t \text{ or}$$

$$y_t = \frac{1}{1-\varphi \text{L}}\, \varepsilon_t \qquad (6.4.12)$$

We approximate the Wold process with *AR*(1) using $\frac{1}{1-\varphi L}$. *AR*(1) is an infinite order series (see (5.4.7) & (5.4.8)) and yet has *only one parameter*, namely φ, and not an infinite number of parameters; it is obtained (approximated) from the ratio of two polynomials, a (degenerate) degree one polynomial in ε_t in the nominator, and a polynomial in the denominator of degree one in y_t.

iii. ***ARMA Process***

Combining *MA* and *AR* processes can result in more accurate and highly parsimonious approximations to the Wold representation. The **Autoregessive Moving Average *ARMA*(*p*,*q*)** model combines *AR*(*p*) and *MA*(*q*) processes, often in low orders, e.g. *p*=2 & *q*=1, to produce highly accurate and parsimonious forecasting models. For example, *AR*(5) may result in the same approximation accuracy as *ARMA*(2,1) with only three parameters to estimate rather than five if we combine *AR* with *MA*.

The simplest *ARMA*(1, 1) is

$$y_t=\varphi y_{t-1}+\varepsilon_t+\theta\varepsilon_{t-1}\text{ ; }\varepsilon_t\sim WN(0,\sigma^2\text{ }),\text{ or}$$

$$(1 - \varphi L)y_t=(1 +\theta L)\varepsilon_t \qquad (6.4.13)$$

The $|\varphi|<1$ condition for the *MA* representation for covariance stationary state leads to $y_t=\frac{1+\theta L}{1-\varphi L}\varepsilon_t$ and the invertibility condition for *AR* representation, $|\theta|<1$, again leads to $y_t\frac{1-\varphi L}{1+\theta L}=\varepsilon_t$. With the ARMA process, these conditions must be checked or satisfied. More generally the *ARMA*(*p*, *q*) process

$$y_t=\varphi_1y_{t-1}+\varphi_2y_{t-2}+\ldots+\varphi_py_{t-p}+\varepsilon_t+\theta_1\varepsilon_{t-1}+\theta_2\varepsilon_{t-2}+\ldots+\theta_q\varepsilon_{t-q}\text{ or}$$

in lag operator form (taking all y terms to the left), and using capital (*theta*)Θ forθ *s* and capital (*phi*) Φ for φ *s,* we have

$$\Phi L=1-\varphi_1L-\varphi_2L^2-\ldots-\varphi_pL^p\text{ and }\Theta L=1+\theta_1L+\theta_2L^2+\ldots+\theta_qL^q,$$

we can rewrite the equation as

$$\Phi Ly_t=\Theta L\varepsilon_t\rightarrow y_t=\frac{\Phi L}{\Theta L}\varepsilon_t\,.$$

The Wold representation with infinite parameters can now be approximated by a ratio of two finite-order lag operator polynomials, neither being degenerate, and results in very accurate models that often require estimating only a few parameters. The *ARMA* processes have constant unconditional mean but a time-varying conditional mean; the autocorrelation functions do not cut off at any particular displacement, but only damps down gradually.

For application, note the difference in the estimation methods required for *MA* and *AR* models by comparing (6.4.3) with (6.4.5). The *MA* (1) process (4) obtains forecast approximation to the Wold theorem by estimating an equation in which the coefficient of the second lag of *y* is the squared of the coefficient on the first lag of *y*, etc. after we invert the unobservable shocks to observable variables of the series. Such a non-linear model in parameters cannot be estimated by *OLS* and must employ a non-linear numerical minimization method. By contrast, the lag coefficients of *AR*(1) process in (6.4.5), or its general version in (6.4.9), are simply the linear in parameter for each lagged variable; therefore *AR* models can easily be estimated by *OLS*.

Example for Canadian Employment series reports several sets of estimates for MA, AR and ARMA. The model starts as an *ARMA*(3, 1), but roots of *MA*(1) and *AR*(3) are roughly of the same size (- .95) , so cancel each other, simplifying the model selected by the Schwarz criteria as the most parsimonious *AR*(2); results shown in table 67.1 here turn out to perform best.

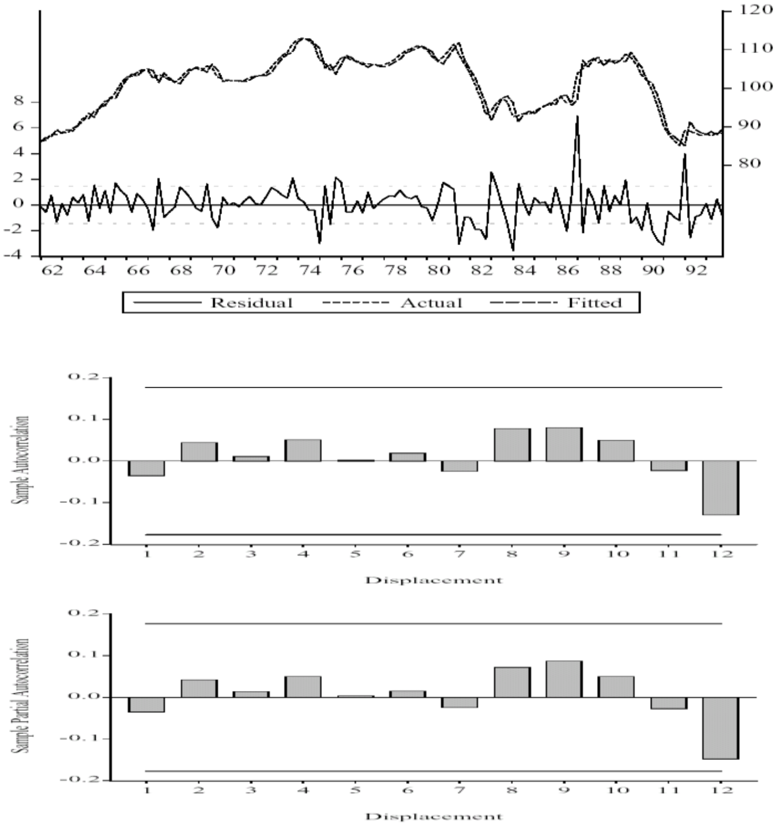
Table 6.1 Dependent Variable is CANEMP; Sample: 1962:1-1993:4; included observations:

128 Convergence achieved after 3 iterations

Variable	Coefficient	Std. Error	t Statistic	Prob.
C	101.2413	3.399620	29.78017	0.0000
AR(1)	1.438810	0.078487	18.33188	0.0000
AR(2)	0.476451	0.077902	6.116042	0.0000

R squared	0.963372	Mean dependent var	101.0176
Adjusted R squared	0.962786	S.D. dependent var	7.499163
S.E. of regression	1.446663	Akaike info criterion	0.761677
Sum squared resid	261.6041	Schwarz criterion	0.828522
Log likelihood	227.3715	F statistic	1643.837
Durbin Watson stat	2.067024	Prob(F statistic)	0.000000

Inverted AR Roots	.92	.52



Estimates suggest good impact for lags, and the residuals appear to be white noise (zero mean), and the correlogram plots back that conclusion as all lags are within 2 s.d from the mean (5% confidence) bands.

6.5 *Forecasting with AR, MA& ARMA processes*

The history of a series y_T is contained in its information set; and there are two ways of expressing it, either in terms its own available past history, y_{T-j}, or, in terms of its present and past shocks, ε_t an d ε_{t-j}. As long as the series is covariance stationary and invertible, we can infer history of ε_t from the history of y_T, and history of y_T from that of ε_t . Therefore the information set at time T contains the present and lagged values of both y_T and ε_t in $\mathbf{\Omega}_T = \{y_t , y_{t-1}, y_{t-2},\dots, \varepsilon_t, \varepsilon_{t-1}, \varepsilon_{t-2},\dots\}$. Based on Ω_T, the **optimal forecast** at time $(T+h)$ is one that minimizes the forecast **expected loss**. This turns out to be equal to the mean of the series conditional on Ω_T, that is $E(y_{T+h}|\Omega_T)$. We find the best linear approximation to the conditional mean of a series to order to obtain **linear forecasts**, or **linear projections** $P(y_{T+h}|\Omega_T)$.

i. Optimal forecasts with MA process

Since the Wold representation is based on past innovations, let us start with the *MA(2)* approximation for such an infinite series as an example

$$y_t = \varepsilon_t + \theta_1\varepsilon_{t-1} + \Theta_2\varepsilon_{t-2} + u_t \tag{6.51}$$

where u_t is a random error term. Using (6.5.1), we operationalize forecasting by replacing unknown parameters with estimates and unobservable innovations with residuals.

To obtain 1-step-ahead projection forecast (*T*+1) at time *T*, first re-write the (6.5.1) process for (*T*+1) period, then set its future innovations u_t equal to 0. $Y_{T+1} = \varepsilon_{T+1} + \theta_1\varepsilon_T + \theta_2\varepsilon_{T-1}$ when the information available at T is $Y_{T+1},T = P(y_{T+h}|\Omega_T) = \theta_1\varepsilon_T + \Theta_2\varepsilon_{T-1}$ since all future innovations $E(\varepsilon_{T+1})=0$. The forecast error is therefore $(e_{T+1}, T)=(Y_{T+1}-Y_{T+1}, T)= \varepsilon_{T+1}$ (white noise error). Hence forecast variance is $\sigma_1{}^2=\sigma^2$.

For 2-step-ahead forecast: $Y_{T+2}= \varepsilon_{T+2} + \theta_1\varepsilon_{T+1} + \Theta_2\varepsilon_T$. Since both $E(\varepsilon_{T+1})=0$ and $E(\theta\varepsilon_{T+1})=0$, the forecast projection at time *T* is therefore $Y_{T+2},T=P(y_{T+h}|\Omega_T)=\Theta_2\varepsilon_T$. Therefore, the forecast error is $e_{T+2}, T=(Y_{T+1}-Y_{T+1}, T)= \varepsilon_{T+2} + \Theta_1\varepsilon_{T+1}$.

The *MA*(1) variance is $\sigma_2{}^2=\sigma^2(1+\theta_1{}^2)$, using property 2 of the variance, see appendix, and

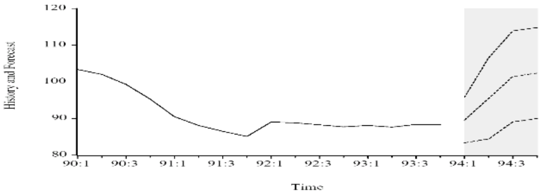
$Var(\varepsilon_{T+1}) = Var(\varepsilon_{T+2})=\sigma^2$.

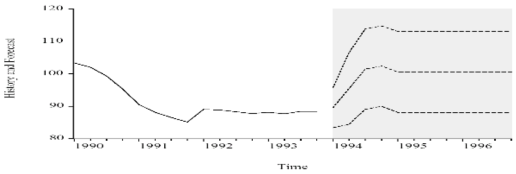
The 3-step-ahead is **unforecastable** with the *MA*(2) process since the right-hand side

$Y_{T+3}, T= \varepsilon_{T+3} + \theta_1\varepsilon_{T+2} + \Theta_2\varepsilon_{T+1}$ are **all 0** at (*T*+3) and therefore

$(e_{T+3}, T)=(y_{T+3},T)=0$ **for all τ>2**. In general, the error with *MA*(2) for a *h*-step-ahead forecast remains unchanged for all subsequent steps. $e_{T+h}, T=(Y_{T+h}, T= \varepsilon_{T+h} + \theta_1\varepsilon_{T+h-1} + \Theta_2\varepsilon_{T+h-2})$ for **all 0** *h*>2; the forecast error is $\sigma_h{}^2=\sigma^2(1+\theta_1{}^2+ \theta_2{}^2)$ as with the 3; that is the error remains the same even for *h*-step-ahead error. This is a dramatic display of the implication that MA process has a short memory.

Example: Canadian Employment forecast with a *MA*(4) process. Note that changes in employment time series is expected to revert to its long-run mean slowly and with a time-lag. Figure below shows the last historical data for 1993.4 to be well below its mean, and yet the forecast displays a sharp rise that is unnatural for a variable like employment. This is a manifestation of a *MA* process's short memory. That is because MA(4) has left out the portion of the lag effects > 4, thus estimating a bigger impact for the first 4 lags. Figure confirms that 4-step-ahead forecast with *MA*(4) is unable to capture persistence as all forecasts beyond lag 4 are 0.





ii. *Optimal forecast with AR process and the Chain Rule of Forecasting*

A simple recursive method called the **chain rule of forecasting** is available to obtain forecasts from an autoregressive process. The rule is as follows: first construct the optimal 1-step-ahead forecast (*T*+1) from the estimates obtained with observations up to time *T*, then construct the optimal 2-step-ahead forecast based on the optimal 1-setp-ahead forecast already available, and repeat the process to obtain a *m*-step-ahead optimal forecast for *T*+*m* time from the forecast available from *T*+(*m*-1) period, etc. Suppose *AR*(1)

$$y_t = \phi\, y_{t-1} + \varepsilon_t \ \ \& \ \ \varepsilon_\tau \sim \mathrm{WN}(0, \sigma^2) \qquad (6.5.2)$$

Using the estimates from this equation we construct the optimal 1-step-ahead forecast, by noting that all future innovations are zero.

For example, write out (*T*+1), (*T*+2) and (*T*+3)-step-ahead forecasts:

1-step ahead forecast: first re-write (6.5.2) process for (*T*+1), then obtain its projection forecast by setting its future innovation equal to 0:

$Y_{T+1} = \varphi\, y_T + \varepsilon_{T+1}$ and its projection forecast $y_{t+1}, T = \varphi\, y_T$

2-step-ahead-forecast:$Y_{T+2} = \varphi\, y_{T+1} + \varepsilon_{t+2}$ and its projection forecast

$(y_{t+2}, T) = \varphi\, y_{T+1}, T = \varphi(\varphi\, y_t) = \varphi^2 y_t$ (substituting Y_{T+1} with φy_T from the 1-step-ahead forecast.)

3-step-ahead-forecast: $Y_{T+3} = \varphi\, y_{T+2} + \varepsilon_{T+3}$ and its projection forecast

$(y_{T+3}, T) = \varphi\, y_{T+3}, T = \varphi(\varphi^2\, y_t) = \varphi^3 y_t\ldots..$

$Y_{T+h} = \varphi\, y_{T+h-1} + \varepsilon_{T+h}$ and its projection forecast
$(y_{t+3}, T = \varphi\, y_{t+h-1}, T) = \varphi(\varphi^{h-1}\, y_t) = \varphi^{h} y_t$
Note for the *AR*(1) only the most recent *y* is used to construct optimal forecasts, that is *the entire set of h-period-ahead forecasts can be expressed in terms of only* φ *and* y_T*, both of which are known from period T .*

This AR method of forecasting allows recursive build-up of forecasts for **any further period**, displaying the longer memory capacity and superior ability to capture and make use of lag history of a series in forecasting its future values. AR forecasts therefore do not display abrupt cut-off forecast beyond displacement τ characteristic of the MA process, though they shrink and decline gradually.

iii. *Forecast with* the *ARMA process*

The ARMA process combines the lag structures of MA and AR models to further improve forecast values.

Take the simple *ARMA*(1, 1) for example:

$$y_t = \phi\, y_{t-1} + \varepsilon_t + \Theta\varepsilon_{t-1}$$

At time *T*+1, $y_{T+1} = \phi\, y_T + \varepsilon_{T+1} + \Theta\varepsilon_T$; its forecast projection is

$y_{T+1}, T = \phi\, y_T + \Theta\varepsilon_T$ since ε_{T+1}=0.

At time *T*+2, $y_{T+2} = \phi\, y_{T+1} + \varepsilon_{T+2} + \Theta\varepsilon_{T+1}$; its forecast projection is

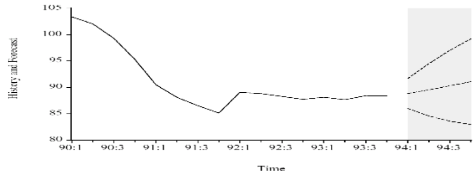
$y_{T+2}, T = \phi\, y_{T+1,} T$, since $E(\varepsilon_{T+2}) = \Theta E(\varepsilon_{T+1}) = 0$.

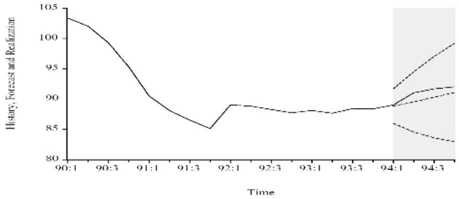
Upon substitution for the 1-step-ahead forecast, this leads to the forecast for (*T*+2) as

y_{T+2} , $T = \phi\,(\phi\, y_T + \Theta\varepsilon_T) = \phi^2 y_T + \phi\ \Theta\varepsilon_T$, substituting for y_{T+1} at T+1 from above.

Continuing in this manner, in general, y_{T+h} , $T = \phi\ y_{T+h-1}$, T for all h>1.

Example: Canadian Employment forecasts were obtained with *AR* and *ARMA* processes, and the model selected as best was the *AR*(2) process. The different nature of the autoregressive process is clear from the figure below that shows 12-step-ahead projection forecasts with a much longer forecast-horizon. Note that there is no sharp rise in the forecasts; consistent with slow adjustment of employment to its long-run mean value with time lags. Figure below illustrates comparison of a 4-quarter-ahead projection forecast and the realization; note the mean appears drastically smaller, suggesting a more accurate forecast.





Readings

For textbook discussion, see Diebold (2006, chapters 3, 7, 8 and 9) and Pesaran (2015, chapter 17); Gonzalez-Rivera (2013, chapters 6 and 7). Jorgenson (1966) introduced the rational distributed lags model from engineering into economics.

Chapter 6 Forecasting with *MA*, *AR* Exercises

Q6.1 Part 1:

For each of the following, determine whether $\{y_t\}$ represents a stable process. Determine whether the characteristic roots are real or imaginary and whether the real parts are positive or negative.

a. $y_t - 1.2y_{t-1} + 0.2y_{t-2}$

b. $y_t - 1.2y_{t-1} + 0.4y_{t-2}$

c. $y_t - 1.2y_{t-1} - 1.2y_{t-2}$

d. $y_t + 1.2y_{t-1}$

e. $y_t - 0.7y_{t-1} - 0.25y_{t-2} + 0.175y_{t-3} = 0$

[*Hint*: $(x - 0.5)(x + 0.5)(x - 0.7) = x^3 - 0.7x^2 - 0.25x + 0.175$]

Part 2: Write each of the above equations using lag operators. Determine the characteristic roots of the inverse characteristic equation.

Q6.2 Fill in the missing steps in (6.4.2) for taking expectation required to obtain the final results.

Q6.3 Given an initial condition for y_o, find the solution for y_t. Also find the s-step-ahead forecast $E_t y_{t+s}$

a. $y_t = y_{t-1} + \varepsilon_t + 0.5\varepsilon_{t-1}$.

b. $y_t = 1.1y_{t-1} + \varepsilon_t$

Q6.4 Download *aic sic and forecasting.dta* set containing the US time-series of GDP.

a. Fit *AR*(1) for *GDP* and obtain Schwarz(sic) and Akaike (aic) critera for model selection.

b. Now fit *AR*(2)-*AR*(4) for *GDP* and select the best model based on the smallest Schwarz(sic) and Akaike (aic) values.

c. Use the selected model to obtain manually the forecasts for the next four periods using the two final observations in the data set for first forecast and then update.

Q6.5 Download *WPI_US.dta* set of the US wholesale price index.

a. Fit an *AR*(1) to *lwpi* series. Explain the model implemented.

b. Now fit a *MA*(1) process to *ln_wpi* and compare the outcome with a. above

c. Fit an *ARMA* (2,1) model to *ln_wpi* series and comment on the outcome.

d. Obtain a one-step-ahead forecast for *ARIMA* (2,0,1) and plot the forecast values against the actual values of *ln_wpi*

Chapter 7 Stationary Series, ARDL, VAR & Impulse-Response Function

Introduction

In a time-series **regression model** we have a time-series of one (dependent or endogenous) variable explained by the time-series of one or more (independent or exogenous) variable(s) that may also include in addition its own lagged values. The *distributed-lag model* excludes lags of the dependent variable as explanatory variables

$$y_t = \beta_0 + \delta_1 x_{t-1} + \delta_2 x_{t-2} + \delta_3 x_{t-3} + \ldots + \varepsilon_t \qquad (7.1)$$

There are a number of problems with this essentially *ad hoc* model. If the number of lags N_x is large, loss of degrees of freedom will violate the forecasting principle of parsimony. Moreover, the various lagged values of x are likely to be severely multicollinear, making coefficient estimates imprecise. The alternatives approach extensively explored in literature are autoregressive models, or the relationships among a system of such models. We examine these alternative models in this chapter.

7.1 *Dynamic Model with Lagged Dependent Variables*

(7.1) leaves out the impact of y_{t-i} on y_t as arbitrary without testing for the presence of y_{t-i} impact. The rational distributed lag (ARDL) models take that impact into account by $y_t = \alpha + \frac{\beta(L)}{\lambda(L)} x_t + \mu_t$ where $\frac{\beta(L)}{\lambda(L)}$ is the ratio of two polynomials in y_t and x_t. It turns out that inclusion of lagged dependent values in (7.1) absorbs residual serial correlation, often resulting in substantially improved forecasts for y_t.

$$y_t = \beta_0 + \sum_{i=1}^{N_p} \lambda_i y_{t-i} + \sum_{i=1}^{N_q} \delta_i x_{t-i} + \varepsilon_t \text{, or } \lambda(\mathrm{L}) y_t = \beta_0 + \delta(L) x_t + \varepsilon_t \qquad (7.1.1)$$

This is the *Autoregressive Distributed Lag Model or the* ***ARDL(p, q)***. In contrast with equation (7.1), the presence of y_{t-1} on the RHS makes equation (7.1.1) a **dynamic** model. $\delta(L)$ is called the *transfer function* because it shows how the movement in exogenous z_t affects, or transfers, the endogenous variable y_t; the coefficients of δ (L) are called transfer function weights.

Note that the model can be expressed, by backward substitution, in terms of moving average errors to represent the rational distributed lag model:

$$\lambda(L)y_t= \beta_0\,\lambda(1)+\delta\,(L)x_t+ \lambda(L)v_t\ ;\ \varepsilon_t= \lambda(L)v_t$$

The ARDL model has been revived lately because of recent developments in time-series analysis that are more easily modeled as an ARDL compared with the rational distributed lag model; by selecting p and q to be sufficiently large, we can obtain a good approximation to the rational distributed lag model.

The simplest is the *ARDL* (1, 1) dynamic model

$$y_t=\alpha_0+\beta_0 x_t+\lambda y_{t-1}+\varepsilon_t. \qquad (7.1.2)$$

To find out the impact of the lags on the dependent variable, substitute for lagged *y* in (7.1.2):

$y_{t-1}=\alpha_0+\beta_0 x_{t-1}+\lambda y_{t-2}+\varepsilon_{t-1}$; therefore,

$$y_t=\alpha_0+\beta_0 xt+\lambda(\alpha_0+\beta_0 x_{t-1}+\lambda y_{t-2}+\varepsilon_{t-1})+ \varepsilon_t =(\alpha_0+\lambda\alpha_0) +\beta_0 xt + \lambda(\beta_0 x_{t-1}+\lambda y_{t-2}+\varepsilon_{t-1})+\varepsilon_t.$$

Substitute again for y_{t-2} in this last equation, then

$$y_t=\alpha_0^*+\beta_0 xt +\lambda\beta_0 xt_{-1}+\lambda^2\beta_0 xt_{-2} +\lambda^3 y_{t-2}+\ldots)+\varepsilon_t^* \qquad (7.1.3)$$

where * indicate compound intercept or residuals after collecting terms.

As long as λ is between 0 and 1, the coefficients smoothly decline, more quickly the smaller (away from 1) the λ value is**,** as in Figure 7.1. That is, the impact of the lags in a dynamic model continuously declines. Formally, the ARDL model is stable if all the roots of the *pth* order polynomial equation

$$\lambda(z)= 1 + \lambda_1 z + \lambda_2 z^2+ \ldots + \lambda_p z^p =0$$

lie outside the unit circle, namely if |z|>1; unstable if $\lambda(1)= 0$; see Appendix on the roots of a polynomial.

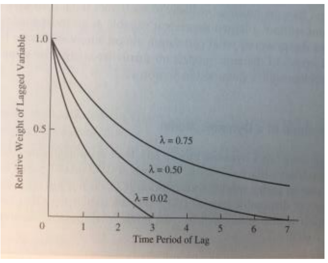


Figure 7.1 *Dynamic Models with Geometric Weighting Schemes*

The *ARDL* (*p*, *q*) can be estimated by the OLS method, however, consistency requires z_t follows a covariance stationary process as $T \to \infty$. This condition is guaranteed if all the roots of the polynomial $\lambda(z)$ lie outside the unit circle; it is not however sufficient for consistent estimation. In addition,), ε_t must be serially uncorrelated to ensure that $\sum z_t \varepsilon_t / T$ converges to a zero vector as $T \rightarrow \infty$.

7.2 *VAR Regression Model: Introduction*

In economics, typically all variables are endogenous and partially affected by most other variables; often we may not be secure if a variable is exogenous. This is specially the case with time-series for a given set of variables since the past values of the lagged dependent variables can influence their current values in addition to the influences of the current and lagged values of the explanatory variables, making all variables interdependent, all being explanatory and dependent variables to each other simultaneously. The general approach would be to extend the transfer function approach and treat all variables symmetrically as endogenous in a system of interdependent equations. Consider the bivariate case

$$\boldsymbol{y_t = b_{10} - b_{12}z_t + \gamma_{11}y_{t-1} + \gamma_{12}z_{t-1} + \varepsilon_{yt}} \qquad (7.2.1)$$

$$\boldsymbol{z_t = b_{20} - b_{21}y_t + \gamma_{21}y_{t-1} + \gamma_{22}z_{t-1} + \varepsilon_{zt}} \qquad (7.2.2)$$

where $\varepsilon_{yt}$and ε_{zt} are assumed white nose errors; uncorrelated with each other. This structure allows z_t and y_t to affect each other, e.g. $b_{12}z_t$ is the contemporaneous effect of a unit change in z_t on y_t . Note that if for example, $b_{12}z_t \neq 0$, then ε_{zt} has an indirect contemporaneous effect on y_t. As it stands, this model cannot be estimated by the *OLS* because of simultaneous equation bias resulting from the correlation of the regressors and the error terms. We can transform the equations into a more easily estimable system written in compact form as

$$Bx_t = \Gamma_0 + \Gamma_1 x_{t-1} + \varepsilon_t \qquad (7.2.3)$$

where $B=\begin{bmatrix} 1 & b_{12} \\ b_{21} & 1 \end{bmatrix}, xt = \begin{bmatrix} y_t \\ z_t \end{bmatrix}$, $\Gamma_0 = \begin{bmatrix} b_{10} \\ b_{20} \end{bmatrix}$, $\Gamma_1 = \begin{bmatrix} \gamma_{11} & \gamma_{12} \\ \gamma_{21} & \gamma_{22} \end{bmatrix}$, and $\varepsilon_t = \begin{bmatrix} \varepsilon_{yt} \\ \varepsilon_{zt} \end{bmatrix}$

Pre-multiplication of (7.2.3) by B^{-1} matrix results in a system of equations with the vector of endogenous variables in x_t as a function of lagged variables only, excluding contemporaneous RH variables.

$$x_t = A_0 + A_1 x_{t-1} + e_t \qquad (7.2.4)$$

where $A_0= B^{-1}\Gamma_0$, $A_1= B^{-1}\Gamma_1$, and $e_t=B^{-1}\varepsilon_t$

We can rewrite (7.2.4) as a of two-equations in an equivalent form to (7.2.1) and (7.2.2) as

$$\boldsymbol{y_t=a_{10} + a_{11}y_{t-1} + a_{12}z_{t-1} + e_{yt}} \qquad (7.2.5)$$

$$\boldsymbol{z_t=a_{20} + a_{21}y_{t-1} + a_{22}z_{t-1} + e_{zt}} \qquad (7.2.6)$$

where the variance-covariance matrix of e_y and e_{zt} is defined as

$$\sum=\begin{bmatrix} var(e_{1t}) & cov(e_{1t,}e_{2t)} \\ cov(e_{1t,}e_{2t}) & var(e_{2t}) \end{bmatrix} = \begin{bmatrix} \sigma_1^2 & \sigma_{12} \\ \sigma_{21} & \sigma_2^2 \end{bmatrix}$$

since all elements of ∑ are independent of time.

The critical point to note is that the error terms of (7.2.5) and (7.2.6) are now transformed into composites errors of the two innovation shocks, ε_{yt} and ε_{zt} , because the new vector is $e_t = B^{-1}\varepsilon_t$. However, since ε_{yt} and ε_{zt} are white-noise processes, both e_y and e_{zt} are also white-noise errors and computed as (using the rule of matrix inversion)

$$\boldsymbol{e_{1t}=(\varepsilon_{yt} - b_{12}\varepsilon_{zt}) / (1 - b_{12}b_{21})} \qquad (7.2.7)$$

$$\boldsymbol{e_{2t}=(\varepsilon_{zt} - b_{21}\varepsilon_{yt}) / (1 - b_{12}b_{21})} \qquad (7.2.8)$$[10]

We call the system of equation (7.2.1) and (7.2.2) a *structural VAR*, vector autoregression, or the primitive system and the (7.2.3) and (7.2.4) system of equations a *VAR in standard form*. Once again, stability requires that $|a_1| > 1$, or the roots of its characteristic equation lie outside the unit circle.

There are two notable features to the above *VAR* model. First, it assumes that y_t and z_t are dynamically related but they are *not contemporaneously* related to each other. That is, only lagged values of y_t and z_t affect the current values of y_t and z_t. Second, we assume that the current period error terms are uncorrelated, that is, they are contemporaneously uncorrelated. Both equations are assumed to have *WN* errors, and the lagged disturbances can be correlated to transmit shocks from one equation to other, but they may also be uncorrelated, in which case σ_{12}=0. If the model based

[10] The inverse of the matrix B (of the structural model's coeficients) is the product of the inverse of its determinant, $\frac{1}{1-b_{12}b_{21}}$, and its adjoint $\begin{bmatrix} 1 & -b_{21} \\ -b_{12} & 1 \end{bmatrix}$; therefore $\begin{bmatrix} e_{yt} \\ e_{zt} \end{bmatrix}$= {$\frac{1}{1-b_{12}b_{21}}$. ($\begin{bmatrix} 1 & -b_{21} \\ -b_{12} & 1 \end{bmatrix}$.$\begin{bmatrix} \varepsilon_{yt} \\ \varepsilon_{zt} \end{bmatrix}$)} renders (7.2.7) and (7.2.8).

on the two assumptions above represent the true dynamic system, and y_1 and y_2 are have coefficients less than 1 in absolute value, with zero-mean contemporaneously uncorrelated residuals, then equation-by-equation estimation of the *VAR* by *OLS* gives estimates that are consistent and asymptotically efficient; even though the lagged errors are correlated, the *SURE* regression cannot improve the efficiency of the estimates since all regressions have identical *RH* variables. We can also estimate the variance-covariance of the *VAR* in standard form, and select *VAR* forecast models based on minimization of *predictive* errors, or minimized *AIC* and *SIC* duly adjusted for the simultaneous equation context and available in software such as Stata.

One important remaining question is whether we can use the *VAR* in standard form to recover the structural *VAR* parameters since the latter cannot be directly estimated due to the correlation between y_t and e_{yt},, and z_t *and* e_{zt}. Since the structural *VAR* contains more parameters than the *VAR* standard, without imposing restrictions on the structural *VAR* system parameters that are not identified from the *VAR* estimates. For example, (7.2.5) and (7.2.6) have six coefficients, plus var (e_{1t}), var(e_{2t}), and cov(e_{1t}, e_{2t}), nine parameters in total. However, the structural *VAR* system (7.2.1) and (7.2.2) has 10 parameters (error terms are assumed uncorrelated with each other). Therefore, the structural *VAR* parameters are under-identified; identification requires imposing a restriction on the structural *VAR* equations. One strategy is to use a *recursive s*ystem of the structural *VAR* by imposing restriction. For example, economic theory may support an asymmetric restriction that b_{12}=0, i.e. that y_t has not contemporaneous effect on z_t, its only effects are through its lag values, while z_t, has a contemporaneous effect on y_t. Then the system is exactly identified, and the restriction manifests itself so that ε_{yt} and ε_{zt} affect the contemporaneous value of y_t, but only ε_{yt} shocks affect contemporaneous values of z_t, that is, the observed values of e_{2t} are totally the result of pure shocks to the $\{z_t\}$sequence. Imposing the restriction on the computed errors for the *VAR* in standard from (7.2.7) and (7.2.8) leads to

$$E_{2t}=\varepsilon_{yt}-b_{12}\varepsilon_{zt}$$

$$E_{1t}=\varepsilon_{zt}$$

The decomposition of the residuals by this asymmetric, triangular method is known as a **Choleski** decomposition.

7.3 *The Impulse-Response Function*

We must address additional questions on the duration of lag effects if we are interested in how shocks to one variable are transmitted to other variables in the *VAR* system, as well as the length of their duration, and their persistence. In other words, we are interested to know how shocks affect the adjustment path of *VAR* variables. To that end, we rely on a *moving average representation o*f the *VAR* system of equations, expressing the sequence of each time-series process $\{x_t\}$ purely in terms of the current and past values of its own shocks, and those of the other *VAR* processes.

Univariate path: Start with the simple case of a univariate series, subject to a shock of size *v* at *t*=1.

$$y_t = \rho y_{t-1} + v_t \qquad v_t \sim WN(0, \sigma^2) \tag{7.3.1}$$

Since we are interested in the dynamic path, the starting value of *y* before the shock is irrelevant; therefore, $y_0 = 0$. To simplify further, assume there are no additional shocks during periods *t*>1.

Following the shock at *t*=1, that is $v_2=v_3=\ldots=0$, the value of y_1 is

at *t*=1, $y_1 = \rho y_0 + v_1 = v$;

at *t*=2, $y_2 = \rho y_1 = \rho v$;

at *t*=3, $y_3 = \rho y_2 = \rho(\rho y_1) = \rho^2 v; \ldots$ (7.3.2)

Thus, the time-path of following the shocks is $\{v, \rho v, \rho^2 v, \ldots\}$ and is known as the *impulse-response function* and the coefficient values $\{1, \rho, \rho^2, \ldots\}$ as *multipliers*. Here the impulse is the shock and the response is the change. As it becomes clear later, it is more convenient to define (or normalize) impulse in terms of units of standard deviations, rather than unit shocks to avoid measurement problems. Restated (7.3.1) in the standard Moving Average form as

$$y_t = b_0\varepsilon_t + b_1\varepsilon_{t-1} + b_2\varepsilon_{t-2} + \cdots = \sum_{i=0}^{\infty} b_i\varepsilon_{t-i}; \quad \varepsilon_t \sim WN(0, \sigma^2) \tag{7.3.3}$$

The coefficient of ε_t is usually normalized to unity but in (7.3.3) it is stated more generally as b_0, though this introduces an ambiguity in that we can always divide and multiply each ε_t term in (7.3.3) by an arbitrary constant *m* to obtain an equivalent model but different parameters and innovation shocks. Normalization by *m*=1 results in the standard *I-R* function (7.3.3), with $b_0 * 1$. However, setting *m*= σ turns out to be a particularly helpful normalization. Let us normalize the equation by dividing throughout by $\frac{\sigma}{\sigma}$:

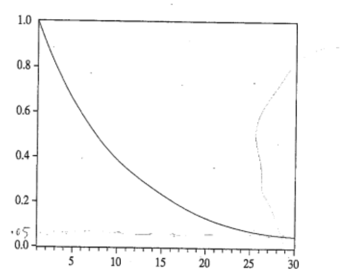
$$y_t = b_0\sigma\left(\frac{1}{\sigma}\varepsilon_t\right) + b_1\sigma\left(\frac{1}{\sigma}\varepsilon_{t-1}\right) + b_2\sigma\left(\frac{1}{\sigma}\varepsilon_{t-2}\right) + \cdots \qquad (7.3.4)$$

set $b_i{}' = b_i\sigma$ and $\varepsilon_i' = \frac{\varepsilon_t}{\sigma}$, then one standard deviation σ shock to ε_t converts (7.3.3) into

$$y_t = b_0{}'\varepsilon_t{}' + b_1{}'\varepsilon_{t-1}{}' + b_2{}'\varepsilon_{t-2}{}' + \cdots ; \qquad \varepsilon_t{}' \sim WN(0, 1) \qquad (7.3.5)$$

because $Var(\frac{\varepsilon_t}{\sigma})$ = $\frac{1}{\sigma^2}$ var(ε_t) = $\frac{\sigma^2}{\sigma^2}$ = 1. Now $b_i{}'$ parameter measures one-standard deviation shock to $\varepsilon_t{}'$, that is the contemporaneous effect of a unit shock to $\varepsilon_t{}'$ at time *t*=1; $b_1{}'$ multiplies $\varepsilon_{t-1}{}'$ to give the effect of a unit standard deviation shock one period later, etc. Hence, the impulse-response multipliers $\{b_0{}',\ b_1{}',\ b_2{}', \ldots\}$ track the complete dynamic response of y.

Example: suppose $\rho = 0.9, \&\ v = 1$, then y will be{1, 0.9, 0.81, … }, so y initially rises by the shock (to y=1) and then gradually returns to its original value.



Impulse responses for an AR(1) model $y_t = 0.9y_{t-1} + e_t$ following a unit shock.

7.4 *The VAR Multivariate Impulse-Response Function*

Generalization of (7.2.1) to the multivariate case follows the same method but more than one shock must be tracked. Now consider the impulse-response function with two time-series with a two-equation *VAR* system of stationary variables expressed in the standard moving average representation; and, just like the univariate case, we also convert the shocks into units of the standard deviations.

$$y_{1t} = \varepsilon_{11} + \varphi_{11}\varepsilon_{t-1} + \varphi_{12}\varepsilon_{t-1} + \cdots ; \qquad \varepsilon_{1_t} \sim WN(0, \sigma_1^2)$$

$$z_t = \varepsilon_{21} + \varphi_{21}\,\varepsilon_{t-1} + \varphi_{22}\,\varepsilon_{t-1} + \cdots ; \qquad \varepsilon_{2_t} \sim WN(0, \sigma_2^2)$$

$$\text{cov}\,(\varepsilon_{1_t}, \varepsilon_{1_t}) = \varepsilon_{12}$$

Just as in the univariate case, the transformed normalization by σ results in the current innovations having unit coefficients; this is called a Vector Moving Average or *VMA* transformation of *VAR*.

Now we have **four** Impulse-Response functions and therefore, track four sequences of shocks: the effect of a shock to *y* on the time-paths of *y* and *x*, and the effect of a shock to *x* on the time-paths of *y* and *x*. This is a more complicated mechanism since it has to allow for (a) interdependent dynamics, and (b) has to identify the correct shock that in fact is not observable from the data. The time paths of the innovation shocks to each process cannot, however, be determined from solely from the variance-covariance of the *VAR* standard errors, and as previously discussed, requires imposing parameter restrictions on the model above. This is known as the **identification problem**. We can rely on the restriction generated by the *recursive ordering* of the Choleski decomposition by, for instance, setting b_{21}=0 so as to rule out any contemporaneous effect of y_t on z_t ; therefore,

$$e'_{1t}=\varepsilon_{yt} - b_{12}\varepsilon_{zt}$$

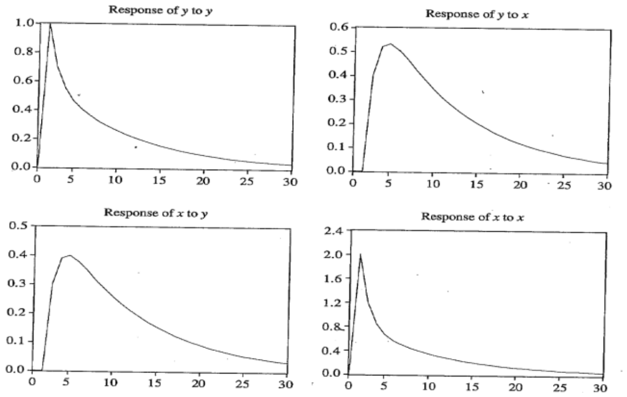
$$e'_{2t}=\varepsilon_{zt}$$

$$cov\,(e'_{1t}\,,\, e'_{2t})=0$$

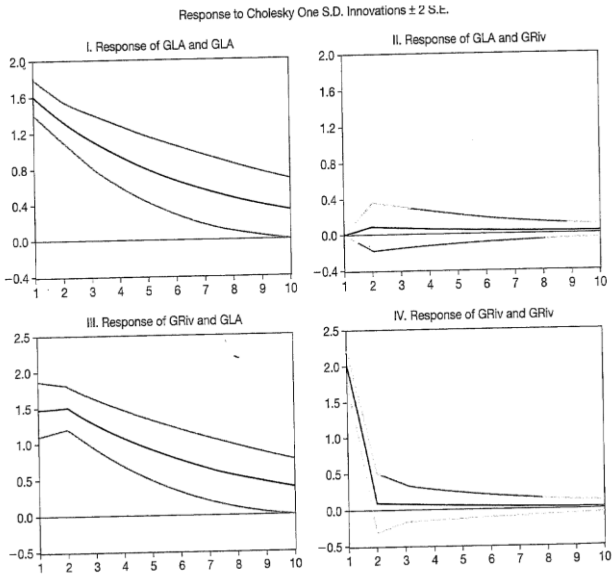
Such restrictions generate an important asymmetry by a normalization ordering of the variables so a ε_{zt} shock affects both e'_{1t} and e'_{2t} , but ε_{yt} does not affect e'_{2t}. Sometimes economic theory can suggest such a restriction, or there may be *a priori* information available, but the importance of the ordering depends crucially on the magnitude of the correlation coefficient between e'_{1t} and e'_{2t} , $\boldsymbol{\rho_{12}=\sigma_{12}/\sigma_1\sigma_2}$. If $\boldsymbol{\rho_{12}}$=0, the ordering is of no consequence since there is then no correlation between the VAR equations and the residuals of the structural VAR, and the VAR standard residuals are equal $e_{1t}=\varepsilon_{yt}$ and $e_{2t}=\varepsilon_{zt}$, that is E($e_{1t}\, e_{2t}$)=0 and both b_{12} and b_{21} can be set equal to zero. At the other extreme, if $\boldsymbol{\rho_{12}}$=1, a single shock contemporaneously affects both variables. We should therefore test the significance of $\boldsymbol{\rho_{12}}$, and then obtain a particular the impulse-response function and compare the results with that function obtained from reversing the ordering. If the outcomes are different, additional investigation is necessary.

Example The four response functions with the numerical values of $\sigma_y = 1, \sigma_x = 2,$

$\partial_{11} = 0.7, \partial_{12} = 0.2, \partial_{21} = 0.3, \partial_{22} = 0.6$ are shown below.



Example: the impulse-response for *LA*, and *Riverside* (a suburb of *LA*) housing markets: 95% confidence of bonds (the bonds including zero indicate statistically insignificant response).Since a shock in Riverside does not have a contemporaneous effect on the LA market, the ordering of the shock is (GLS, GRiv)



Note that a shock to *LA* lasts 10 quarters in both *LA* and *Riverside* markets (two left graphs), while a shock in *Riverside* disappears after two quarters in *Riverside* and has no effect over time in *LA*.The *LA* market dominates the dynamics in both markets.

Variance Decomposition

An alternative method to express the impulse-response function is by means of **Variance Decomposition** that estimates "how much of the variance for the *h*-step-ahead error of the variable

y_t is explained by the innovations of the variable z_t? ” To see how the decomposition is obtained, obtain the n-step-ahead forecast starting from the same information given:

$$x_{t+n}=\mu+\sum_{i=0}^{\infty} b_i \varepsilon_{t+n-i};$$

Therefore, the forecast error of the n-period is equal to the difference at time $t+n$ of the realized value from its expected value at $t+n$:

$$x_{t+n} - Ex_{t+n} = \sum_{i=0}^{n-1} b_i \varepsilon_{t+n-i}$$

The n-step-ahead error variance for the $\{y_t\}$ sequence, for example, is thus

$$y_{t+n} - Ex_{t+n} = b_{11}(0)\varepsilon_{y_{t+n}} + b_{11}(1)\varepsilon_{y_{t+n-1}} + \ldots + b_{11}(n-1)\varepsilon_{y_{t+1}}$$

$$+b_{12}(0)\varepsilon_{z_{t+n}} + b_{12}(1)\varepsilon_{z_{t+n-1}} + \ldots + b_{12}(n-1)\varepsilon_{z_{t+1}}.$$

Then the n-step-ahead forecast error variance $\sigma(n)^2$ of y_{t+n} is

$$\sigma(n)^2=\sigma^2_y[b_{11}(0)^2 + b_{11}(1)^2 + \ldots + b_{11}(n\text{-}1)^2] + \sigma^2_z[b_{12}(0)^2 + b_{12}(1)^2 + \ldots + b_{12}(n\text{-}1)^2]$$

We can use this equation to decompose $\sigma(n)^2$ into the proportions due to each shock of the $\{y_t\}$ and $\{z_t\}$ sequences:

$$\{\sigma^2_y[b_{11}(0)^2 + b_{11}(1)^2 + \ldots + b_{11}(n\text{-}1)^2]\} \,/\, \sigma(n)^2$$

$$\{\sigma^2_z[b_{12}(0)^2 + b_{12}(1)^2 + \ldots + b_{12}(n\text{-}1)^2]\} \,/\, \sigma(n)^2$$

The decomposition then shows the error variance due to own shock and that due to the other shock. Note the important implication that since all the values inside square brackets are non-negative, *the variance of the forecast error increases as $n \to \infty$*. If the shock in one sequence can explain all the forecast error variance at all forecast horizons, then the other sequence would be entirely endogenous. In applied work, it is typical that a shock from a variable explains almost all of the error at short horizons but decreasingly smaller proportions at longer horizons. We then expect ε_{zt} shocks to have small contemporaneous impact on y_t but with increasingly bigger impacts on the $\{y_t\}$ sequence with a lag. Finally, note that the decomposition uses the same information as the reverse ordering but processes that information differently. If the identification is a minor issue because correlations among the innovations are small, then the two methods should yield similar results; in general, the ordering approach is more popular, and it is not necessary to present the results by both methods.

Readings

For textbook discussion, see Enders (2015, chapters 5 and 6), Hamilton (1994, chapters 17, 18, and 19). Phillips (1954) proposed the ECM; Engle and Granger (1987) proved the equivalence between cointegration and equilibrium time-series.

Chapter 7 Stationarity, ARDL VAR & Impulse-Response Exercises

Q7.1 Consider the following two-variable *VAR* model with one lag and no intercept, and contemporaneously uncorrelated error terms:

$$Y_t = \beta_{11}Y_{t-1} + \gamma_{11}X_{t-1} + u_{1t} \sim WN\,(0, \sigma_y^2)$$

$$X_t = \beta_{21}Y_{t-1} + \gamma_{21}X_{t-1} + u_{2t} \sim WN\,(0, \sigma_x^2)$$

a. What are the special features of this *VAR* model that permits the *OLS* equation-by-equation estimation of each equation? How would you justify the absence of the intercepts in this model?

b. Show that the two-period-ahead forecast for Y starting back at period (t-2) can be written as $Y_{t/t-2} = \delta_1 Y_{t-2} + \delta_2 X_{t-2}$, and drive values for δ_1 and δ_2 in terms of the coefficients in the *VAR*.

Q7.2 Suppose the residuals of a VAR are such that var(e_1)=0.75, var(e_2)=0.25, and cov$(e_1, e_2) = 0.25$.

a. Using (5.55)-(5.58) in Enders (2015) as guides, show that it is not possible to identify the structural VAR.

b. Using the Choleski decomposition such that b_{12}=0, find identified values of b_{21} $var(\varepsilon_1)$ & $var(\varepsilon_2)$.

c. Using the Choleski decomposition such that b_{21}=0, find identified values of b_{12} $var(\varepsilon_1)$ & $var(\varepsilon_2)$.

Q7.3 Download *lutkepohl12.dta* again to analyze the impulse-response dynamic of the 1978 OPEC oil shock to US income, consumption and investment.

a. Fit a 3-varaible *VAR* model and estimate simple and dynamic *IRF*s(exogenous unit change effect on endogenous variables over time); use Choleski ordering (dln_inv dln_inc dln_consump)and graph the shock from income (dln_inc) to consumption(dln_consump)over the subsequent 10 periods.

b. Change the Choleski ordering from shock to income to shock to (dln_inc dln_inv dln_consump), estimate a new IRF and put graphs and tables of estimates into the same file and comment on the outcome.

Chapter 8 Stationary Tests, Cointegration, Granger Causality & VEC Models

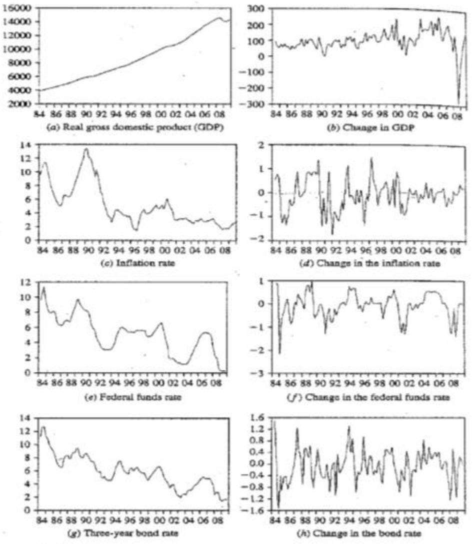
Introduction

The graphs in Figure 8.1 show four time series in terms of **level** and **change** over the previous period: GDP, inflation, federal funds rate, and bond rate. On the left panel based on levels, the GDP series trends upward, while the other three series wonder randomly either up, or down, or up and down. On the right panel graphs are obtained from one-period changes; the top two appear to show that the mean of the series **changes over time**, while the bottom two display a stable mean (around zero) over time. We call the first two series as **non-mean reversion** or **non-stationary**, and the bottom two as **mean-reversion** or **stationary** series.

Forecasting requires a time series that is stationary around its mean; non-stationary series must be transformed into stationary series before we can use them to obtain reliable forecasts.

Non-stationary time series have non-constant variance and may also have non-constant means as in figure (a). Such series do not converge; regression analysis with non-stationary series is misleading in suggesting relationships among series where, in fact, none exists. That is, non-stationary characteristics generates **spurious correlations** because the different unrelated series may have a common random trend, for xample, GDP and the gold price appearing to move together only because inflation has a common effect on both time-series**.**

Figure 8.1 *Stationary & Non-Stationary Time-Series*



(*a*) Real gross domestic product (GDP)

(*b*) Change in GDP

(*c*) Inflation rate

(*d*) Change in the inflation rate

(*e*) Federal funds rate

(*f*) Change in the federal funds rate

(*g*) Three-year bond rate

(*h*) Change in the bond rate

In this chapter we discuss testing for mean-reverting or the stationary characteristics of a time-series process, also known as the *Unit-Root Test for Integration*. Moreover, stationary tests are also critical for both testing of macroeconomic theories and designing of public policies, since we must first establish whether there are genuine co-movements between two or more time-series by a *Co-integration Test* and if so, employ models known as the *Error-Correction* Models, that can quantify such co-movements.

We need to recognize various type of non-stationary characteristics, test for the presence of these non-stationary characteristics, and learn how to convert non-stationary series into stationary series so we can apply the various models of time series analysis and forecasting we have examined so far.

8.1 *Unit Root Test of Integrated Series*

Start with *AR* (1) stationary model with the error term v_t that is independent with mean zero and variance σ^2:

$$y_t=\rho y_{t-1} + v_t \quad |\rho|< 1$$

when the y_t series is stationary, that is, tends to converge to its long-run mean because of $|\rho|< 1$.

Now contrast this special case *AR* (1) with ρ=1:

$$y_t= y_{t-1} + v_t \quad \rho=1 \qquad \text{or } (\Delta y_t= v_t) \qquad (8.1.1)$$

This is a non-stationary series where each y_t is equal to its value in the previous period y_{t-1} plus an unpredictable random shock. The series “wanders” around slowly up and down with no apparent pattern. Such a non-stationary process is known as the **Random Walks Model (*RWM*)**. Note that optimal forecast with this model is independent of the forecast horizon: any shock that moves the series up or down also moves the forecasted values up or down the same way *permanently* (not diminishing). Unlike the shocks with *AR* or *ARMA*, the shocks with this model have permanent rather than decaying effects on the forecast. Such series do **not** have the property of mean reversion. The Random Walk Model provides the basis of various tests of non-stationary characteristics.

For understanding the forecasting implication, we note that at time t, y_{t+1} is unknown random variable; the minimization of the expected squared forecast error is obtained from the probability of the conditional expectation of E(y_{t+1} | I_t) also minimizes the unconditional E(y_{t+1}), for a random variable y_t. Both are equal because both are zero. That is, we have a sequence in which the past has no useful information about predicting the future. In general, such a sequence is known as a **Martingale Differenced Series**. The series takes its name from a **Martingale Process** in which at any point in time its expected value is equal to its most recent value, E(y_{t+1} | I_t) = y_t; applying differencing of such a series to both sides of this process renders a series equal to zero . However, a Martingale Differenced Sequence is a stronger condition than a Random Walks Sequence, in that the latter states that a serially uncorrelated sequence cannot be forecast on the basis of a linear function of its past values. A Martingale Differenced Sequence generalizes that to state that no function of past values, linear or nonlinear, can forecast the sequence. The RWM is employed to explain that the behavior of the financial time-series from one day to the next is completely random, and it also occupies an important place in modern Macroeconomics with some versions of the efficient market hypothesis maintaining that the market variables time-series incorporate all past and current information. Past values have negligible ability to predict future behavior.

i. *There are three types of random walks*

1-*Random walks with stochastic trend.* By backward substitution (see the earlier notes on conversion of the *MA* and *AR* process into each other), we can re-write (8.1.1) as the starting value of the series plus the sum of all v_t from each *t* period:

$y_t = y_0 + \sum_{s=1}^{t} v_t$. Since $y_0 \approx 0$ (too far in the past), y_t is determined by $\sum_{s=1}^{t} v_t$ component or its **stochastic trend**. To see why ρ=1 violates the stationary conditions; examine its mean and variance:

$$E(y_t)=y_0+ E(v_1+ v_2+ \ldots+ v_t)= y_0$$

because v_t are independent with mean zero.

$$Var(y_t)=Var(v_1+ v_2+ \ldots+ v_t)= t\sigma_v^2 \qquad (8.1.2)$$

That is, the series has a constant mean but its variance increases as *t* becomes larger.

2- *Stochastic random walks with drift*

We can extend the stochastic trend model to account for a series that also displays “wandering” or drifting patterns by introducing an ”intercept” into the above model.

$$y_t = \alpha + y_{t-1} + v_t \qquad \rho=1 \qquad (8.1.3)$$

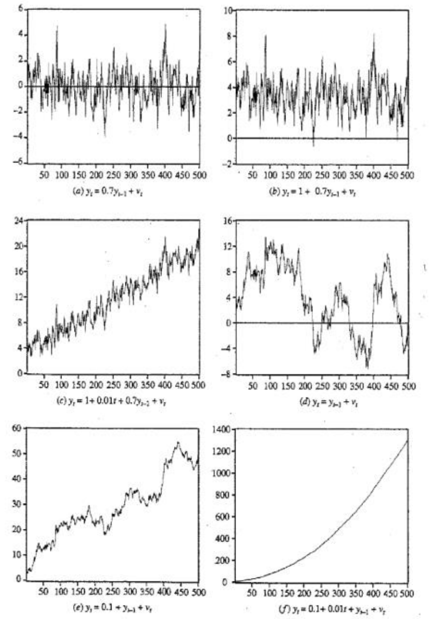
3-*stochastic random walks with drift plus a deterministic trend*

We further extend (8.1.3) by adding a deterministic linear time trend for series with a pattern like the GDP in Figure 8.1 (c)

$$y_t = \alpha + \delta t + y_{t-1} + v_t \qquad \rho=1 \qquad (8.1.4)$$

Following the same procedure as above, we can show non-stationary characteristics in this case in terms of a non-constant mean and variance of the series:

$E(y_t)= \alpha+t\delta$ and $Var(y_t)= t\sigma_v^2$. In this case, however, both the mean and variance are non-constant, both increasing with *t*.



(a) $y_t = 0.7y_{t-1} + v_t$

(b) $y_t = 1 + 0.7y_{t-1} + v_t$

(c) $y_t = 1 + 0.01t + 0.7y_{t-1} + v_t$

(d) $y_t = y_{t-1} + v_t$

(e) $y_t = 0.1 + y_{t-1} + v_t$

(f) $y_t = 0.1 + 0.01t + y_{t-1} + v_t$

The graphs above illustrate different types of stationary series: (a) is zero-mean $AR(1)$, (b) has a constant non-zero mean, (c) wanders around by a fixed amount α, (d) is a random walk series, (e) a random walk that has a drift, and (f) has a drift plus a time trend. What separates the first three graphs and the last three is $\rho<1$ *v.* $\rho=1$ (characteristic feature of random walk patterns)

*The **key idea** for converting a nonstationary into a stationary series:*

Why do series have stable mean in differenced form but not in levels? Start from the simplest case $y_t=y_{t-1}+v_t$; with independent $v_t \sim N(0, \sigma^2)$, then $\Delta y_t = y_t - y_{t-1} = v_t$. Since v_t is mean zero stationary, so is Δy_t; that is, *working with differenced series rather than one in levels converts a nonstationary series into a stationary one*.

ii. *Tests of non-stationary behavior*

There are many tests for the non-stationary behavior of a time series, but the **Dickey-Fuller**

(*DF*) Test is the most popular and the one we use in this section of the course (but see below). The DF tests are based on (8.1.1), testing if $\rho=1$. Such tests are known as **unit root tests** for stationarity; each version of the DF test corresponds to each of the three types of non-stationary random walks examined above.

1-*Test for the basic stochastic trend non-stationary series.*

It is more convenient to work with the first differenced version of (8.1.1) by subtracting y_{t-1} from both sides of (8.1.1):

$$y_t - y_{t-1} = - y_{t-1} + \rho y_{t-1} + v_t \rightarrow \Delta y_t = (\rho - 1)\, y_{t-1} + v_t \rightarrow \Delta y_t = \gamma y_{t-1} + v_t \qquad (8.1.5)$$

where $\gamma = (\rho - 1)$.

2-*Test for a non-stationary stochastic trend series with drift.*

$$\Delta y_t = \alpha + \gamma y_{t-1} + v_t \qquad (8.1.6)$$

3-*Test for a non-stationary stochastic trend series with drift plus a deterministic trend.*

$$\Delta y_t = \alpha + \delta t + \gamma y_{t-1} + v_t \qquad (8.1.7)$$

In all three versions of the unit root tests the null and alternatives are:

$$Ho\text{: } \rho = 1 \Leftrightarrow \gamma = 0 \qquad v.\ Ha\text{: } \rho < 1 \Leftrightarrow \gamma < 0$$

Since the interest is almost always in testing for $\rho < 1$, the alternative hypothesis *Ha* is set up for a one-sided DF test. Note that the null is that the series is non-stationary, so *rejection of Ho suggests the series is stationary while failure to reject it indicates it is non-stationary*. Moreover, we cannot test non-stationarity with the standard *t*-test for $\gamma < 0$ because the increasing variance of a non-stationary series with the sample size, as shown in equation 2 above, (*t)* changes the usual distribution of the *t*-statistic. The DF tables provide the correct critical *tau* (τ) values that are smaller than the corresponding critical *t* values; see the table of DF critical values below.

iii. *Augmented Unit-root Test*

Finally, another extension the *DF* test is to account for possible autocorrelation of the error term, and clear the residual of any serial correlation in order to ensure that we have a white noise error term (remember the consequences of the exclusion of relevant variables). If the model contains an insufficient number of lags, then add more first-differenced lags to remove the residual autocorrelation. For example, with two first-differenced lags, the DF test consists of

$$\Delta y_t = \alpha_0 + \gamma y_{t-1} + \alpha_1 \Delta y_{t-1} + \alpha_2 \Delta y_{t-1} + v_t$$

where $\Delta y_{t-1} = (y_{t-1} - y_{t-2})$ and $\Delta y_{t-2} = (y_{t-2} - y_{t-3})$.

This is called the **augmented *DF* (*ADF*)** test of stationary behavior; therefore, the ADF tests contain differenced lags, while the standard *ADF* employs lags in levels. The critical values for the null and alternative are the same as above, *Ho* : $\gamma = 0$, depending on the presence of a drift and a time-trend in the test equation.

Application example. First plot the series to check the type of non-stationary behavior; for example, if the series fluctuates around a non-zero mean, then (8.1.6) is appropriate, if it fluctuates around a linear trend, then apply (8.1.7). Below (8.1.6) is applied to the US federal funds rate series.

$$\Delta F_t = 0.174 - 0.045\, F_{t-1} + 0.501 \Delta F_{t-1}$$

$$(tau)\ (-\ 2.505)$$

Note that the test employs the first-differenced F_t, but *testing is on the coefficient of the lagged variable in levels, not on the coefficient of the lagged difference*, as it must according to (8.1.1) above. The critical value for the *DF* test with drift but no time trend at 5% = - 2.86 < - 2.505 (further below zero), so we *fail* to reject *Ho*: $\rho = 1 \Leftrightarrow \gamma = 0;$ we conclude the series is non-stationary , DF smaller negative (or positive) fails to reject, (see below).

iv. *ARIMA* & Integrated series

Re-write (8.1.1) as$\Delta\ y_t\ =\ (y_t\ -\ y_{t-1}) =\ v_t$. Since v_t is independent with zero mean and constant varianceσ^2, the $\Delta\, y_t$ series in first-difference is stationary, even though the original series in levels is not. This is an example of a non-stationary series made stationary by differencing. In this case, the stationary series is **integrated of order one *I(1)***, or "*eye-one*" for short. The order of integration is the minimum number of times a non-stationary series must be differenced in order to convert it into a stationary one, hence ***I(0)*** stands for a stationary series; ***I(1)*** is stationary in first-difference, etc. Such series are an extension of *ARMA* for which the series are defined in differences rather than levels and the model is called the **Autoregressive Integrated Moving Average (*ARIMA*)** Model. The *ARIMA(p, d, q)* is simply the *ARMA(p, q)* after differencing the series *d* times to make it a stationary series. In practice, we often employ *I(0)* and *I(1) ARIMA*, rarely, if ever, with *I(2) ARIMA*. Combined with low order *I(d)*, *ARIMA* captures dynamic features of a time series remarkably well.

An example may be applied to the federal funds rate to test if the first differencing of the federal funds rate ($\Delta F_t = F_t - F_{t-1}$) is stationary.

First differencing of ($\Delta F_t = F_{t-1} + v_t$) results in

$$\Delta\,(\Delta\, F_t\,) = -0.447\ \Delta\, F_{t-1}.$$

(*tau*) (- 5.487)

The DF critical value with (5) at 5% = - 01.94> -5.487), so we reject the null of non-stationary, concluding, the series is stationary, the DF statistic is a larger negative than the critica value rejects the null.

v. Power of Unit-Root Tests

How powerful are the *DF* unit-root tests? To answer this question, let us examine the variance of the *RWM* by transforming the model from *AR* into a *MA* sequence by backward substitution, using *AR*(1) for simplicity:

$$y_t = y_0 + \sum_{i=1}^{t} \varepsilon_t$$

The variance of this *MA* process is time-dependent, given the value of y_t:

$$Var(y_t) = Var(\varepsilon_t, \varepsilon_{t-1}, \ldots, \varepsilon_1) = t\sigma^2 \text{ \& } Var(y_{t-s}) = Var(\varepsilon_{t-s}, \varepsilon_{t-s-1}, \ldots, \varepsilon_1) = (t-s)\sigma^2$$

That is the variance, $Var(y_t) \neq Var(y_{t-s})$, is non-constant (heteroskedastic); moreover, as $t \rightarrow \infty$, $Var(y_t) \rightarrow \infty$, suggesting a random walk process with no tendency to increase or decrease. The autocorrelation function of the above process is not covariance stationary, a key condition for linear forecasting; and it is easily verified by forming the covariance of y_{t-s}:

$$E(y_t - y_0)(y_{t-s} - y_0) = E[(\varepsilon_t, \varepsilon_{t-1}, \ldots, \varepsilon_1)(\varepsilon_{t-s}, \varepsilon_{t-s-1}, \ldots, \varepsilon_1)] = E[(\varepsilon_{t-s})^2 + (\varepsilon_{t-s-1})^2 + \ldots + (\varepsilon_1)^2] = (t-s)\sigma^2$$

We obtain the correlation coefficient ρ_s as a result of dividing the covariance by the product of the standard deviations of y_t & y_{t-s}:

$$\rho_s = \frac{(t-s)}{\sqrt{(t-s)t}} = [\frac{(t-s)}{t}]^{0.5}$$

This result highlights the crucial weakness of the *DF* tests to detect a non-stationary series. It suggests, for the first few autocorrelations when the sample size is large relative to the number of lags, when s is small, the ratio $(t\text{-}s)/t$ is approximately equal to unity but as s increases, ρ_s will decline, but only slightly. Applied to sample data, the autocorrelation function for a random walk process, is either unity (nonstationary), or declining slowly, that is, stationary but converging very slowly. That means we cannot always use the autocorrelation function of the unit-root test; it is not good at distinguishing between a unit-root process and a stationary process when the autoregressive coefficient is close to one. This is in part, due the weak power of the DF test.

The **power** of a test is the probability of rejecting a false null hypothesis, equal to (1 – the probability of a type II error); a good power should reject the null of unit-root (non-stationary characteristic) when the series tested is, in fact, stationary. A simple Monte Carlo experiment can demonstrate this. Suppose the data-generating process for $\{y_t\}$ is $y_t=a_0 + a_1y_{t-1} + \varepsilon_t$ where $| a_1 | < 1$. The power of the D-F unit-root test depends on the value of a_1. Since the size of a_0 is not important, it is set equal to zero, and the initial value of y_0 is set equal to the unconditional mean of zero; we set a value for a_1, e.g. 0.8. Next, estimate the series in this form: $\Delta y_t=a_0 + \gamma\, y_{t-1} + \varepsilon_t$. Draw a $\{\varepsilon_t\}$ random sequence from a standard normal distribution, and repeat the experiment 10, 000 times. Then repeat this process at other values of a_1 progressively closer to one. The table below shows the proportions (out of 10, 000 simulations) the *DF* unit-root tests *reject* such stationary processes at different values of $a_{1,}$ for different levels of confidence, 10, 5 and 1 percent, and falsely conclude a time-series as non-stationary. At a_1= 0.8, the test result is quite reasonable; however, as the process remains stationary but with an autocorrelation coefficient with a value close to one. The power of the *DF* unit-root stationarity test diminishes very rapidly.

A Monte Carlo experiment for rejection of Ho: γ at different values of α_1

a_1	10%	5%	1%
0.80	95.9	87.4	51.4
0.90	52.1	33.1	9.0
0.95	23.4	12.7	2.6
0.99	10.5	5.8	1.3

It is also a common practice to lower the confidence level threshold and conclude stationary with a *DF* critical value of 10% rather than a 5% significance level; it has also been found that a generous lag structure improves the power of *DF* test, so it is a good idea to check the test results with various number of lags to see if stationarity is supported. In such circumstances, we must try detecting a unit-root process with a more powerful test such as a *KPSS* test, or a test for unit-root by pooling time-series and cross section data that improves the power, as proposed by Im, Pesaran and Shine(2003); We examine such tests later.

8.2 *Cointegration*

We learned that a non-stationary process can be made stationary by differencing. Once the general consensus was that this approach is equally valid in a multivariate context, that is linear combination of two on-stationary series was also non-stationary. However, more recent econometrics development has shown that a multivariate stationary state has a more complex structure. The tests of stationary structure with a vector of variables are designed to detect co-movement among different variables, and are known as the **Cointegration Tests**. Here we examine an extension of the DF test to the multivariate vector of time-series variables based on the regression residual obtained from correlation of the variable series, and will be explored more extensively next; in the context of the *Error Correction Model* of multivariate stationary series.

If the two non-stationary series x_t and y_t have a common stochastic trend, any linear combination of the two series, or their equation residual, would be stationary and the two series are said to be **cointegrated**. The *DF test in this case is applied to the errors rather than the series themselves*.

$$\hat{e}_t = yt - \beta_o - \beta_2\, x_t,$$

The test equation is $\Delta \hat{e}_t = \gamma \hat{e}_{t-1} + v_t$ where $\Delta \hat{e}_t = \hat{e}_t - \hat{e}_{t-1}$. Note that there is no constant if the mean of the residuals is zero even if the original series has a drift term.

We apply the *DF* test to the estimated values of the residuals since the error values are unobservable, and the *DF* tests based on the estimated residuals, then apply the *DF* critical values. Use the *DF* table critical values specifically obtained for cointegration tests depending on the type of non-stationary structures listed above. An example is the application to the relationship between the federal funds rates *F* and bond rates *B* (with drift)

$$B_t = 1.140 + 0.91\, F_t,\ R^2 = 0.881.$$
$$t \quad (6.584) \quad (29.421)$$

The unit root test applied to

$$e_{t\_hat} = B_{t\_hat} - 1.140 - 0.91\, F_t\ .$$

The residuals from this equation results in

$$\Delta \hat{e}_t = -0.225\, \hat{e}_{t-1} + 0.254\, \Delta\, \hat{e}_{t-1}$$
$$tau \qquad (-4.196)$$

Note that the test is on the coefficient of the lagged variable, not on the differenced variable, since this is just an extension of (8.1.1) above; therefore, the same logic applies.

H_0: residuals are non-stationary; the DF critical values for the cointegration test at 5%= -3.37> -4.196, reject H_0, namely, that the linear combination of the two series is stationary.

Important implication: public policy effectiveness requires establishing first that B_t and F_t have a real relationship to each other; if B_t and F_t were spuriously correlated, then monetary policy would have little impact on the economy.

i.*Two remedies for non-stationary time-series*

1-Use first-differencing if the non-stationary series are of the *difference stationary* type, for example:

$y_t = \alpha + y_{t-1} + v_t$ made stationary by $\Delta y_t = \alpha + v_t$.

2-Consider $y_t = \alpha + \delta t + v_t$. This is a *trend stationary* series since it is possible to convert it into a stationary series by "*de-trending*" it: $y_t - \alpha - \delta t = v_t$. This model is made stationary around a deterministic trend.

Cointegration addresses the question that arises if two series are genuinely rather than spuriously related to each other. The next equation is to understand which way the causal direction of the link between the two proceeds. To answer this equation, note that there are two possible ways to model how two variables y_t and x_t , say inflation and GDP, relate to each other:

$$y_t = \beta_{10} + \beta_{11} x_{t-1} + \beta_{12} y_{t-1} + e_t^y \; ; \; e_t^y \sim \mathrm{N}(0, \sigma^2)$$

$$x_t = \beta_{20} + \beta_{21} y_{t-1} + \beta_{22} x_{t-1} + e_t^x \; ; \; e_t^x \sim \mathrm{N}(0, \sigma^2)$$

Because, in this model, everything depends on everything else, β parameters are determined simultaneously. However, the normally distributed residuals of each equation makes it possible to estimate this interdependent system of equations one-by-one as though the equations stands on their own and can thus each be estimated as a single equation. This is an example of **vector autoregressive or VAR** with a two variable vector examined in notes below. In such cases, a special type of test called *Granger-Causality* is required to determine whether the direction of the effect is from *x* to *y*, or from *y* to *x*. That is why LH contemporaneous terms are excluded.

8.3 *Granger Causality Test*

In time-series regression models, we may wish to know that lags of an independent variable have predictive power in addition to the other regressors in the model. The *Granger Causality Test* provides a useful F test of the predictive power of such variables: the null hypothesis is that the coefficients on all lags of that independent variable are jointly zero.

$$y_t = \beta_0 + \sum_{i=1}^{N_x} \alpha_i y_{t-i} + \sum_{i=1}^{N_x} \delta_i x_{t-i} + \varepsilon_t \qquad \& \ H_0\text{: } \delta_1 = \delta_2 = \ldots \delta_N = 0$$

The idea is to test that, after controlling for the effects of past values of y_{t-1}, whether past values of x_t have any impact on prediction future y_t. Note carefully that the test employs only lagged variables as independent variables; it has no bearing on *contemporaneous* causality between x_t and y_t, and therefore we have no cross-sectional Ganger test since its application does not make sense to apply it to cross-sectional data.

The Granger test is a pair of F tests that run in *opposite* directions; first, we test that the coefficients on all x_{t-i} are zero with y_t as the dependent variable, and then we test that the coefficients on all y_{t-1} are zero with x_t as the dependent variable. When *x* come in time before y, with significant estimated coefficients, then we say x_{t-i} "causes" y_t, or more accurately *predicts* the value of y_t. In that case the first *F* test rejects *Ho*, but the second F test fails to reject H_0: $\alpha_1 = \alpha_2 = \ldots \alpha_N = 0$, that is *y* does **not** "cause" *x*. Granger tests can be useful particularly in the context of ***VAR*** model where everything causes everything else because every lagged variable is an independent variable in every equation. Building a simultaneous system of equations often requires simplifying the model by identifying those variables with no predictive power. Of course this is not a test of causality in the philosophical sense, for example, appearance of New Year cards do not cause the arrival of the New Year! It is merely a shorthand for saying x_{t-i} are useful for prediction of y_t. Moreover, the Granger causality test must not be mistaken for an exogeneity test; an exogeneity variable z_t is unaffected by the contemporaneous values of y_t, while the Granger causality tests for the effects of past values of y_t on the current value of z_t (indirectly via z_{t-1}). It is also possible to extend Granger causality tests beyond the bivariate *VAR* by adding *more variables to the VAR system. For example, in a three-variable system with* w_t, z_t, *and* y_t, *the test is* whether lags of w_t Granger cause either z_t or y_t. Such a test is called a **block-exogeneity** test since it tests the restrictions that all lags of w_t in the z_t, and y_t $\sum$ are equal to zero. This cross-equation restriction can be tested using the likelihood ratio test estimating the restricted and unrestricted variance-covariance matrix and obtaining the likelihood ratio test statistic given c, the maximum number of regressors in the largest equation

$$(T-c)(\ln|\textstyle\sum_r| - \ln|\textstyle\sum_u|)$$

This is a *chi*-squared test with df equal to 2p where p is the number of excluded w_t lags.

Example: We use U.S. seasonally adjusted housing starts (*ST*) and completions (*CO*) 1968-1996 (absolute *t*-ratios in brackets), two key indicators of the U.S. business cycles; 4 lagged values.

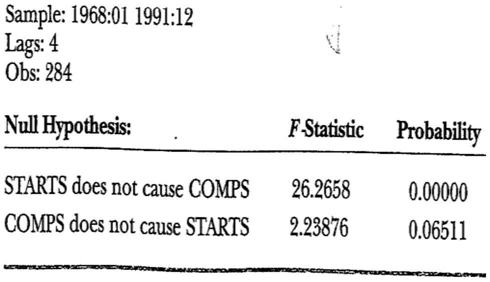
ST= 0.147 (3.32)+0.600 (10.76)ST_$_1$+0.230 (3.16)+ST_$_2$+0.143 (1.97)ST_$_3$+0.008(0.12)ST_$_4$+

0.032(0.31)CO_$_1$– 0.121(1.16)CO_$_2$– 0.021 (0.20)CO_$_3$– 0.027(0.29)CO_$_4$

CO= 0.045 (1.76)+0.075 (2.09)ST_$_1$+0.040 (0.94)+ST_$_2$+0.047 (1.11)ST_$_3$+0.082(2.13)ST_$_4$+

0.237(3.95)CO_$_1$– 0.206(3.41)CO_$_2$+ 0.121 (2.56)CO_$_3$+0/157(2.84)CO_$_4$

In this case, the *F*-statistic decisively rejects the null that there is non-causality from starts to completions, but interestingly the null for non-causality from completions to starts is also rejected, though much less decisively. So, in this example we have **feedback** effects:



Sample: 1968:01 1991:12
Lags: 4
Obs: 284

Null Hypothesis:	*F*-Statistic	Probability
STARTS does not cause COMPS	26.2658	0.00000
COMPS does not cause STARTS	2.23876	0.06511

8.4 *Vector Error-Correction*

The *ARIMA* allows for a flexible dynamic that regression-based analysis does not process, but at the expense of ignoring economic theory that identifies the long-run relationship between variables. Conventional consensus in pre-1980 econometrics was that all non-stationary series should be modeled as *VAR*. The more recent development special case of *VAR* can be modeled to contain both non-stationary and stationary variables. Extending the *ARIMA* to incorporate the lag relationship between time-series variables, generates a new model capable of showing that divergence from the long-run equilibrium sets in motion forces that change the estimates for the explained variables so as to bring the series back to equilibrium. Time-series models with both stationary and non-stationary influences are known as the **Error Correction Models** (*ECM*) since they are all based on the long-term regression residual error between the variables. It turns out the residual from such time-series is co-integrated, hence the application of the *ECM* must start with a co-integration test. However, there are different approaches to testing and modeling such multivariate co-integrated series; we examine three of them here. The first approach due to Engle and Granger (1987), focused on the regression residual and established the link with co-integration and an error correction series; it is sometimes called the *residual-based* approach. The second, developed by Johansen (1988) relies on the rank and characteristic roots of the coefficient matrix of a multivariate *VAR*; and the third and the most recent employs a non-residual *ADRL* approach. We examine each in turn, starting with the residual-based approach.

i. Engle-Granger ECM as co-integrated VAR

Consider

$$y_t=\beta_0+\beta_1x_t+\beta_2x_{t-1}+\beta_3y_{t-1}+\varepsilon_t \qquad (8.4.1)$$

where x_t and y_t are measured in logarithms; economic theory suggests x_t and y_t will grow at the same rate, that is (y_t-x_t)=constant (except for a random ε_t). Understanding how to incorporate the effects of equilibrium into equation (8.4.1) involves:

a. obtaining expressions for the coefficients in equilibrium;

b. rewriting (8.4.1) by manipulation so as to contain the equilibrium relationship within a dynamic structure that adjusts to divergences from equilibrium.

Impose the long-run equilibrium conditions on (8.4.1) to obtain a. The long-run conditions are defined by the absence of short-run lag effects, so in the long-run we have y_t=y_{t-1} and x_t=x_{t-1}; and ε_t =0. Apply these restrictions to (8.4.1) and collect terms in y_t & x_t :

$$(1-\beta_3)\, y_t = \beta_0+(\beta_1+\beta_2)\, x_t \Rightarrow \quad y_t = [\beta_0 / (1-\beta_3)] +[(\beta_1+\beta_2)/ (1-\beta_3)]\, x_t \qquad (8.4.2)$$

This suggests a one-period disequilibrium lag-away distance from the equilibrium, when$\varepsilon_{t-1} \neq 0$, is measured by

$$y_{t-1} =[\beta_0 / (1-\beta_3)]+[(\beta_1+\beta_2)/(1-\beta_3)]x_{t-1}+\varepsilon^*_{t-1} \qquad \Rightarrow \varepsilon^*_{t-1}=\mathrm{y}_{t-1} -\phi-\theta x_{t-1} \quad (8.4.3)$$

where $\beta_0/(1 - \beta_3)=\phi$ and $[(\beta_1+\beta_2)/(1 - \beta_3)]=\theta$.

With respect to *b*., rewrite (8.4.1) by subtracting $-y_{t-1}$ from both sides, and by adding and subtracting $\beta_1 x_{t-1}$ on the RHS.

$$y_t - y_{t-1} = \beta_0+\beta_1 x_t+\beta_2 x_{t-1}+\beta_3 y_{t-1}-y_{t-1} + \beta_1 x_{t-1} - \beta_1 x_{t-1}+\varepsilon_t \quad \Rightarrow$$

$$\Delta y_t = \beta_1 \Delta x_t + \{\beta_0+(\beta_1+\beta_2)x_{t-1} - (1-\beta_3)\, y_{t-1}\} + \varepsilon_t \quad \Rightarrow$$

$$\Delta y_t = \beta_1 \Delta x_t - (1-\beta_3)\, \{y_{t-1} - \beta_0/(1-\beta_3) - [(\beta_1+\beta_2)/(1-\beta_3)]x_{t-1}\} + \varepsilon_t \qquad (8.4.4)$$

Finally, by substitution from (8.4.3) into (8.4.4) we have

$$\Delta y_t=\beta_1 \Delta x_t+\{(\beta_3 - 1).(\mathrm{y}_{t-1} -\phi-\theta x_{t-1})\} + \varepsilon_t \text{ or } \Delta y_t=\beta_1 \Delta x_t+\{(\beta_3 - 1).\varepsilon^*_{t-1}\} + \varepsilon_t \qquad (8.4.5)$$

(8.4.5) is called the **(*EC*) Error-Correction Model** because it contains a term (ε^*_{t-1}), defined by the expression inside the curly brackets, that measures the one-period divergence from equilibrium. It can be shown that in equilibrium, $\beta_1+\beta_2+\beta_3=1$ implying that $\beta_3<1$. The term ε^*_{t-1} in (8.4.5) corrects for disequilibrium by the following dynamic: if the error in y_{t-1} grows positive too quickly, that is $\mathrm{y}_{t-1}> -\phi-\theta x_{t-1}$, y_t should fall and Δy_t should be negative; while if the error in y_t grows negative too quickly, that is $\mathrm{y}_{t-1}< -\phi -\theta x_{t-1}$, y_t should rise and Δy_t should be positive. Therefore, the disequilibrium gap (8.4.3) forcing Δy_t in (8.4.5) to correct the error and move the relationship between y_t and x_t closer to equilibrium.

ii. ***Error-Correction Identified as Co-integration***

Note how the *EC* model (8.4.5) mixes two very different types of variables in differences and levels in the same equation. The *EC* term shows that if the variable in levels is *I*(1), the linear combination obtained from the regression of y_t on x_t is *I*(0), that is, the difference between y_t and x_t, namely the error term of the regression, should have a constant mean. This explains why the DF test for co-integration of the relationship between two time-series is applied to their error term rather than the series, and suggests a new meaning for co-integration, namely, that variables in equilibrium must be co-integrated. The co-integrated variables move together closely in the long-run because they have a **Common Trend**. Be careful; if x_t and y_t are co-integrated, then x_{t-1} and y_{t-1} in (8.4.1) will be highly collinear, and there would be a temptation to drop one of them. This, however, would have a disastrous consequence for the *EC* model since we would remove the co-integrating relationship and all the information it contains to improve forecast estimates.

An example is the difference between the long-run and short-term interest rates; that is, subtraction of the short-term rate from the long-term, called *the term spread*, has eliminated the trend in both of the individual rates. This suggests an alternative to removing non-stationary characteristic by differencing; namely, taking the difference between variables and testing if they are co-integrated. Suppose y_t and x_t are not stationary but are integrated in first-difference. If their linear combination y_t-θx_t is integrated as $I(0)$, then y_t and x_t are co-integrated. The coefficient θ chosen to eliminate the common trend, often taken as $\theta=1$, is called the **Cointegration Coefficient**, and (y_t-θx_t) is the **Eror-Correction Term**.

iii. Vector Error Correction (VEC) Model

A major problem remains with a single-equation representation of the *EC* model. The model implicitly assumes that all of the explanatory variables are exogenous; that rules out cross-variable effects from y_t to x_t. In order to avoid a prior exclamation of exogeneity of the *RH* variables in (8.4.2), we adapt a more general simultaneous-equation for the *EC* model in terms of lagged values of all the other variables. We would then have a *vector of single-equations* in structures:

$$\Delta y_t = \beta_{10} + \beta_{11}\Delta y_{t-1} + \cdots + \beta_{1p}\Delta y_{t-p} + \gamma_{11}\Delta x_{t-1} + \cdots + \gamma_{1p}\Delta x_{t-p} + \alpha_1 \hat{\varepsilon}_{t-1} + \mu_{1t}$$

$$\Delta x_t = \beta_{20} + \beta_{21}\Delta y_{t-1} + \cdots + \beta_{2p}\Delta y_{t-p} + \gamma_{21}\Delta x_{t-1} + \cdots + \gamma_{2p}\Delta x_{t-p} + \alpha_2 \hat{\varepsilon}_{t-1} + \mu_{2t}$$

where $\hat{\varepsilon}_{t-1} = y_{t-1} - \theta x_{t-1}$; α_1 and α_2 are the ***speed of adjustment coefficients***, measuring how much of the previous disequilibrium error is corrected during one unit of time taken by moving from

(t - 1) to t. We assume $\varepsilon_{1t} \sim N(0,\sigma^2)$ and $\varepsilon_{2t} \sim N(0,\sigma^2)$; and the two may be contemporaneously correlated, namely, cov(ε_{1t}, ε_{2t})≠0. This model is known as the V**ector Error Correction (*VEC*) Model**. Note the key features of *VEC*:

1-**All** variables, both on the *RH* and *LH*, are stationary, Δx_t and Δy_t differenced as $I(1)$, and $\varepsilon_{t=}(y_t$-$x_t)$ in linear combination of levels, $I(0)$.

2-At least one adjustment coefficient must be different from 0 to ensure short-term movements toward the equilibrium: $\alpha_1 \neq 0$ and/or $\alpha_2 \neq 0$.

3-*VEC* combines the long-run and short-run dynamics in a single model, something that is not possible with the *VAR* model.

Note the contrast between the *VEC* model here and the single equation *EC* model (8.4.5). Solving a system of *VEC* equations one-by-one by *OLS* requires excluding current Δy_t & Δx_t for the same reasons discussed in the context of the *VAR* model. With a single equation *OLS* estimation of the *EC* model, variable interdependence is ruled out; Δy_t & Δx_t treated as exogenous (for more, see *Dynamin Specification*, Hendry, Pagan and Sargan (1992).

A better understanding of the error correction dynamic can be obtained by examining the simplest form of the *VEC*, consisting solely of the error-correction term where all lagged variables in differenced-levels are assumed to be insignificant, and, therefore, dropped.

Consider two non-stationary variables y_t and x_t that are integrated of order 1: y_t~I(1) and x_t~I(1), so $y_t=\beta_0+\beta_1 x_1+e_t$ in a co-integrated relationship, i.e., $\hat{e}_t$~I(0). The *VEC* in this case is

$$\Delta y_t = \alpha_{10} + \alpha_{11}(y_{t-1} - \beta_0 - \beta_1 x_{t-1}) + \mu_t^y$$

$$\Delta x_t = \alpha_{20} + \alpha_{21}(y_{t-1} - \beta_0 - \beta_1 x_{t-1}) + \mu_t^x$$

(only the dependent variables differ). This can equivalently be written as a *VEC* model

$$y_t = \alpha_{10} - \alpha_{11}\beta_0 + (\alpha_{11} + 1)y_{t-1} - \alpha_{11}\beta_1 x_{t-1} + \mu_t^y \qquad (8.4.6)$$

$$x_t = \alpha_{20} - \alpha_{21}\beta_0 + \ \alpha_{21} y_{t-1} - (\alpha_{21}\beta_1 - 1)x_{t-1} + \mu_t^x \qquad (8.4.7)$$

Therefore, both (8.4.6) and (8.4.7) equations contain the common co-integrating relationship. This requires $-1<\alpha_{11}\leq 0$ and $0 \leq \alpha_{21}\leq 1$; therefore, both $|\alpha_{11|}< 1$ and $|\alpha_{21|}< 1$, but change in opposite directions so as to keep the *VEC* equation system from exploding, remain stable while moving it back towards equilibrium. Now consider $e_{t-1}>0$, similar to a simple *EC* examined above. The negative error-correction term (α_{11}) in the first equation (8.4.3) forces Δy_t to fall while the positive one (α_{21}) ensures that Δx_t rises, thereby correcting the error; the reverse happens when $e_{t-1}<0$.

The graph explains the *VEC* dynamic to equilibrium for the co-integrated relationship $y=\alpha_{o+}\alpha x.$ At time (t-1), the system is at the point (x_{t-1}, y_{t-1}) and out of equilibrium by as much as $\varepsilon^*_{t-1=}z_{t-1}>0$. Given a gravitational pull of the co-integration/equilibrium relationship, the system moves to the point (x_t,y_t) where x_t has increased Δx_t>0 and y_t has decreased so Δy_t<0. There is still disequilibrium, but now by a smaller amount z_t>0. From one period to the next, the system partially corrects itself until it reaches equilibrium defined by the line $y=\alpha_{o+}\alpha x$ on which $\varepsilon t=z_t=0$.

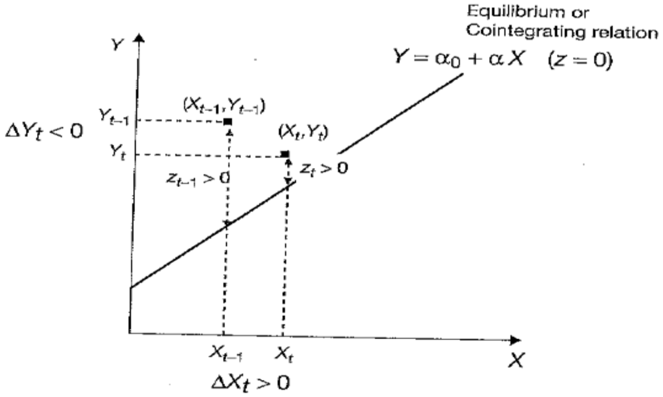


Figure 8.2 Adjustment to equilibrium by EC mechanism

The Engle and Granger original definition of the *ECM* refers to co-integrated variables that each are of *the same order of integration*, typically either all I(0) or I(1).

***iv.** Implementation*

To implement this approach, first test if the *ECM* is appropriate:

1. If all series are individually integrated and also co-integrated, then the *OLS* is the appropriate estimator; the *ECM* is not relevant. If the series each are integrated but with a different order of integration, the Granger *ECM* cannot be applied (but ECM model of AEDL is applicable).
2. If the series are not integrated and their linear combination is also not co-integrated, then the relevant model is *VAR* based on differencing, not the *ECM*.
3. If the series are not integrated but their linear combination is co-integrated, then the *ECM* is appropriate to apply.

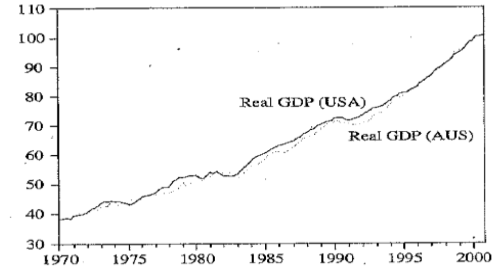
The application of *VEC* involves the following steps:

First, estimate the long-run equilibrium relationship suggested by economic theory and generate the lagged residuals $\hat{e}_{t-1}= y_{t-1}-b_0-b_1x_{t-1}$.

Second, test if the residual is co-integrated by the unit-root test.

Third, if the residuals pass the *DF* test of stationarity, include the lagged error term in a *VEC* in a system of equations expressed by (8.4.6) and (8.4.7) as the *RH* error-correction terms.

Example: The quarterly real *GDP* of a small economy (Australia, A) and a large economy (USA, U). The graph residuals of real *GDP* indicate that both terms are non-stationary but quite possibly cointegrated. The unit-root tests confirm that both series are in fact non-stationary.

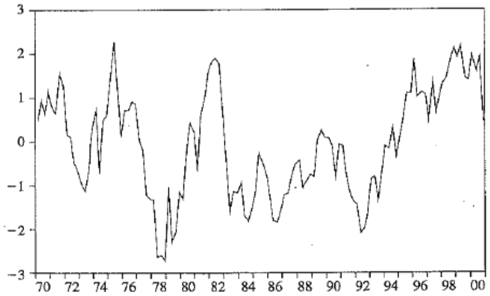


Real gross domestic products (GDP = 100 in 2000).

To check for cointegration for steps 1 and 2 alone, the long-term relationship was estimated without an intercept (it has no meaning in this example):

$$A_t=0.985U_t.$$

Note that we normalize by *A*, that is, put a coefficient of A_t equal to 1, because a large economy is more likely to affect the behavior of a small economy, not the reverse. The residuals of the relation $\hat{e}_t=A_t - 0.985U_t$ is shown in the graph below; they appear to have a zero intercept and no evident trend.



Now for step 2 where we perform a formal unit-root and obtain

$$\Delta\hat{e}_t=-0.128\ \hat{e}_{t-1}$$

tau (-2.889)

Since the unit-root's *tau*-value= -2.889<-2.76 the 5% value (remember to use the critical value for co-integration; not the single-series D-F -1.94), we reject non-stationarity. Thus, the two real GDP series are co-integrated. This suggests a unit change in the GDP of the USA results in 0.985 of a unit change in Australia. To estimate how much of that change takes place within one quarter, we implement the third stage here and estimate the error-correction *VEC* model by the *OLS*:

$$\Delta\hat{A}_t = 0.492 - 0.099\hat{e}_{t-1}$$

$$t \qquad\qquad (2.077)$$

$$\Delta\widehat{U}_t = 0.510 + 0.031\hat{e}_{t-1}$$

$$t \qquad\qquad (0.789)$$

where there is a positive co-integrating error. The negative -0.099 indicates a fall in *ΔA*, and a positive 0.031 indicates a rise in *ΔU*. Together, the two adjustment coefficients correct the disequilibrium error. However, 0.03 is insignificant, suggesting ΔU does not react to the co-integrating error, but -0.099 is significant at 5%, therefore, at least one of the two adjustment coefficients is ≠0; this outcome is consistent with the view that a large economy affects a smaller one, and not the other way around.

More general example: the relationship between the log of production and the log of consumption in the United States: the data set passed the unit root co-integration test, so estimate the disequilibrium error:

$logY_t$=0.84+.95$logC_t$+z^*_t ⇒ z^*_t = $logYt$−0.84−.95$logC_t$

The application of Schwarz model selection picks two lags of $\Delta logY_t$ and two lags of $\Delta logC_t$, therefore, we estimate a *VEC*(2) system with a disequilibrium term obtained in *i*. The table below illustrates the outcome.

Note that the adjustment coefficient is not statistically significant in the consumption equation (you can tell that this is the case, given that the size of the coefficient is nearly zero); therefore, it is mainly the movement in production that forces the system towards equilibrium. This suggests that in the unit time period (in a quarter), 11% of the disequilibrium error is corrected. Note also that the dynamics of production (lagged values) are important in consumption series and vice versa, that is, the first lags are significant in both equations.

Engle & Granger (*Econometrica*, 1987) proved that equilibrium and co-integration are equivalent conditions; that every co-integrating relationship represents an equilibrium relationship. The link between co-integration and error correction models is known as the *Granger Representation Theorem:*

> Consider two $I(1)$ series y_t and x_t, both with unit root processes (in levels). If y_t and x_t are co-integrated, then:
>
> 1-There exists a linear combination such as $z_t=y_t-\alpha_0-\alpha x_t$ that is a stationary process $I(0)$.
>
> 2-There exists an error correction representation as
>
> $$\Delta x_t = c_1 + \gamma z_{t-1} + \beta_{11}\Delta x_{t-1} + \beta_{12}\Delta x_{t-2} + \cdots + \emptyset_{11}\Delta y_{t-1} + \emptyset_{12}\Delta y_{t-2} + \cdots + \varepsilon_t$$

The question is what to do if the co-integration unit-root test indicates that long-run relationship between x_t and y_t is *not* stationary? We have already examined this case in detail: If *x* and *y* are not co=integrated, we have to estimate a vector autoregressive (*VAR*) model in first-differences.

Generalizing the above to the *n*-variable model:

$$\Delta x_t = \pi_0 + \pi x_{t-1} + \pi_1 \Delta x_{t-1} + \pi_2 \Delta x_{t-2} + \cdots + \pi_p \Delta x_{t-p} + \varepsilon_{1t} \qquad (8.4.8)$$

Where π_0 = (*n*.1) vector of intercepts with π_{i0} elements

π_i = (*n*.*n*) coefficient matrices with elements π_{jk}(i)

π = a matrix with elements γ_{jk} such that one or more of the π_{jk}≠0

ε_t= (n.1) vector with elements ε_{it}

Let all x variables be $I(1)$; the linear combination of the $I(1)$ variables is stationary. Write (8.4.8) in compact form and solve for γx_{t-1}:

$$\pi x_{t-1} = \Delta x_t - \pi_0 - \sum \pi_i \Delta x_{t-i} - \varepsilon_t$$

a. If all elements of π are zero, Δx_t does not respond to the deviation from equilibrium in the last period; therefore, there is no error-correction representation, and a usual *VAR* in first differences is applicable.
b. If one or more π_{jk}differ from zero, Δx_t does respond to the deviation from equilibrium in the last period. Therefore, x_t has an error-correction representation, and estimation of *VAR* in first differences is inappropriate.

We note that in a bivariate error-correction model, the rows of π are not linearly independent if the variables are co-integrated. This suggests we use the rank of π, that is the number of its independent rows, to determine co-integration. This is the insight of the Johansen (1988) approach we examine next.

8.5 *Johansen unit root by characteristic roots*

Going beyond the univariate, residual-based approach to co-integration testing and VEC modeling, there should be a broader alternative when examining a multivariate set of time-series for co-integration. This generalizes of the DF tests and residual-based VEC to a multivariate system of time-series variables; the fundamental core of the Johansen (1988) unit root test for co-integration is formulated in terms of the relationship between the rank of a matrix and its characteristic roots. Building on that relationship enables the Johansen procedure to provide a multivariate generalization of the Dicky-Fuller method.

i. *Johansen Unit Root Tests*

Take the univariate case of a y_t series where stationarity is tested on the magnitude if the coefficient of y_{t-1}

$$y_t=\alpha_1 y_{t-1}+\varepsilon_t$$

adding and subtracting y_{t-1} to the left-hand of this relationship leads to the standard DF test equation

$$\Delta y_t=(\alpha_1\text{-}1)y_{t-1}+\varepsilon_t$$

$(\alpha_1\text{-}1)=0$ for a $\{y_t\}$ unit root process, and $(\alpha_1\text{-}1)\neq 0$ for a stationary $\{y_t\}$. Generalize this to an n-element vector of x variables yields

$$x_t=A_1 x_{t-1}+\varepsilon_t$$

where A_1 is an ($n.n$) matrix of parameters, and x_t & ε_t are (n.1) vectors. The DF version of this multivariate equation would be

$$\Delta x_t=A_1 x_{t-1}\ \text{-}x_{t-1}\ +\varepsilon_t=\ (A_1-I)x_{t-1}+\varepsilon t=\pi x_{t-1}+\varepsilon_t$$

with $\pi=(A_1-I)$ & I is an $(n.n)$ identity matrix. The rank of $\pi=(A_1-I)$ is equal to the number of its distinct cointegrated vectors; hence if $\pi=(A_1-I)=0$ all the $\{x_{it}\}$ processes are unit root and thus not cointegrated, while if $\pi=(A_1 - I)=$ n, then all the variables are stationary if we exclude characteristic roots greater than 1 to ensure a convergent system of difference equations.

The basic Johansen procedure can be generalized to include a drift term for the possibility of a linear time-trend in the data-generating process and allow for higher order autoregressive terms for an augmented unit root test. For a drift modification, let

$$\Delta x_t=A_0+\pi x_{t-1}+\varepsilon_t$$

Where A_0 is a ($n.1$) vector of constants $(\alpha_{10}, \alpha_{20}, \ldots \alpha_{n0})^{/}$. The drift term should be included if the plot of the series suggests a clear pattern of increase or decrease over time; in this case, the rank of π is equal to the number of "de-trended" long-run relationship among the system of variables. The augmented Johansen test is written as

$$\Delta x_t=\pi x_{t-1}+\sum_{i=1}^{p-1} \pi_i \Delta x_{t-1}+\varepsilon_t$$

where $\pi= - (1 - \sum_{i=1}^{p} A_i)$ and $\pi_i= -\sum_{j=i+1}^{p} A_j$. Once again, the rank of π determines the number of independent cointegrated vectors; if rank$(\pi)=0$, the above is the *VAR* model in first differences, while if rank$(\pi)=n$, the above is the ECM model with all vectors cointegrated. An intermediate case is presented by rank$(\pi)=1$ when there is just a single cointegrated vector with πx_{t-1} as the EC term. However, in general when 1< rank$(\pi)< n$, there will be multiple and distinct cointegrated vectors and Johansen procedure provides two different methods of checking the significance of the characteristic roots for this general case. The matrix π has n characteristic roots ordered as $\lambda_1> \lambda_2>\ldots > \lambda_n$; if the rank is one, there will be only one cointegrated vector and the rest $I(1)$ processes, hence $0< \lambda_1<1$. Then with no cointegration, $\lambda_n=0$ & $ln(1)=0$, we can obtain the difference between the first and other vectors as $ln(1- \lambda_1)<0$ when $0< \lambda_1<1$ and $ln(1- \lambda_2)=$

$\ln(1- \lambda_3)=\ldots = \ln(1- \lambda_n)=0$ when $\lambda_I >1$ (all non-stationary).

After obtaining the estimates of π and solve for its characteristic roots (see appendix on how to solve for the roots (eigenvalues) of a characteristic equation), the Johansen procedure tests for e number of characteristic roots λ_i insignificantly different from unity by computing two different test-statistics.

$$\lambda_{trace}(r)= - T\sum_{i=r+1}^{n} ln\,(1 - \hat{\lambda}_i)$$

$$\lambda_{max}(r, r+1) = -T\,ln\,(1 - \hat{\lambda}_{r+1})$$

where ^ indicates the estimated values of characteristic roots, T is the number of observations, and r is the rank of π. The procedure tests the null that the number of distinct cointegrated vectors are greater or equal to the rank of the matrix against a general alternative. (r)=0 if all λi=0; the further from zero, the more negative $ln\,(1 - \hat{\lambda}_i)$, and the larger the λ_{trace} statistic. λ_{max} test for number of cointegrated vectors=r against the alternative of ($r+1$); as before, if the estimated characteristic root is close to zero, λ_{max} will be small. The critical values of λ_{trance} and λ_{max} have non-standard values that depend on a) the number of non-stationary components (n-r) under the null hypothesis of stationarity, b) on the form of A_0: no constant or drift, with a drift for a time-trend, and with a constant intercept.

Example: demand for money from Johansen and Juselius (1990) as a linear function of (log) real income, and real money supply, bond rate and interest rate on saving (T=53), 4 by 4 matrix of eigenvalues with 4 variables. Take second column for λ_{max}; for λ_4 we have 2.35= -53*ln (1-0.0434); for λ_3=6.34=-53*ln(1-0.1128); however, the third column shows λ_{trace} statistics as simple arithmetic sums, hence λ_2 for example, we have 19.5=2.35+6.6.34+10.36, etc.

	λ_{max}	λ_{trance}
λ_1=0.4332	30.09	49.14
λ_2=0.1776	10.36	19.05
λ_3=0.1128	6.34	8.69
λ_4=0.0434	2.35	2.35
Sum total	49.14	

First with λ_{trace} statistic, we test Ho: r=0 against the most general H_A: r=1, 2, 3, or 4 (here n=4) by comparing the sum of the four eigenvalues =49.14 (tests no cointegrating vector against all cointegrated) with the Johansen characteristic root critical values for (n - r) =0 (with a constant included) and for significance levels 10%=49.65, 5%=53.12 and 1%=60.15. The restriction is not binding; the variables are **not** cointegrated. To take another example, Ho: $r \leq 0$ against a less general H_A: r=2, 3, or 4 with (n - r) =3, with λ_{max} statistic summed over 2 to 4 is equal to 19.05, the corresponding critical values are 10%=32.00, 5%=34.91 and 1%=41.07. Hence, the test shows that the restriction =0, or r=1 is not binding; once again no cointegrated series in this study.

Second with λ_{max} in contrast, we test for an specific H_A, namely Ho: r=0 against H_A: r=1; the computed value for $\lambda_{max}(0, 1)$=-53*ln(1-0.4332)=30.09. The critical values for (n-r) =4 are 25.56 (10%), 28.14(5%) and 33.24(1%). Thus at 5% or even 10% we can reject the null and conclude that there is one cointegrated vector, i.e. r=1. However, check why Ho: r=1 against H_A: r=2 cannot be rejected for practice.

This example shows that the λ_{max} & λ_{trace} can lead to conflicting test conclusion, but since λ_{max} test has a more specific and focused H_A, it can identify the exact number of cointegrated vectors and thus is regarded as more reliable.

ii. Modelling Trend and intercept by Johansen Procedure

Once the Johansen cointegration is confirmed, further tests for parameters can be carried out. An issue that can be dealt with by reformulating the Johansen model employed is the modelling of the drift and intercept terms. The inclusion of various drift terms α_{i0}, when the variable displays a clear tendency to increase or decrease, permits the presence of a linear time trend in the data-generation process. In this case, we have the "detrended" cointegrated vectors, with the long-run πx_{t-1}=0, hence each {Δx_{it}} sequence has an expected value of α_{i0} and aggregation of all such changes over t give the deterministic value $\alpha_{i0t.}$ However, if the inclusion of a constant intercept is warranted, then it would hard to identify the intercept from the trend effects separately. One solution is to manipulate the elements of A_0 so as to include a constant without affecting the deterministic time trend of the system of equations. For instance, if rank (π)=1, the rows of each sequence can differ only by a scalar, thus for each sequence, α_{i0} can be restricted so as to have α_{i0}= $S_i\alpha_{i0}$ for all {Δx_{it}}. This in effect purges the linear trend from the system in favor of a general solution for all {Δx_{it}}. For example, for two data generating process with α_{10} and α_{20} trend, if we restrict α_{10}= - α_{20}, as α_{10}=1 & α_{20}= -1 for S_i= -1, then the drift trend will be removed but the deterministic time trend. If, however, the plot suggests a including a drift term and economic theory supports a cointegration vector with an intercept, then the intercept of the relationship is not identified and an identification method is necessary. A commonly employed method is to identify the portion belonging to the cointegrating vector as the amount necessary to produce an EC term with a sample mean zero. Otherwise, most studies include a drift term if data displays one, or either include a drift or a deterministic trend, but not both. If both should be included in the cointegrating vectors, then we must test to see if the drift we can be suitably restricted.

iii. Hypothesis Testing with Johansen Procedure.

An advantage of the Johansen method over the DF is that it permits restricted forms of the cointegrated vectors. The key about the Johansen strategy of testing parameter restrictions is that *with r cointegrating vectors, there can only be r stationary linear combinations.* That implies that if the restrictions are not binding, the number of cointegrating vectors must not decrease; that is, the difference between the restricted and unrestricted models should be small. For instance, in a system of four potentially cointegrated equations with two cointegrated vectors with just two cointegrated vectors, we test for cointegration by impose cointegration restriction on all four vectors and then again on just on two of the four. The difference between the two eigenvalue vectors will be insignificant if the restrictions are not be binding on the remaining vectors (these are not stationary); if the difference is significant, then the restrictions is binding and there are more than two cointegrated relationships.

First, suppose we want to test for the presence of an intercept as opposed to an unrestricted drift A_0. Then, estimate the model in two forms, and order the characteristic roots, unrestricted and restricted (with a constant) π: $\hat{\lambda}_{1,}\hat{\lambda}_{2,\dots,}\hat{\lambda}_n$ & $\hat{\lambda}_{1*,}\hat{\lambda}_{2*,\dots,}\hat{\lambda}_{n*}$. Assuming the none restricted model has r nonzero roots,

$$- T\sum_{i=r+1}^{n}[ln\ (1-\ \hat{\lambda}_{i*})\ \text{-}\ ln\ (1-\ \hat{\lambda}_{i})]$$

This test statistic has an asymptotic χ^2 distribution with (n-r) restrictions. The idea behind the test is that if the restriction is not binding, the above difference should be small, hence acceptable to include a constant; otherwise rejection would imply the presence of a linear time-trend.

Second, other restrictions, the Johansen defines two matrices α for speed of adjustment parameters and β for cointegrating parameters, both with ($n.r$) dimensions such that $\pi= \alpha\ \beta^{/}$

The presence of cross-equation restrictions makes the estimation non-linear in parameters and unsuitable for OLS application, however the maximum-likelihood method can provide estimation of $\beta^{/}$ to allow selecting α so as to make π equal to $\alpha\ \beta^{/}$. One way to restrict α is to allow rows of π to differ only by a scalar. This is easy to see in application of a single cointegrating vector that makes the rows π *all linear combinations of each other:*

$$\Delta x_{11}= \pi_{11}x_{1t-1}+\ \pi_{12}x_{2t-1}+\ldots+\ \pi_{1n}x_{nt-1}+\ldots+\varepsilon_{1t}$$

$$\Delta x_{21}= s_2(\pi_{11}x_{1t-1}+ \pi_{12}x_{2t-1}+\ldots+ \pi_{1n}x_{nt-1})+\ldots+\varepsilon_{2t}$$

$$\vdots$$

$$\Delta x_{11}= s_n(\pi_{11}x_{1t-1}+ \pi_{12}x_{2t-1}+\ldots+ \pi_{1n}x_{nt-1})+\ldots+\varepsilon_{it}$$

where s_i are scalars, and the matrices are left out notational for simplicity. Now, defining $\alpha_i= s_i.\pi_{11}$ and $\beta_i= \pi_{1i}/ \pi_{11}$ allow each equation to be rewritten so as to have $\pi= \Delta x_t=\pi x_{t-1}++\varepsilon_t$:

$$\Delta x_{it}= \alpha_i\ (x_{1t-1}+ \beta_2 x_{2t-1}+\ldots+ \beta_n x_{nt-1})+\ldots+\varepsilon_{it}$$

or in compact form as identical to the model above a

$$\Delta x_t= \alpha\ \beta^{/}x_{t-1}+\sum_{i=1}^{p-1}\pi_i\Delta x_{t-1}+\varepsilon_t$$

Where vector of cointegrating parameters $\beta=(1,\ \beta_2,\ \beta_3,\ \ldots,\ \beta_n)^{/}$ and speed adjustment parameters is presented by $\alpha=(\alpha_1,\ \alpha_2,\ \ldots,\ \alpha_n)^{/}$ identified as the coefficients of x_{1t-1} in each vector.

Third, with α *and* $\beta^{/}$ estimated, we can then test restriction on each vector separately. To test restriction on β, we solve for the restricted and unrestricted characteristic roots and compute the log difference statistic as above; the restriction is binding if the statistic is greater than the critical χ^2value.

Example of testing $\beta^{/}$ restrictions: Johnsen and Juselius (1990) impose the restriction that money and income move proportionally. They normalize on the coefficient of income (set equal to unity) and, given the unrestricted model has r=1, obtain unrestricted $-T\ln\ (1-\ \hat{\lambda}_i)$=30.09 as above; while restricted $\hat{\lambda}_{i*}$==0.433 and $+T\ln\ (1-\ \hat{\lambda}_{i*})$= ln(1-0.433)*53= - 30.04. Since there is one restriction imposed on β, and given df=1, the difference between the two equal -30.04 - (-30.09)=0.05 < $\chi^2_{\text{df=1}}$ critical values, so the restriction is not binding.

Finally, example of testing restrictions α: suppose we test for the hypothesis that only money demand, not consumption, responds long-run deviations from equilibrium, so we test for $\alpha_2=\alpha_3=\alpha_4=0$ restriction. We rely on the same testing method. The largest estimated characteristic root in the restricted model is -23.42 and the difference with the unrestricted model equals -23.42-(-30.09)=7.67 < $\chi^2_{\text{df=1}}$ critical values, the restriction still not binding.

We can follow this testing strategy further with another issue: to test for the presence of more than one cointegrated relationship among the variables, for example, when to test for the presence of two cointegrated relationship among four variables. In theory we can test each pair of variables for cointegration by restricting the individual cointegrating vectors. In practice, the interpretation of the outcome would not be easy; it would be hard to reconcile the simultaneous cointegrated with separate equilibria processes, or when the separate cointegrated unit root test contradict each other. Another problem with estimation and testing in a multivariant system is the greater potential of misspecification error in one cointegrating vector to affect the other vectors. This suggest for a single cointegrating relationship, the Engle-Granger residual-based and Johansen unit root tests have asymptotically the same distribution; however, the former is superior since it is much more robust to misspecification and to processes with non-integer unit roots.

Readings

For textbook discussion, see Enders (1915, chapters 5 and 6), Hamilton (1994, chapters 17, 18, and 19). Phillips (1954) proposed the *ECM*; Engle and Granger (1987) proved the equivalence between cointegration and equilibrium. Granger (1969) proposed the causality test; Johansen (1988) developed the rank-based cointegration tests.

Chapter 8 Stationarity Tests, Cointegration, Granger Causality & *VEC* Exercises

Q8.1 Suppose you estimate π to be:

$$\pi = \begin{bmatrix} 0.6 & -0.5 & 0.2 \\ 0.3 & -0.25 & 0.1 \\ 1.2 & -1.0 & 0.4 \end{bmatrix}$$

a. Show that the determinant of π is zero.

b. Show that two of the characteristic roots are zero, and that the third is 0.75.

c. Let β=(3 -2.5 1) be the single cointegrating vector normalized with respect to x3. Find the (3 x 1) vector a such that π=αβ.

d. Show that the three characteristic roots are: (0.0, 0.5, and 0.9)

Q8.2 Suppose that x_{1t}& $x_{2t}$are integrated of order 1 & 2, respectively. The answers to the following questions provide a sketch the proof that any linear combination of x_{1t}& $x_{2t}$is integrated of order 2.

a. Allow x_{1t}& $x_{2t}$to be the random walk processes: $x_{1t} = x_{1t-1} + \varepsilon_{1t}$, & $x_{2t} = x_{2t-1} + \varepsilon_{2t}$

i. Given the initial conditions x_{10} & x_{20}, show that the solution for x_{10} & x_{20} have the form $x_{1t} = x_{10} + \varepsilon_{1t-1}$, & $x_{2t} = x_{20} + \varepsilon_{2t-1}$.

ii. Show that the linear combination $\beta_1 x_{10} + \beta_2 x_{20}$ will generally contain a stochastic trend.

iii. What assumption is necessary to ensure that x_{1t} & $x_{2t}$are *CI* (1, 1)?

b. Now let x_{2t}be integrated of order 2. Specifically, let $\Delta x_{2t} = \Delta x_{2t-1} + \varepsilon_{2t}$. Given initial conditions for x_{20} & x_{21}, find the solution for x_{2t}. [you may allow ε_{1t}& $\varepsilon_{2t}$to be perfectly correlated]

i. Is there any linear combination of x_{1t} & x_{2t}that contains only a stochastic trend?

ii. Is there a linear combination of x_{1t} & x_{2t}that only contains a stochastic trend? Is there any linear combination ofx_{1t} & x_{2t}that does not contain a stochastic trend?

c. Provide an intuitive explanation for the statement: if x_{1t}& $x_{2t}$are integrated of order d_1& d_2 where $d_2 > d_1$, any linear combination of x_{1t}& $x_{2t}$is integrated of order d_2.

Q8.3 Download *usa.dta* and use the interest rate series for **f** (federal fund rate) and **b**(bonds rate).

a. Test **f** and **b** by DF procedure ffor unit roots in levels and in 1st differences.

b. Test for unit root by *kpss* procedure in levels and in 1st differences.

Q 8.4 Download *lutkepohl2.dta,* Quarterly SA West German macro data

a. Test ln-ivn, ln_inc and ln_consump series for unit root by DF procedure inclusive of the correct number of lags.

b. Fit a system of 3-equation, first-differenced VAR model with 2 lags from 1978q4 onward, request *sic* and *aic* values (lutstats), correction for small sample df.(dfk).

c. Include and treat ln_inv variables as exogenous.

d. Fit the model in c. with additional constraints of 2nd lags of dln_inc & dln_consump excluded.

Q8.5 Download *usa.dta* of inflation & *GDP* time-series.

a. Fit a 2-equation *VAR* model of inflation and *GDP* with 1-4 lags, and each model for Granger causality.

b. Test the output for *Granger* causality with models of 1-4 lags.

c. Now fit the same 1-4 lags *VAR* system in first differences and test each model for Granger causality.

Q8.6 Download gdp_US_AS.dta, the data set *gdp_US_AS.dta* contains the gdp time series of USA (*usa*) and Australia (*aus*). Implement the following steps to test for cointegration and estimate an Engle-Granger VEC model

a. Obtain the graph of the two time series variables to check for any pattern of co-movements between them. Test for cointegration between the two after selecting the correct number of lags. Why you might or not include a drift term in your Dicky-Fuller statistic and comment on the test outcome?

b. Select the correct number of lags for a system of two interdependent equations Δ*usa* and Δ*aus*, and estimate a VEC model. What are the error correction adjustment coefficients?

c. Interpret the coefficient estimates; which series makes the adjustment toward equilibrium and at what speed?

Q8.7 Download *txhprice.dta* for housing prices in 4 major cities of Texas. Implement the following steps to test for cointegration by Johansen method, then estimate VEC models.

a. Regress VEC model for Dallas house prices on Houston house prices as a two-equation model with correct number of lags, examine its residual time-series for cointegration; provide an interpretation of Johansen rank-order cointegration test.

b. Now fit a 4-equation model test for the number of cointegrated series by the Johansen procedure, using 3-lag equations.

c. Estimate VEC models for step a. and b. equations.

Chapter 9 ARDL, Panel Unit-root, ECM & ARFIMA

This short chapter provides a more complete discussion of *VEC* estimation and unit-root testing of integrated series not dealt with so far. We first examine an alternative approach to the estimation of co-integrated series incorporating information on the exogenous status of some parameters of the *VEC* system of equations. We, secondly, address a solution involving a more powerful integration test by exploiting additional information available from cross-section data. We can test for co-integration and estimate a *VEC* model using either the residual-based or rank-based approaches if the variables are jointly determined, and their interdependence makes it hard to identify the dependent and the independent variables. However, in other circumstances, that distinction may be clear from making some of the variables exogenous; there are potential benefits in incorporating such information into a *VEC* model. The *ADRL* model of *VEC* discussed in section 9.1, is the appropriate approach in such cases. Moreover, sometimes we can combine cross-section data with time-series to generate a panel series, which is a time-series of repeated observations on a cross-section of countries or states. The extraction of the cross-sectional mean of the panel from the individual time-series can offer a more powerful test of integration; it is discussed in section. 9.2.

9.1 *ARDL VEC*

Let us specify what exogeneity is with a simplest bivariate *VEC*(1, 1) model with no lag short-term structure in reduced form rather than structural form:

$$\Delta y_t = a_1(y_{t-1} - \beta z_{t-1}) + e_{1t} \quad (9.1.1)$$

$$\Delta z_t = a_2(y_{t-1} - \beta z_{t-1}) + e_{2t} \quad (9.1.2)$$

We write the relationship between the structural shocks and reduced form error terms (see section 7.2 VAR regressin model) as

$$\begin{bmatrix} e_{1t} \\ e_{2t} \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix} \begin{bmatrix} \varepsilon_{yt} \\ \varepsilon_{zt} \end{bmatrix}$$

In the structural VAR, the shocks are uncorrelated; therefore, $E\, \varepsilon_{yt}\, \varepsilon_{zt}=0$, however, $E\, e_{1t}\, e_{2t} \neq 0$ and in general is correlated if c_{12} and/or c_{21} differ from zero. Let the two reduced form error terns be related as

$$e_{1t}=\rho e_{2t}+v_t \quad ,(9.1.3)$$

v_t is the innovation in e_{1t}; therefore, e_{2t} and WN v_t are uncorrelated. This is, in fact, a Choleski decomposition employed with respect to the two error terms in (9.1.3); Δz_t does not respond to innovations in Δy_t, but Δy_t responds to innovations in Δz_t . Substituting (9.1.3) and (9.1.2) into (9.1.1) results in

$$\Delta y_t = a(y_{t-1} - \beta z_{t-1}) + \rho \Delta z_t + v_t \qquad (9.1.4)$$

where v_t is the difference between the two sides of (9.1.3), e_{1t} a function of z_t by (9.1.1), and $a=a_1 - \rho a_2$.. (9.1.4) cannot be estimated by the OLS because of the simultaneity problem caused by the correlation of Δz_t with v_t. That aside, the OLS estimation of a cannot separately identify a_1 and a_2. However, we can specify the conditions under which the simultaneity and identification problems are removed and the OLS estimates are efficient; namely, if $a_2= 0$ (z_t does not respond to disequilibrium discrepancy), and $c_{21}=0$ (z_t does not respond to ε_{yt}). Taken together, the two conditions make z_t weakly exogenous and causally prior to ε_{yt}. A variable that does not respond to deviations from equilibrium is called **weakly exogenous**, that is, its speed adjustment coefficient in the VEC model is zero. We can then write the VEC in an alternative

$$\Delta y_t = \beta_1 y_{t-1} + \beta_2 z_{t-1} + \beta_3 \Delta z_t + v_t \qquad (9.1.5)$$

The coefficients of (9.1.5) are unrestricted and the error-correction model in this unrestricted form is called ARDL to distinguish it from the ECM form. Although (9.1.1) and (9.1.5) are equivalent representations, (9.1.5) contains both Δz_t and v_t Estimates by (9.1.5) have smaller variance and more precise parameter estimates. A second benefit of ARDL is that short-term dynamics are not constrained by the long-term equilibrium dynamics since y_{t-1} and z_{t-1} are unrestricted; unlike the Engle-Granger and Johansen approaches that force Δy_t to be a constant proportion of last period disequilibrium.

9.2 *Integration Test with Heterogenous Units*

The standard tests of co-integration such as the Dicky-Fuller based on the null of stationarity have weak power and are often unable to distinguish between co-integrated and non-co-integrated series, with downwardly biased test statistics. There are several ways to improve the low power of AD and Johnsen unit roots tests; by relying on information provided by panel data series of *pooled* time-series and cross-sectional information, by employing a test with the *stationarity as the null* hypothesis, as with the *KPSS* test, and yet another solution is to apply unit root tests to models based on slow convergence, *fractional differencing,* examined in chapter 14 when discussing spectral analysis. Here we focus on panel series; for *KPSS*, see exercise Q8.3.

Most Macroeconomic time-series are affected by one or more common unobserved shocks. Suppose we can apply the unit root test to $1, 2, \dots n$ series, each observed over $1, 2, \dots T$ time periods y_{it} with an ADF test of the form written as

$$\Delta y_{it}=\alpha_{i0}+\Upsilon_i y_{it-1}+\alpha_{i0}t+\sum\nolimits^{i}_{j=1}\beta_{ij}\Delta y_{it-j}+\varepsilon_{it} \;;\; \mathrm{i}=1, 2, \dots n \qquad (9.2.1)$$

where we have included a time-trend variable, though if added, it should be included in all a single equation. Moreover, since different series may have a different lag structure, testing for lag length should be done separately for each series. Testing for panel unit root assumes contemporaneously uncorrelated error terms; if $\mathrm{E}(\varepsilon_{it}\ \varepsilon_{jt})\neq 0$, that effect can be contained by estimating (9.1.1) in mean-deviations forms with $\Delta\tilde{y}_{it}$ and $\Delta\tilde{y}_{it-1}$, and then we can apply the test for unit root to (9.1.1) in mean-deviations forms.

However, the application *ADF* tests to a panel series faces an asymmetric problem that does not exist in non-panel contexts. Suppose we wish to conduct a panel unit root test. Most tests are based on the null hypothesis of having a unit root. Rejection would imply a stationary series, that is all *n* time-series are independent random walks.

$$\mathrm{H_o}\text{:}\ \ \Upsilon_1=\Upsilon_2=\dots=\Upsilon_n=0$$

The formulation of the alternative hypothesis, however, depends on assumptions made about the homogeneity/heterogeneity of the panel. With the assumption that the autoregressive parameter Υ_i is identical for all cross-sectional units, that is cross-sectionally pooled, the alternative hypothesis would be

$$H^a{}_1\text{: } \Upsilon_1=\Upsilon_2=\dots=\Upsilon_n=\ \Upsilon_1=\Upsilon_2=\dots=\Upsilon\ \&\ \Upsilon>0$$

The problem with a formulation based on $H^a{}_1$ is the likelihood of frequent rejection even if a few of the *i* series are stationary, rendering the test results unconvincing. There are time-series for which the homogeneity is an inappropriate assumption; for instance, the *PPP* hypothesis provides no support for homogeneity of $\Upsilon_i=\Upsilon$. The other extreme alternative hypothesis would be to assume that at least one of the *n* series is stationary.

$$H^b{}_1\text{: } \Upsilon_i<0 \text{ for at least one } i.$$

With large *n* and *T*, the loss in degree of freedom with $H^b{}_1$ would be costly; the panel unit root tests would lack power unless *n* is relatively small. More recent developments consider alternatives that are between $H^a{}_1$ & $H^b{}_1$ with a more appropriate heterogeneity null as

$$H^c{}_1\text{: } \Upsilon_i < 0, \text{ i=1, 2, } \ldots n_{1;\, i=}\ n_1+1,\ n_1+2, \ldots,\ n \text{ such that } \lim n\rightarrow\infty\ \frac{n}{n_1}=\delta,\ 0< \delta\leq 1.$$

This formulation leads to

$$H_o\text{: } \Upsilon_i = 0 \ \&\ H^c{}_1\text{: } \delta > 0$$

In this case, reject would be evidence in favor of rejecting the unit root hypothesis for a non-zero fraction of panel's *n* members as $n\rightarrow\infty$.

To sum up, the heterogeneity of panel series introduces a new asymmetry into the formulation of the null *v.* alternative hypothesis of unit root tests, that is the specification of the null hypothesis is designed to be the same for all *i* series but the alternative hypothesis is allowed to change by ***i***; the problem is often concealed by assuming cross-sectional homogeneity, but the neglect of parameter heterogeneity can lead to spurious results in dynamic panels, namely, with lagged dependent variables as explanatory variables. In short, there is no justification to pool country-specific panels if *T* is large (over 100 observations).

The panel data asymptotic theory suggests that the mean of *n* independent and unbiased estimates of a coefficient will also be unbiased, and, by the central limit theory, the sample mean will have a normal distribution around the true mean if those estimates are independent. Testing for unit roots with panel series must be based on critical values for standard error distributions across both *i and T* by forming the sample mean of the *t*-statistic as

$$\bar{t} = 1/n \sum_{i=1}^{n} t_i$$

It is then possible to construct an asymptotically normal variance from

$$Z_{\bar{t}} = \frac{\sqrt{n}\ [\bar{t} - E(\bar{t}\,)]}{\sqrt{(\ \bar{t})}}$$

Im, Pesaran and Shin (2003) demonstrate that $Z_{\bar{t}}$ has an asymptotically standard normal distribution and obtain the theoretical mean and variance of $\bar{t}$ to determine robust *cross-sectional*

ADF, or CADF critical values corrected for the OLS bias. As n and T increase in size, t_{rob}= $\widehat{\Upsilon_\iota}$/ $\sqrt{\widehat{Var}\ (\hat{\gamma}_{i)}}$ is asymptotically standard normally distributed and therefore, consistently estimating the OLS variance of $\widehat{\Upsilon_\iota}$. The IPS critical value of the CADF tests depend on n and T , reported separately with and without the inclusion of a time-trend in the unit root equation. For example, suppose we have GDP series with n=7, T=50 and a time-trend included, then the 5% IPS critical value corrected for the OLS bias is -2.06 compared to an ADF value of -3.50. Neglecting the cross-section changes in panel data can easily lead to misleading test results.

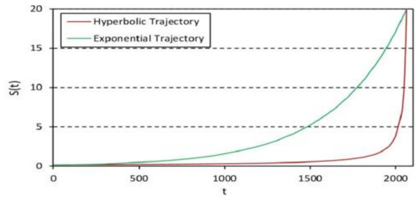
Example: an eight-country panel series from 1980Q1-2013Q1 (no time-trend) suggested $\widehat{\Upsilon_\iota} = -0.049\ (-1.678)$ for Australia based on the inclusion of five lags of Δy_{it} and the average eight-country t-statistic is -2.44, and each series has T=133 observations. Critical values at 5% and 1% n=7 and T>70 are -2.15 and -2.40; therefore, we reject the null that all $\widehat{\Upsilon_\iota}$ are zero. Nonetheless, the residuals between country-specific residuals may not be insignificant. The error correlation between Germany and France is 0.67. The practice is to obtain the mean value of each series by t$\bar{y}_t$ = $1/n \sum_{i=1}^{n} y_{it}$, then subtract this common mean from each observation y^*_{it}= (y_{it} - $\bar{y}_t$), then apply the CADF to y^*_{it}. With this correction, $\widehat{\Upsilon_\iota} = -0.043\ (-1.434)$, and although T and lag numbers remains unchanged, the mean eight-country t-statistic is now -2.50, though the null of stationarity is still rejected. Note that with $H^b{}_1$ as the alternative hypothesis, some of the series would be non-stationary; subtracting a non-stationary $\bar{y}_t$ from a stationary series introduces distortion into this method and has generated critical values of $\bar{t}$ by bootstrapping techniques.

9.3 *ARFIMA*

9.3.1 *Introduction*

Chapter 8 examined how non-stationarity series can be made stationary by differencing the series d times to obtain stationarity; where d is the integer order of differencing that renders the series $(1 - L)^d\, y_t$ stationary. We also noted that the Dicky-Fuller/Johansen unit-root weak power test, in distinguishing between a non-stationary $I(1)$ series and slowly converging stationary $I(0)$ series, result in false rejection of stationary too often. Since the standard unit-root tests have $I(1)$ as the null hypothesis and $I(0)$ as the alternative, it is also a good idea to check for false rejection of stationarity by the KPSS unit-root test that has $I(0)$ as the null and $I(1)$ as alternative. Then, a stationary series has d=0 differencing while the test result support non-stationarity, the solution is to adopt *d*=1 differencing. However, that may not be enough when a series displays to much dependence on its own past values, with an autocorrelation function that converging very slowly, for example, inflation or interest rate time-series. Such time-series have covariance stationary processes fall between the exteremes of the series with unit-roots and those with short-memory that have absolutely summable autocovariance functions decaying geometrically, see discussion of *AR* (*p*) process in chapter 6. That is, the dependence of such series on own past values decays very slowly. Such series have *long-memory* and require differencing by a **fractional** order of integration rather than an integer order of *d* in order to capture the long-run parameters. The autocorrelation of a fractionally-differenced time series decay slow hyperbolically compared to short-memory exponential or geometric series as shown in the plot below. Such a time-series is said to have *square-summable* autocovariances $\sum_{j=0}^{\infty} \varphi_j^2 < \infty$, and longer memory compared to that with absolutely-summable autocovariance in the sense that its order of integration is factional as opposed to long-memory AR (p) process with based on non-fractional order of integration. In general, it must be analyzed by the ***autoregressive moving average fractionally integrated*** (**ARFIMA**) model. The ARFIMA applications are common in hydrology and was first introduced to econometrics by Granger and Joyeux (1980) who argued that the autocorrelation of ARMA decay exponentially, while that of ARFIMS more slowly hyperbolically; hence the latter can more effectively obtain separate estimates of the long-run and short-run dynamics.

Plot of exponential v. hyperbolic curves



The pure time-series ARIMA as three sets of parameters.

$$A(L)\ \mathrm{y}_t = (1 - L)^d\, \mathrm{y}_t = \propto + B(L)\varepsilon_t \qquad (9.3.1)$$

where $A(L)=1 - \rho_1 L - \rho_1 L^2 - . . . - \rho_p L^p$; $B(L)=1 - \theta_1 L - \theta_1 L^2 - . . . - \theta_q L^q$, the first *p* set determines the autoregressive polynomial in the *L* operator while the second *q* set that of the moving average polynomial in the *i.i.d*, residual process; and the integer order of integration parameter *d* , typically with *d*=1, that defines an *ARIMA* (*p, d, q*) model. The estimation of an *ARIMA* requires that the

$A(L)$ be invertible, so for differenced y_t^*, after multiplying both sides of (9.3.1) by $A(L)^{-1}$, we have

$$y_t^* = A(L)^{-1}(\propto + B(L)\varepsilon_t) \qquad (9.3.2)$$

That implies the characteristic roots of $A(L)$ polynomial to lie strictly outside the unit circle, for example with *AR* (1), $|\rho|$ must be less than 1. If the conditions are met, then the time-series will be representable by a *MA* (∞) model (see chapter 7), and estimated by ML based on the Kalman filter, a nonlinear procedure that predicts results the current stage of a time-series based exclusively on the information available from the previous location, examined in chapter 16. Similarly, stability requires the $B(L)$ to be invertible with the characteristic roots strictly outside the unit circle, so remultiplying the series the inverse of the $B(L)$ polynomial leads to an AR (∞), and that in turn requires, for example, a *MA* (1) $|\theta|<1$.

As discussed in chapter 7, long-memory time-series cannot rely on an *MA* (q) model that dies at exactly q lags, and though an AR (p) model has an infinite memory containing all the past residual values the process follows a geometric lag, quickly decaying for near-zero values. Moreover, d=1 differencing can result in removing the long-run dynamic effects from the time-series by *over-differencing* it. Granger and Joyeux (1980) suggested applying *fractional* ARIMA, or *ARFIMA* (*p, d, q*) with mean μ by allowing d in (9.3.1) to take on fractional values -0.5 < d < 0.5, written as

$$\emptyset(L)(1 - L)^d(\mathrm{y}_t - \mu) = \Theta(L)\varepsilon_t; \;\; \varepsilon_t \; i.i.d.(0, \sigma_\epsilon^2) \qquad (9.3.3)$$

where $(1 - L)^d$ is the fractional differencing operator based on the gamma function. If the inverse of $(1 - L)^d$ exists and $d < 1$, then an infinite series can be approximated by (***d*** **-1**) order of integration, see Hammilton (1994), chapter 15; esp. the appendix. For stationarity and invertibility, all roots of $\emptyset(L)$ and $\Theta(L)$ must be strictly outside the unit-circle and $|d|<0.5$; the process is non-stationary with an infinite variance if $d \geq 0.5$; in that case, we should diference the process before applying (9.3.3), for example for d=0.7, the process becomes

$$(1 - L)^{\boldsymbol{-0.3}}(1 - \mathrm{L})\,\mathrm{y}_t = \Theta(L)\varepsilon_t$$

The ARFIMA process displays long-memory if $d \in (0, 0.5)$ and intermediate memory or negative long-range dependence if $d \in (-0.5, 0)$.

There are two methods of the ARFIMA estimation, exact LM parametric one and by semi-parametric spectral function, see chapter 16. Here, we briefly mention in passing the former proposed by Sowell (1992) that must specify the *p* and *q* lag structures and then estimation the *full* ARFIMA conditional model by

$$y_t = (1 - L)^{-d}\left(\emptyset(L)\right)^{-1}\Theta(L)\varepsilon_t \qquad (9.3.4)$$

This approach first obtains the short-run effects by setting *d*=0; the long-run effects are obtained from fractional differencing process $(1 - L)^{-d}y_t$ using $\hat{d}$ values, see exercise Q 14.4 for an application.

Readings

Enders (2015, chapters 5 and 6), Hamilton (1994, chapter 1) on autoregressive distributed lags; Im, Pesaran and Shine (2003) proposed the panel data unit-root test. Granger & Joyeux (*J. T-S. A.* 1:15-29) introduced the analysis of slow long-memory series with ARFIMA, text discussion includes Pesaran (2015, section 15.8), and Hamiltion (1994, section 15.5).

Chapter 9 ARDL, Panel Unit-Root, ECM, & ARFIMA Exercises

Q9.1 Let the realized value of the $\{z_t\}$ sequence, and exogenous $\{y_t\}$ sequence, to be such that z_1=1 and all other values of z_i=0.

a. Using ARDL equation $y_t = \alpha_1 y_{t-1} + c_0 z_t + \varepsilon_t$, a one-unit shock in z_t has the initial effect of increasing y_t by c_o units. Use this equation to trace out the $\{z_t\}$ sequence on the time path of y_t.

b. Using ARDL equation $\Delta y_t = \alpha_1 \Delta y_{t-1} + c_0 z_t + \varepsilon_t$, a one-unit shock in z_t has the initial effect of increasing *the change in* y_t by c_o units. Use this equation to trace out the $\{z_t\}$ sequence on the time path of y_t.

Q9.2 Download *natural_gas_prices.dta* containing EU and US gas price time-series with 195 0bservations.

a. plot eur/us prices in levels and first differences for informal evidence of co-movement between the series, and then fit an ARDL model of eur on us; select the optimal lag number by AIC and comment on the outcome

Q9.3 Download *pennxrate.dta* containing real exchange rate data for a panel of large number of countries observed over 34 years.

a. Apply the Im, Pesaran & Smith (*IPS*) panel unit-root method to test if all *lnrxrate* series contains unit roots for the subset of OECD countries, explain the table outcome and comment on the test results.

b. Test for unit roots in **a.** by allowing for *serially correlated errors* (for this question you need to use *xtunitroot* regression command, thus, you must first inform Stata that the dataset has a panel structure).

Q9.4_ Download *campito.dta*, botanical data on the historical growth of tree trunks.

a. plot the series and its autocorrelation, estimate *ARMA*(2, 1) and *ARFIMA*, *ARFIMA* with *AR*(1), and comment on the outcome.

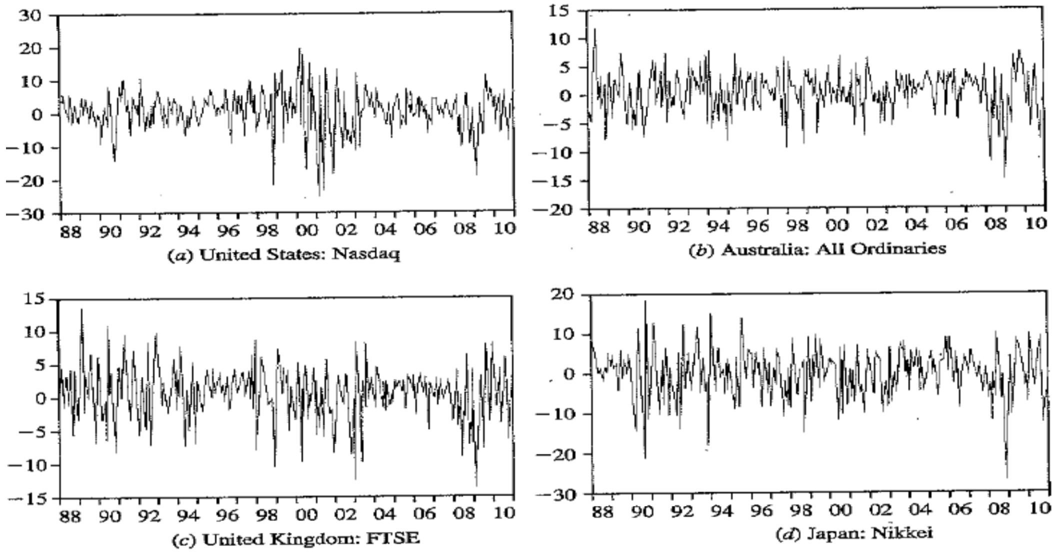
Chapter 10 Volatility Analysis, ARCH & GARCH Processes

Introduction

A non-stationary series exhibits a mean that is not stable over time. However, there are also other important unstable series that have constant means but with conditional variances that change over time. Models with non-constant variance are particularly well suited to analyzing the dynamics of financial time series that often display **volatility**, in addition to **clustering**, that is, radical changes in the series tightly compressed into very short time intervals.

Figure 10.1 shows four series of monthly returns to stock market prices that have long-run constant means around zero but deviate radically from their means in some periods, the series are *volatile*. Moreover, volatility is *clustered,* that is, periods of large deviations from the mean followed by other large deviations are closely packed together. These series have unstable variances, namely. they are heteroskedastic.

Figure 10.1-*Time series of returns to stock indices*



(*a*) United States: Nasdaq

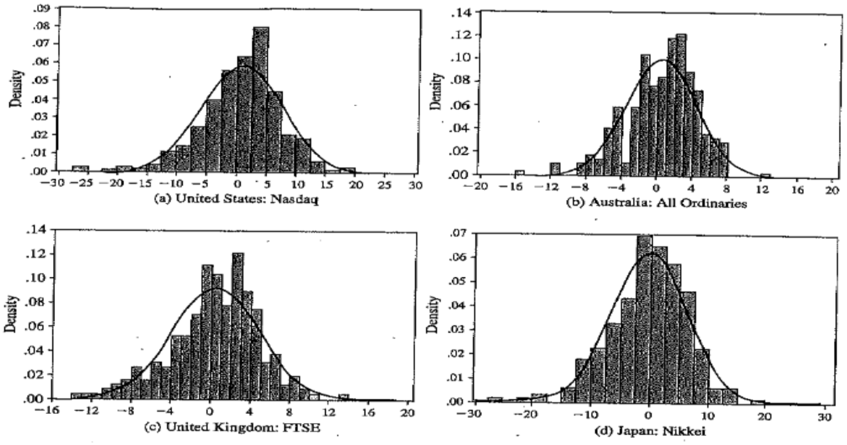
(*b*) Australia: All Ordinaries

(*c*) United Kingdom: FTSE

(*d*) Japan: Nikkei

Moreover, the histogram of the returns is not normally distributed, as it is evident from Figure 10.2 which imposes normal distribution on the top. Note how the unconditional distributions cluster around the mean and have fat tails relative to the normal distribution- the series have **leptokurtic** unconditional distributions. To analyze such distributions, we must define volatility as a function of the error e_t, or "news", or "shocks" in the financial markets, and accounting for e_{t-1} lag and clustering effects. The model employed for this type of analysis, introduced by Engle (1982), is called **Autoregressive Conditional Heteroscedasticity** or ***ARCH***, a particularly useful model for analyzing financial markets.

Figure 10.2-*Histograms of returns to stock indices*



(a) United States: Nasdaq

(b) Australia: All Ordinaries

(c) United Kingdom: FTSE

(d) Japan: Nikkei

10.1 *ARCH*

It is helpful for understanding *ARCH* process to contrast its mean and variance with those of the AR(1) process.

$$y_t = \phi + u_t;$$

$$u_t = \rho u_{t-1} + \varepsilon_t;\ |\,\rho| < 1\ \&\ \varepsilon_t \sim WN(0, \sigma^2)$$

The unconditional mean of this model is constant (zero) and its conditional mean varies over time, while the conditional and unconditional variances are constant, that is, independent of time.

Let us now allow the conditional variance to change over time while the conditional mean remains constant. We start with the simplest model with one period time lag *ARCH*(1), and then generalize to *ARCH* models with several lags, later.

$$y_t = \theta + u_t$$

$$u_t\,|\mathrm{I}_{t-1} \sim N(0, h_t) \qquad (10.1.1)$$

$$h_t = \alpha_0 + \alpha_1 u^2_{t-1}\ ; \qquad \alpha_0 > 0 \text{ and } 0 \leq \alpha_1 \leq 1$$

Note that y_t in (10.1.1) is run on just a constant called the **mean equation**. The second equation states that the error term's **conditional normality**, that is, that it is normally distributed conditional on information available at time *t-1*, with mean 0, and time-dependent variance h_t. The third equation makes the change in h_t a function of a constant term plus the lagged error squared at time *t-1*. In addition, we impose the condition that α_1 must be less than 1 in order to prevent h_t series to explode. The conditional normality means that time *t*-3, for instance, $u_3 | I_{2\sim} N(0, \ \alpha_0 + \alpha_1 u^2_2)$. *ARCH*(1) is the simplest example where h_t depends on one period lag u^2_{t-1}. The difference between *ARCH* (1) and *AR*(1) highlights the contrast between a mean non-stationary series and a variance non-stationary series; *AR*(1) has a time-varying conditional mean but constant conditional variance, while *ARCH*(1) has a time-varying conditional variance but constant conditional mean (unconditional mean and variance are time-independent in both *AR* and *ARCH*), see below.

i. *ARCH (1) moments*

(10.1.1) states that conditional on u^2_{t-1}, the mean of *ARCH*(1) for u_t is constant(zero), but its variance is not, h_t depends on time. Compare these with the unconditional first two moments of *ARCH* (1). The unconditional distribution of u_t is obtained by the standardized errors

$$(u_t/\sqrt{h_t} \mid I_{t-1}) = v_t \sim N(0,1);$$

since v_t has a standard normal distribution independent of $u_{t-1.}$ Therefore, the unconditional distribution of $(u_t/\sqrt{h_t}) = v_t \sim N(0,1)$, implies that v_t and u^2_{t-1} are independent, and we can thus write the unconditional mean of u_t as

$$E(u_t) = E(v_t) * E(\sqrt{\alpha_0 + \alpha_1 u^2_{t-1}}) = 0$$

because by assumption $E(\varepsilon_t)$=0. The unconditional variance is

$$Var(u^2_t) = E(u^2_t) = E(v^2_t) * E(\sqrt{\alpha_0 + \alpha_1 u^2_{t-1}})^2$$

Since by assumption, $E(v^2_t)$=1 and $E(u^2_t) = E(u_t - \bar{u})^2$ and $\bar{u}$ =0, we can rewrite

$$Var(u^2) = E(u^2_t) = \alpha_0 + \alpha_1 E u^2_{t-1} = \alpha_0 + \alpha_1 E(u^2_t),$$

since by assumption the errors are normally distributed over time, they are independent in different time periods, and hence $E(u^2_t) = E(u^2_{t-1})$, and finally we have

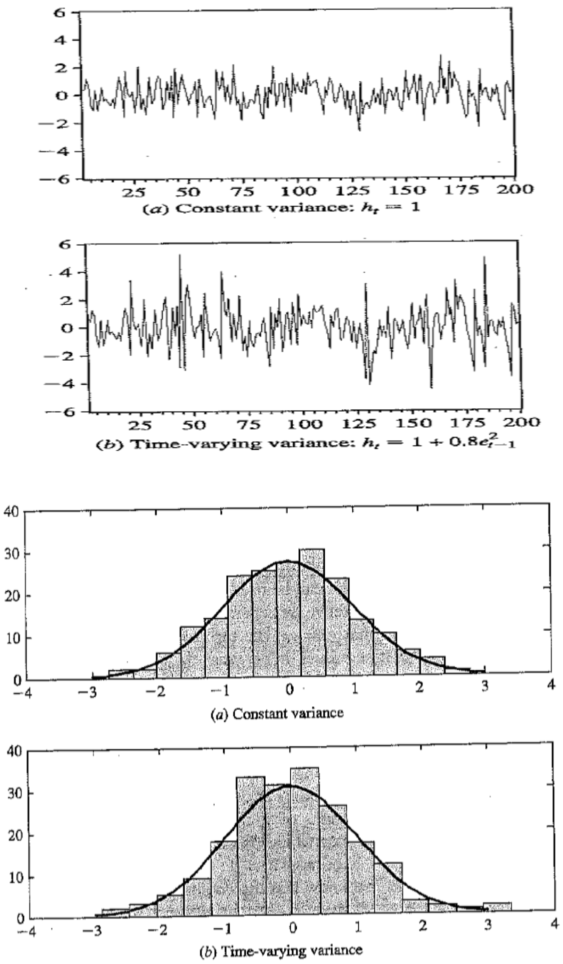
$$E(u^2_t)(1 - \alpha_1) = \alpha_0 \text{ or}$$

$$\sigma^2_t = Var(v^2_t) = \frac{\alpha_0}{1-\alpha_1}$$

namely. Constant unconditional variance (independent of time).

The *ARCH* models provide an effective method of analyzing financial risk, and financial markets usually employing volatility of variance as a measure of risk. The ability of *ARCH* models to employ post information on volatility to improve on forecasting of future risk therefore plays a critical rule in financial risk analysis. The models of *ARCH* and its generalized versions are common in risk management models of asset pricing, portfolio selection and options pricing; though they also prove effective in instability analysis of inflation or growth; see applied examples. Figure 10.3 presents a hypothetical example of two time series with constant and variable variances.

Figure 10.3-Constant & Time-varying Variances



(*a*) Constant variance: $h_t = 1$

(*b*) Time-varying variance: $h_t = 1 + 0.8e^2_{t-1}$

(*a*) Constant variance

(*b*) Time-varying variance

ii. General p-order ARCH

We can generalize ARCH to higher orders. Suppose an observable time-series for y_t takes a *p*-order *AR*(*p*) of the form

$$y_t = c + \phi_t y_{t-1} + \phi_2 y_{t-2} + \ldots + \phi_p y_{t-p} + u_t$$

with a white noise u_t as

$$E(u_t)=0$$

$$E(u_t u_\tau)=\begin{cases} \sigma^2 & \text{for } t=\tau \\ 0 & \text{otherwise} \end{cases}$$

Covariance stationary characteristic of a time-series requires that the characteristic roots of the series

$$1 - \phi_1 z - \phi_2 z^2 - \ldots - \phi_p z^p = 0 \qquad (10.1.2)$$

be *outside* the unit circle.

We obtain the conditional mean of this model from the linear forecast of the level of y_t as

$$E(y_t | y_{t-1}, y_{t-2}, \ \ldots) = c + \phi_t y_{t-1} + \phi_2 y_{t-2} + \ldots + \phi_p y_{t-p}$$

While the unconditional mean of $y_t$is

$$E(y_t) = c/(1 - \phi_t - \phi_2 - \ldots - \phi_p)$$

Thus, the conditional mean of $y_t$changes over time while its unconditional mean is constant. So far, we have assumed constant variance σ^2, but the conditional variance of u_t can change over time, for example, when forecasting a volatile time-series such as inflation. Based on the above additive error term model, we can employ an *AR*(*m*)process to describe this.

$$c + \phi_t y_{t-1} + \phi_2 y_{t-2} + \ldots + \phi_p y_{t-p} + u_t \text{ by}$$

$$u_t^2 = \zeta + \phi_t u_{t-1}^2 + \phi_2 u_{t-2}^2 + \ldots + \phi_m u_{t-m}^2 + w_t \qquad (10.1.3)$$

With a white noise $w_t$process

$$E(w_t)=0$$

$$E(w_t\text{w})=\begin{cases}\lambda^2 & \text{for } t=\tau \\ 0 & \text{otherwise}\end{cases}$$

The conditional variance based on the previous m periods is given by

$$E(u_t^2|u_{t-1}^2,u_{t-2}^2,\ldots)=\zeta+\phi_t u_{t-1}^2+\phi_2 u_{t-2}^2+\ldots+\phi_m u_{t-m}^2$$

And the covariance stationarity requires

$$1-\alpha_1 z-\alpha_2 z^2-\ldots-\alpha_m z^m=0$$

or, given $\alpha_j>0$, $\alpha_1+\alpha_2+\ldots+\alpha_m<1$.

Given the above, we obtain the unconditional variance from

$$\sigma^2=E(u_t^2)=\ \zeta/((1-\alpha_1-\alpha_2-\ldots-\alpha_p)$$

Finally, we obtain the s-period forecast of $\hat{u}_{t-s|t}^2$from

$$(\hat{u}_{t-j|t}^2-\sigma^2)=$$

$$\alpha_1(\hat{u}_{t-j-1|t}^2-\sigma^2)+\alpha_2(\hat{u}_{t-j-2|t}^2-\sigma^2)+\ldots+\alpha_m(\hat{u}_{t-j-m|t}^2-\sigma^2)$$

for j=1, 2, . . . , s. The white noise error process $u_t$of m-order is denoted as u_t~ *ARCH* (m).

However, the above linear model would be more tractable if specifying a multiplicative disturbance. Suppose this alternative representation has a serially dependent error term given by

$$u_t=\sqrt{h_t}.v_t\ ,\ E(v_t)=0\ \&\ E(v_t^2)=1 \qquad (10.1.4)$$

Moreover, assume

$$h_t=\zeta+\alpha_t u_{t-1}^2+\alpha_2 u_{t-2}^2+\ldots+\alpha_m u_{t-m}^2 \qquad (10.1.5)$$

This implies

$$E(u_t^2|u_{t-1},u_{t-2},\ldots)=\zeta+\phi_t u_{t-1}^2+\phi_2 u_{t-2}^2+\ldots+\phi_m u_{t-m}^2$$

Therefore, the linear forecast of this multiplicative *ARCH* (m) is also the conditional expectation.

Furthermore, the conditional variance of *ARCH* (m)can be obtained by the substituting of the squared equation (10.1.4) into equation (10.1.3), using (10.1.5) for h_t

$$u_t^2 = h_t.v_t^2 = h_t + w_t$$

or

$$w_t = h_t.(v_t^2 - 1) \qquad (10.1.6)$$

Thus, the conditional variance of *ARCH* (*m*) changes with *t*, even though the unconditional variance of (10.1.3), λ^2, is constant.

However, (10.1.6) reflects the fourth moment of u_t, or the second moment of w_t; that moment does not exist for all stationary *ARCH* models. For example, a real solution does not exist λ for an *ARCH* (1) unless $\alpha_t^2 < \frac{1}{3}$, see Hamilton (1994, p. 660). However, the above linear model would be more tractable by specifying a multiplicative disturbance.

We examine econometric applications of volatility models with *ARCH* (1); generalization to more complex volatility models is straight forward.

iii. Testing

A Lagrange Multiplier (*LM*) is the usual test employed to detect *ARCH* offsets. The test procedure is as follows:

First estimate the mean equation, that is, a regression of y_t on a constant, though variables may also be included, in order to obtain e_t. Then, for example, to test for *ARCH* (1) effects, regress equation

$$\hat{u}_t^2 = \gamma_0 + \gamma_1 \hat{u}^2_{t-1} + \mu_t$$

where μ_t is a random error, and test for

$$H_0: \gamma_1 = 0 \ vs. \ Ha: \gamma_1 \neq 0.$$

Given *ARCH* effects ($\gamma_1 \neq 0$), R^2_μ will be relatively high due to dependence of $\hat{u}_t^2$ on $\hat{u}^2_{t-1}$. *LM* test statistics is $(T\text{-}q)R^2_u$ where *T* is the sample size and *q* is the order of terms in the equations. With *ARCH* (1), we reject H_0 if $(T\text{-}q)R^2_u > \chi^2_q$ at a given confidence level.

Example: estimated mean equation for returns on shares of a light bulb producing company is

$$r_t = \beta_0 + e_t$$

This is the *mean equation* with just an intercept and where r_t is the monthly return. Obtain estimated residual and run an autoregressive equation of residuals squared:

$$\hat{e}_t^2 = 0.908 + 0.353\, \hat{e}^2_{t-1}$$

$$\textit{t-ratio} \quad (8.41)$$

and $R_e^2 = 0.124$, $T = 500$.So $(T\text{-}1)R^2 = 499 * 0.124 = 61.876 > \chi^2_1$ at $\alpha_{0.5\,\%} = 3.841$, so we reject the null; there are *ARCH* effects in the residual of *ARCH*(1).

10.2 *Estimation*

The standard applications of the maximum likelihood estimator for a normally distributed residual with zero mean and constant variance lead to the first-order conditions that are easy to solve since they are linear. That is not the case with the *ARCH* and *GARCH/MLE* applications with non-linear first-order equations. As an example, consider a simple *ARCH*(1) process with a normally distributed u_t=y_t−βx_t, with a zero mean and a constant variance$\sigma^2$and define $u_t = v_t\sqrt{h_t}$. Given each realization of u_t, and $h_t$as the conditional variance, the joint likelihood of $u_t$realization t=1, 2, …,T is

$$\text{L}=\prod_{t=1}^{T}\left(\frac{1}{\sqrt{2\pi h_t}}\right)\exp\,(\frac{-u_t^2}{2h_t})$$

Therefore, the log-likelihood function becomes

$$lnL = -\frac{T}{2}\ln(2\pi) - 0.5\sum_{t=1}^{T} lnh_t - 0.5\sum_{t=1}^{T}(\frac{u_t^2}{2h_t})$$

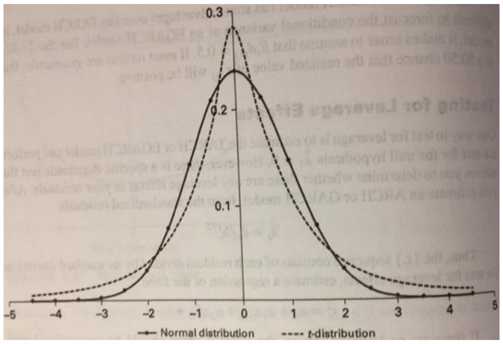
Now substitute for the conditional variance of *ARCH*(1) process $h_t = \alpha_0 + \alpha_1 u_{t-1}^2$, given u_t=y_t−βx_t leading to

$$lnL = -\frac{T-1}{2}\ln(2\pi) - 0.5\sum_{t=2}^{T}\ln\,(\alpha_0 + \alpha_1 u_{t-1}^2) - \frac{1}{2}\sum_{t=2}^{T}[\frac{(y_t - \beta x_t)^2}{(\alpha_0 + \alpha_1 u_{t-1}^2)}]$$

Notice that we lose the initial observation for u_0 at t=1 period. There are no analytical solutions to the first-order conditions for maximization of this equation and numerical optimization cannot guarantee optimal solutions if the partial derivatives are close to zero.

So far, we assumed normally distributed errors whereas the unconditional distribution of many time-series, particularly financial assets, have flatter tails than those from the Gaussian family, namely, higher probability of a very large loss (or gain). Then the normally distributed maximum likelihood is not an appropriate estimator. However, we can use the same basic approach with non-Gaussian errors drawn from a *t*-distribution for *fat-tailed* distribution as shown in Figure 10.5

Figure 10.4-Normal and *t*-distributions:3 degrees of freedom



More specifically, the *ML* application provides an estimate of the degree of freedom *v* as a parameter of *t*-distribution conditional on the scale parameter of that distribution M_t from its density given by

$$f(u_t|M_t) = \frac{\Gamma[\frac{v+1}{2}]}{\sqrt{\pi v}\,\Gamma(\frac{v}{2})} M_t^{-1/2}[1 + \frac{u_t^2}{M_t v}]^{-(v+1)/2}$$

where $\Gamma(.)$ is the gamma function, see Hamilton (1994, p. 662) for details.

However, even if the assumption of a Gaussian error distribution is invalid, we can still employ the ARCH process to obtain consistent parameter estimates; such an estimator, called a **Quasi-Maximum Likelihood estimator** (***Q-MLE***), can provide consistent linear forecasts of the squared value of the error observations even if the distribution of $u_t$is non-Gaussian as long as v_t satisfies $E(v_t) = 0$ & $E(v_t^2) = 1$ conditions stated for equation (10.1.3) above, though the standard errors would have to be adjusted, see Hamilton (1994, p. 663).

i. *Example (a):* we estimate the ARCH model by maximum likelihood estimator using numerical methods, so the starting values must be chosen carefully to obtain global and not just local optimized estimates. For the mean equation (10.1.3), the result is

$$\hat{r}_t = \hat{\beta}_0 = 1.063 \qquad (10.2.1)$$

$$Var(\hat{e}_t)=\hat{h}_t=\hat{\alpha}_0 + \hat{\alpha}_1\hat{e}^2_{t-1} \; ; =0.642+0.569\, \hat{e}^2_{t-1}$$

t-ratio (5.54)

The t-ratio suggests significant ARCH effects. Note that both $\alpha_0>0$ & $\alpha_1>0$, the conditions required for positive variances of the ARCH model and its convergence.

ii. Forecasting

Investment decisions on shares are taken on the basis of their risk, not just their returns, and volatility offers a measure of risk. To follow from the example of the lighting company, from (2) we obtain

Example (b):

$$\hat{r}_t=\hat{\beta}_0=1.063 \qquad (10.2.2)$$

Since (10.2.2) does not change with time, it is both the conditional and unconditional mean return. We also obtain an estimate for $\hat{e}_t=r_t - \hat{r}_t$ in order to estimate 1-step-ahead forecast $\hat{h}_{t+1}=\hat{\alpha}_0 +$

$\hat{\alpha}_1 (r_t - \hat{\beta}_0)^2$ after substitution $\hat{\beta}_0$ for $\hat{r}_{t.+1}$ (since $\hat{\beta}_0$ is independent of t ($\hat{r}_t$ in (10.2.1) only changes with $\hat{e}_t$, so $\hat{r}_{t.}= \hat{r}_{t.+1}=(\hat{\beta}_0)$. Given $\hat{e}_t=(r_t - 1.063)^2$, we finally have

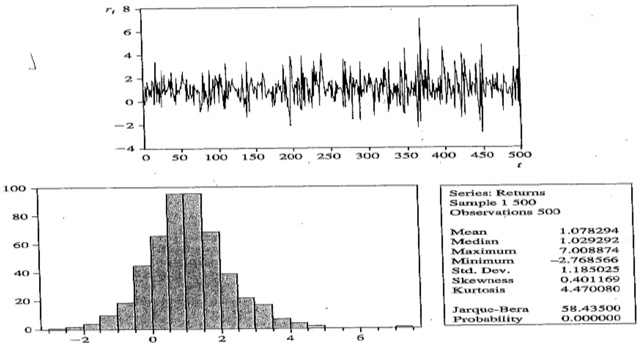
$$\hat{h}_{t+1}=\hat{\alpha}_0 + \hat{\alpha}_1 (r_t - \hat{\beta}_0)^2=0.624+5.69(r_t - 1.063)^2 \qquad (10.2.3)$$

Note that $\hat{h}_{t+1}$ does change by time because of its dependence on r_t at time t. The conditional variance for this time series and the histogram of its returns are shown below (Figure 10.5) - a large change is observable around 370. Note the difference between the graphs of the series r_t which may be above or below the mean and its variance h_t, a squared and, therefore, a positive series; however, volatility and clustering are evident in both.

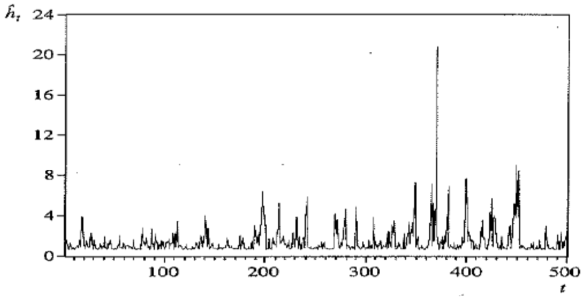
10.3 *GARCH*

Estimating of *ARCH* (p) requires p+1 parameter estimates. The loss in degrees of freedom when p is large results in inaccurate estimates. The **generalized *ARCH*, *GARCH,*** model is designed to capture long dynamic effects with fewer parameters. *GARCH*(1,1) is the simplest version of the general GARCH(p,q) where p is the number of lags of e terms and q is the number of lags of h terms. Specifically, we can generalize *ARCH* (M) process (10.2.1)-(10.2.3) to allow the conditional variance to depend on infinite lags of u^2_{t-1}.

Figure 10.5-*Time series and histogram of Lighting share prices*



Plot of conditional Variance



$$h_t = \zeta + \pi(L)u_t^2 \qquad (10.3.1)$$

where $\pi(L) = \sum_{j=1}^{\infty} \pi_j L^i$.

It is natural to parametrize $\pi(L)$as the ratio of two finit order polynomials by

$$\pi(L) = \frac{\alpha(L)}{1 - \delta(L)} = \frac{\alpha_1 L^1 + \alpha_2 L^2 + \ldots + \alpha_m L^m}{1 - \delta_1 L^1 + \delta_2 L^2 + \ldots + \delta_r L^r}$$

Assuming the roots of $1 - \delta(z)$=0 from equation (1) are

$$[1 - \delta(L)]h_t = [1 - \delta(L)]\zeta + [1 - \delta(L)]\pi(L)u_t^2$$

or

$$h_t = k + \delta_1 h_{t-1} + \delta_2 h_{t-1} + \ldots + \delta_r h_{t-r}$$

$$+\alpha_t u_{t-1}^2 + \alpha_2 u_{t-2}^2 + \ldots + \alpha_m u_{t-m}^2 \qquad (10.3.1)$$

where now $k \equiv [1 - \delta_1 + \delta_2 + \ldots + \delta_r]\zeta$. (10.3.1) is the generalized autoregressive conditional heteroscedasticity model, and denoted $u_t \sim$ *GARCH* (*r*, *m*). To ensure non-negative h_t, (7) must have k>0, α_j ≥0, and δ_j ≥0. Then $u_t^2$is covariance-stationary if $w_t$has finite variance and the roots of

$$1 - \delta_1\alpha_1 z - \delta_2\alpha_2 z^2 - \ldots - \delta_p\alpha_p z^p = 0$$

or if

$$\delta_1\alpha_1 z + \delta_2\alpha_2 z^2 + \ldots + \delta_p\alpha_p z^p < 1.$$

If this condition for covariance stationary characteristics holds, then the unconditional mean of u_t^2 is

$$E(u_t^2) = \delta^2 = k/[\delta_1\alpha_1 z + \delta_2\alpha_2 z^2 + \ldots + \delta_p\alpha_p z^p]$$

We can now understand the reason for the popularity of *GARCH* over *ARCH* models. *GARCH*(1,1) requires estimation of three parameters(δ , α, β), whereas if *p* is large, for instance *p*>6, *ARCH*(*p*) requires at least seven parameter estimates. In line with the principle of parsimony, we work with *GARCH*, especially *GARCH* (1,1), in econometric analysis of volatility and its forecasting. This outcome is not surprising in the light of the discussion in chapter 6. Note the similarity in moving from *ARCH* to *GRACH* to that of moving from distributed lags model to a dynamic one by substituting lags of dependent variable for lags of the independent variable, and with similar results, namely, both *GARCH* and dynamic models conserve degrees of freedom to offer more parsimonious models with improved forecasting ability.

Example (c): continuing with the same lighting bulb data:

$$\hat{r}_t = 1.049$$

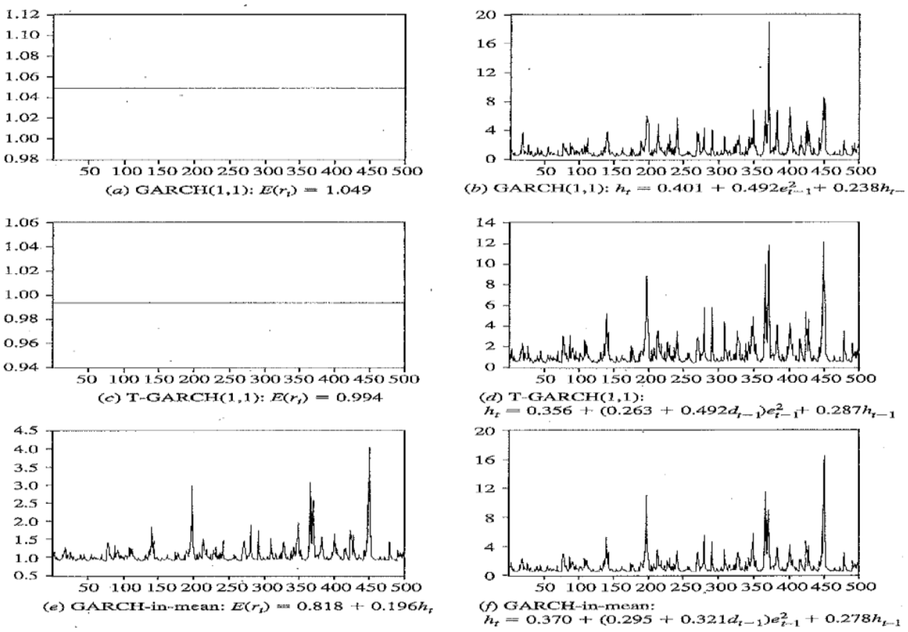
$$Var(\hat{e}_t) = \hat{h}_t = \hat{\alpha}_0 + \hat{\alpha}_1 \hat{e}^2_{t-1} + \hat{\alpha}_2 \hat{h}_{t-1} \text{ ; } = 0.401 + 0.492\, \hat{e}^2_{t-1} + 0.228\hat{h}_{t-1}.$$

t-ratio (4.83) (2.14)

Significant $\hat{h}_{t-1}$ points to improved modeling with $\hat{h}_{t-1}$ included.

The graphs mean and variance of this example are shown in Figure 10.6.

Figure 10.6-*Estimated means and variances of ARCH models*



(a) GARCH(1,1): $E(r_t) = 1.049$

(b) GARCH(1,1): $h_t = 0.401 + 0.492e^2_{t-1} + 0.238h_{t-}$

(c) T-GARCH(1,1): $E(r_t) = 0.994$

(d) T-GARCH(1,1): $h_t = 0.356 + (0.263 + 0.492d_{t-1})e^2_{t-1} + 0.287h_{t-1}$

(e) GARCH-in-mean: $E(r_t) = 0.818 + 0.196h_t$

(f) GARCH-in-mean: $h_t = 0.370 + (0.295 + 0.321d_{t-1})e^2_{t-1} + 0.278h_{t-1}$

i. *TGRACH & EGARCH for Asymmetric response*

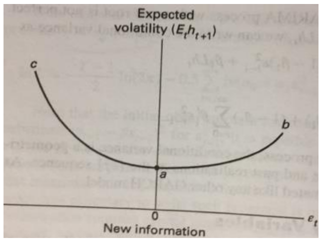
Note that *GARCH*(1,1) variance is only affected by the squared value of the last period's return, therefore, the positive/negative signs for return has no effect on volatility, thus, the *GARCH* model treats bad and good news or shocks symmetrically. However, negative news hits financial markets more severely than positive news, inducing greater volatility. The **threshold *GARCH***, or ***TGARCH*** model provides such asymmetric responses to conditional variance in order to allow greater volatility impact on returns of bad relative to good news. *TGRACH* (1, 1) conditional variance function is

$$h_t=\sigma^2_t= \omega+ \alpha e^2_{t-1} + \gamma D_{t-1}e^2_{t-1}+ \beta h_{t-1} \qquad (10.3.3)$$

where $D_t \begin{cases} 1 if\ e_t < 0 & (bad\ news) \\ 0 if\ e_t \geq O & (good\ news) \end{cases}$

When the lagged return is positive, D_t=0, the effect on the current conditional variance remains as before, that is α. However, if the lagged return is negative (bad in previous period), D_t=1 and the effect equal (α+ γ)> α . With γ=0 restriction imposed, we have a symmetric response; the asymmetric response factor is called the **leverage effect** because a negative shock to stock prices raises the aggregate debt/equity ratio, thereby increasing leverage. Thus, with the TGARCH process, volatility is driven by both size and *sign* of shocks. Figure 10.7 shows segment ac is steeper than ab, thereby a positive $u_t$shock will have a smaller effect on volatility than a negative shock of equal size.

Figure 10.7-The Leverage Effect



Example (d): r_t=0.994

$\hat{h}_t$= 0.356+0.263e^2_{t-1} +0.492$D_{t-1}e^2_{t-1}$ +0.287h_{t-1}

t-ratio (3.267) (2.405) (2.488)

In this example, negative e_t (bad news) increases volatility by (0.263+0.492)>0.263 compared with good news when e_t >0. The graph above illustrates the outcome; note the heightened volatility around 200.

One problem with *GRACH* and *TGARCH* models is the restriction that all of the estimated coefficients are positive. A more flexible model that is not subject to that restriction expresses volatility as a log-linear function. Consider for example a *GRACH* (1, 1) process.

$$lnh_t = \alpha_0 + \alpha_1(\varepsilon_{t-1}/h^{1/2}_{t-1}) + \gamma_1|\varepsilon_{t-1}/h^{1/2}_{t-1} + \beta_1 lnh_{t-1} \qquad (10.3.4)$$

(10.3.4) is known as *exponential GRACH* or *EGARCH*. EGRACH have some advantages over T*GARCH*. First, since (10.3.1) has a log-linear functional form, it does not require non-negative coefficient estimates because lnh_t cannot be negative; relaxing that restriction results in a more flexible functional form. Second, the *EGARCH* model uses a standardized value of shocks ε_{t-1}, divided $h^{1/2}_{t-1}$, giving it a natural interpretation expressed in unit-free measurement.

ii. GARCH-M

Financial markets show a positive relationship between risk and return; high risk stocks offer high returns. To allow for the risk *vs.* return trade-off, we define the mean equation h^2_t to be a function of volatility directly. This is the ***GRACH*-in-mean**, or ***GARCH-M*** model.

$$y_t=\beta_0+\theta\, h_t + e_t\ ;\ \ e_t\,|I_{t-1}\sim (0,\, h_t)$$

$$h_t=\delta+\alpha e^2_{t-1} +\beta h_{t-1};\ \delta > 0,\ 0\leq\alpha 1\leq 1\ \&\ 0\leq\beta 1\leq 1$$

The last equation shows the effect of conditional variance on the dependent variable measure of volatility. Note that even though the mean of the error term in *GRACH_M* is zero, therefore constant, the mean of h_t is unlikely to be constant once it is defined as a function of h_{t-1}. .

Example (e):

$$\hat{r}_t=0.818+0.196h_t$$

$$\textit{t-ratio}\quad (2.915)$$

$$\hat{h}_t= 0.370+0.295e^2_{t-1} +0.321D_{t-1}e^2_{t-1} +0.2787h_{t-1}$$

$$(3.426)\qquad (1.979)\qquad\qquad (2.678)$$

The estimations show increased volatility due to higher risk increases return by a factor of 0.196.

The graphs of figure 7 above show that expected mean is no longer constant. (compare means in the graphs for *GARCH & TGRACH*.)

Finally, we note that the standard *GARCH* volatility equation can be extended to include exogenous variables such as the volume of financial markets:

$$h_t=\sigma^2_t= \omega+ \alpha e^2_{t-1} + \beta h_{t-1}+\gamma x_t$$

where *x* is a positive exogenous variable.

iii *Multivariate GARCH*

We may be interested in *GARCH* dynamics of several time-series simultaneously. Multivariate *GRACH* allows mapping how volatility shock to one variable could affect the volatility of related variables. Consider a simple two-variable, y_{1t} & y_{2t}, with only two processes

$$u_{1t} = \sqrt{h_{11t}}.\,v_{1t}$$

$$u = \sqrt{h_{22t}}.\,v_{2t}$$

with $var(v_{1t})$=1 & $var(v_{2t})$=1, so h_{11t}& h_{22t} represent the conditional variances of u_{1t} & u_{2t}, and $h_{12t}$the conditional covariance, that is $h_{12t} = E_{t-1}u_{12t}u_{12t}$. Then we can construct a simple multivariate *GARCH* (1, 1) process to allow volatility spill-overs.

$$h_{11t} = \alpha_{10} + \alpha_{11}u^{2}_{1t-1} + \alpha_{12}u_{1t-1}u_{2t-1} + \alpha_{13}u^{2}_{2t-1}$$

$$+ \beta_{11}h_{1t-1} + \beta_{12}h_{12t-1}\varepsilon_{2t-1} + \beta_{13}h_{22t-1} \qquad (10.3.5)$$

$$h_{12t} = \alpha_{20} + \alpha_{21}u^{2}_{1t-1} + \alpha_{22}u_{1t-1}u_{2t-1} + \alpha_{23}u^{2}_{2t-1}$$

$$+ \beta_{21}h_{1t-1} + \beta_{22}h_{12t-1}\varepsilon_{2t-1} + \beta_{23}h_{22t-1} \qquad (10.3.6)$$

$$H_{22t} = \alpha_{30} + \alpha_{31}u^{2}_{1t-1} + \alpha_{32}u_{1t-1}u_{2t-1} + \alpha_{33}u^{2}_{2t-1}$$

$$+ \beta_{31}h_{1t-1} + \beta_{32}h_{12t-1}\varepsilon_{2t-1} + \beta_{33}h_{22t-1} \qquad (10.3.7)$$

That is, the conditional variances, h_{11t} & h_{22t}, of each variable depends on its own and other lags; the conditional covariance between them, h_{12t}, on the lagged squared error and their product. It would clearly be hard to implement this model of volatility spill-over involving many parameters estimates; even in this simple case, there are 21 parameters and the numbers grow rapidly as the order of *GRACH* process increases, with mean estimation and explanatory variables added. Moreover, solution for maximization of (10.3.5)-(10.3.7) cannot be obtained analytically and must rely on numerical iterative methods. There will then be convergence problems. If the model is overparametrized, and a coefficient estimate has a large confidence interval, so slight changes to the estimate will have little impact on maximization; it will be difficult to pin down its value, resulting in non-convergence. In order to avoid these problems, multivariate *GARCH* models employ suitable restrictions on (10.3.5)-(10.3.7) parameters. Such restrictions typically remove much of the interactive terms which of interest to volatility spill-over analysis, see Enders (2013,pp. 167-9).

iv. Volatility Impulse-Response Function

The following approach can plot the dynamics of volatility shocks for the (10.3.5)-(10.3.7) system despite the above difficulties.

For example, from h_{11t}, the one-step ahead forecast h_{11t+1} is

$$E_th_{11t+1} = \alpha_{10} + \alpha_{11}u^{2}_{1t} + \alpha_{12}u_{1t}u + \alpha_{13}u^{2}_{2t} + \beta_{11}h_{1t} + \beta_{12}h_{12t}\varepsilon_{2t} + \beta_{13}h_{22t}$$

Updating this equation by two periods and obtaining its conditional expectation, we have

$$E_th_{11t+2} = \alpha_{10} + \alpha_{11}u^{2}_{1t+1} + \alpha_{12}u_{1t+1}u_{2t+1} + \alpha_{13}u^{2}_{2t+1} + \beta_{11}h_{1t+1} + \beta_{12}h_{12t+1} + \beta_{13}h_{22t+1}$$

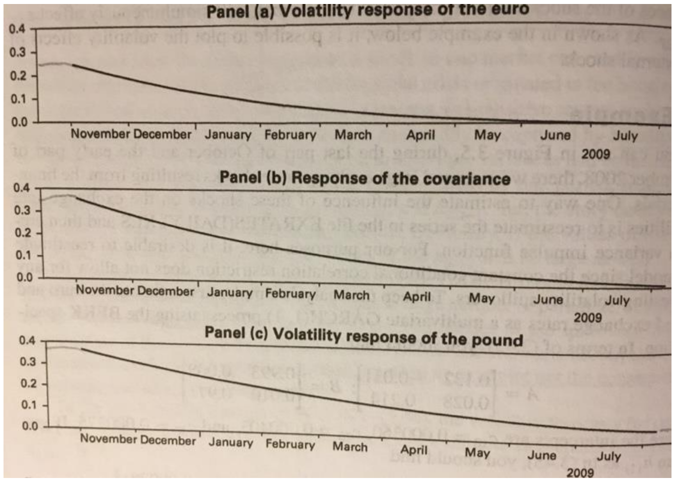
Since $E_t\varepsilon^{2}_{it+2}$= E_th_{iit+2},, and $E_t u_{it+2}u_{jt+2}$= E_th_{ijt+2}, the above simplifies to

$$E_th_{11t+2}= \alpha_{10}+(\alpha_{11} +\beta_{11})E_t h_{11t+1}+(\alpha_{12} +\beta_{12})E_t h_{12t+1}++(\alpha_{13} +\beta_{13})E_t h_{22t+1}$$

The differences between the volatility forecasts for any two sets of the initial values provide the impulse-response function in this approach. The procedure requires disturbing the conditional volatility forecast E_Th_{11T+i}, for t=1, 2, . . . , T, and i=1, 2, … by one or more of the ε_{iT} in order to obtain its forecast $E_{T*}h_{11t+i}$. Then, the difference between the two sets of forecasts, that is [$E_{T*}h_{11t+I}$ - E_Th_{11t+j}], constitutes the *I-R* function in this approach; if an external shock affects both u & u_{2t}, we can plot its volatility effects.

Example: Several exchange rates shocks from the financial crisis occurred in the late October-early November 2008. We select a set of external shocks equal to the actual residuals on October 29, 2008 to both error terms and obtain the corresponding volatility forecasts using the values of u_{1T*} & u_{2T*}as the initial shocks. A comparison of these values with the forecast volatilities obtained from actual values of u_{1T} & u_T appear in Figure10.8 (a-c), showing volatility increases of the euro and the British pound (panels a & c); though the latter displays sharper volatility that persisted to mid-2009. Therefore, the forecast covariance between the two (panel b) is higher than otherwise.

Figure 10.8-Volatility Impulse-Response Plots



Readings

For textbook discussion, see Hamilton (1994, chapter 21), Enders (2015, chapter 3). Engle (1982) developed the ARCH volatility model.

Chapter 10 Volatility Analysis, ARCH & GARCH Exercises

Q10.1 Download the data set *byd.dta* that contains a single time-series on the returns from purchase of shares in a lighting bulb company. Replicate the steps and the results reported in the text by following the Examples (a) through (e).

Q10.2_ Suppose that the $\{\mu_t\}$ sequence is the *ARCH* (*q*) process

$$\mu_t = v_t(\alpha_o + \alpha_1\mu_{t-1}^2 + \ldots + \alpha_q\mu_{t-q}^2)^{1/2}$$

Show that the conditional expectation of $E_{t-1}\mu_t^2$has the same form as the conditional expectation of (2).

Q10.3_ Consider the *ARCH-M* model represented by

$$y_t = \beta_0 + \theta h_t + e_t \,;\ \ e_t\,|I_{t-1} \sim (0,\, h_t)$$

$$h_t = \delta + \alpha e^2_{t-1} + \beta h_{t-1} \,;\ \delta > 0,\ 0 \leq \alpha 1 \leq 1\ \&\ 0 \leq \beta 1 \leq 1$$

where e_t is a white noise error; assume for simplicity that $Ee_t^2 = Ee_{t-1}^2 = \ldots = 1$.

Find the unconditional mean of Ey_i. How does the change in δ affect the mean? Show that changing β & δ from (-4, 4) to (-1, 1) preserves the mean of $\{y_t\}$ sequence

Q10.3 Download *WPI_US.dta* for the US quarterly wholesale price index.

a. Fit a constant only model by *OLS* and test for *ARCH* effects using the *LM* test.

b. Fit a *GARCH* (1, 1) for the conditional variance; the *ARMA* mean modeled as *AR*(1) & *MA*

(1 4) to control seasonality.

c. Fit the *EGRACH* model to obtain evidence for differences in unanticipated price increases news and price decreases news.

Chapter 11 Non-parametric and semi-parametric econometrics

Introduction

We rely on the density function for easily visualized measures of central tendency (mean, dispersion, etc.) of a distribution of a variable such as income, or consumption. Histograms provide a starting point for measures of central tendency by grouping the observations into "bins", but they can be misleading as the result depend arbitrarily on the width and the number of bins chosen. If you expand or reduce the choice of the bin boundary, the histogram's shape will change, and if the data is continuous, namely, for total expenditure, then the graph will display a discrete distribution where none exists. Non-parametric methods of density estimation provide an alternative that avoids these problems.

11.1 *Nonparametric Density Estimation*

The simplest is to have the observations on, say, income on the *x*-axis and define a *sliding* "band" for each point from the fraction of sample observations that is "near" to *x*, and plot the outcome as the density at the mid-point of the band. This would define the so-called naïve density estimator, given *h* as its ***band width,*** and 2*h* its ***window width.*** At each *x* in the sample, the estimator gives a weight of 1 to each point within *h*/2 on either sides of *x* and zero otherwise, thereby estimating a total score as the fraction of the sample size divided by *h*:

$$\tilde{f}(x) = \frac{1}{nh}\sum_{i=1}^{n} 1\,(-\frac{h}{2} \leq x - x_i \ \leq \frac{h}{2}). \qquad (11.1.1)$$

With only a few points, we need to widen the band to bring in data from the difference part of the distribution, risking bias estimates by including dissimilar observations. However, as the sample size increases, the bandwidth shrinks toward zero; the bandwidth becomes smaller at a slower rate than the rate of increase in the sample size in order to ensure consistent estimates. (11.1.1) is the so called *naïve* nonparametric density estimator; it is a simple example of the ***kernel*** method of nonparametric density estimation, with a *rectangular kernel* that give all the points within the band equal weight of 1, instead of more weight to observations closer to *x* and less to those that are father away. The latter problem can be dealt with by defining a kernel indicator function K to rewrite the naïve estimator as

$$\tilde{f}(x) = \frac{1}{nh}\sum_{i=1}^{n} K\,(\frac{x-x_i}{h}). \qquad (11.1.2)$$

Because the kernel is a weighting function it must have the following properties

i) It should be positive and integrate to unity over the band,

ii) It should be symmetric around zero so the points below zero receive the same weights as those with the same distance above zero

iii) It should be decreasing in the absolute value of its argument.

The rectangular kernel, for instance, puts equal weight on all observations in the band. Therefore, it does not have the last property. There are several well-known kernel estimators based on (11.1.2) that meet i-iii conditions, each based on a different weighing scheme for *K*. The three most commonly employed are the Epanechniko kernel, the Gaussian kernel, and the quartic or "biweight" kernel. In each of the following *h* is the bandwidth and $t_i = (x\text{-}x_i)/h$.

The **Epanechnikov** kernel $K_E(t_i)$ is of the form:

$$K_E(t_i) = \frac{0.75(1-0.2t_i^2)}{\sqrt{5}} \; t_i^2 < 5;$$

$$= 0 \; otherwise$$

The Epanechnikov weights have an inverted U-shape curve, falling to zero at the band's edges. The Gaussian kernel is a classic symmetric density function that does not use a discrete band; it gives no weight to out of the band observations, but within band observations all receive some weights. The Gaussian normality, however, ensures the density estimate at point *x* gives very little weight to any observation further than 3*h* from *x*.

The **Gaussian** kernel $K_G(t_i)$ is of the form:

$$K_G(t_i) = \frac{1}{\sqrt{2\pi}} e^{-\frac{t_i^2}{2}}$$

The quartic kernel is similar to the Epanechnikov but with derivatives that are continuous at the band's edge. The Biweight kernel $K_B(t_i)$ is of the form:

$$K_B(t_i) = \frac{15}{16}(1 - t_i^2)^2 \;\; |t_i| < 1$$

$$= 0 \; otherwise$$

Therefore, this kernel is more suitable for a continuous variable such as total expenditure.

Empirical evidence shows that the choice between these alternative kernels is not as critical while the choice of a bandwidth, also known as a *smoothing parameter*, is rather important. This is because the bandwidth controls the trade-off between bias and variance: large bands give smooth curves but risk bias by using observations dissimilar to the data point *x*, while small band provide a more accurate picture of the data but at the cost of introducing greater variability into the density plot.

In parametric inference, this trade-off is based on minimizing the *mean-squared error* (**MSE**) of the parameter estimates; the MSE measures the dispersion around the *true value* of the parameter being estimated. For unbiased estimators, this is equal to the OLS variance measured by dispersion around the *sample mean*, but for biased estimators there is an optimal trade-off between bias and efficiency measured by the MSE. However, the importance of this trade-off arises also from its relationship to the quadratic loss function that measures predicted error based on the difference between the parameter estimate and its true population value. This error is equal to the MSE. To see this, write the expected value the estimate $\hat{\beta}$ from its true population value as

$$\mathrm{E}(\beta - \hat{\beta})^2 = \mathrm{E}(\hat{\beta}^2) + \mathrm{E}(\beta^2) - 2\,\beta \mathrm{E}(\beta) = [\mathrm{E}(\hat{\beta}^2) - \beta^2][+\beta^2 + \mathrm{E}(\beta^2) - 2\,\beta \mathrm{E}(\beta)] =$$

$$\mathrm{Var}\,\hat{\beta} + [\mathrm{E}(\hat{\beta}) - \beta]^2 = [\mathrm{Variance} + \mathrm{Bias}^2]$$

The MSE is a positive and a weighted measure that squares the positive and negative bias to ensure a positive bias^2, and it gives equal weight given to the sum of variance and squared bias. However, unlike the parametric inference, the aim of the nonparametric application is not to estimate a parameter, but a function with *mean squared errors for each point* of the estimated density. Therefore, it is natural to minimize the expectation of the integral of the squared error over the whole density, called the ***mean integrated squared error*** **(MISE)**, leading to:

$$MISE(f^e) = \int \Big(E\big(f^e(x)\big) - f(x)\Big)^2 dx + \int var\big(f^e(x)\big)\, dx \qquad (11.1.3)$$

which is the integrated squared bias plus the integrated variance. It makes sense to minimize ***MISE*** by choosing ***h*** and a weighting kernel function. Rewriting ***K* = K/h*** , we note that:

$$E(f^e) = \frac{1}{n}\sum_{i=1}^{n} E\big(K^*(t_i)\big) = \int K^*(t) f(x) dx \qquad (11.1.4)$$

which, for a given $\boldsymbol{f}$, does not depend upon n but only on $\boldsymbol{K}$ and $\boldsymbol{h}$. This suggests increasing the sample size alone will not reduce the bias. The choices of $\boldsymbol{h}$ and $\boldsymbol{K}$ are important. Silverman (1986) obtained an approximate optimal bandwidth for the MISE of a kernel function. For kernels that are symmetric about zero with continuous derivatives at all orders with a variance v_k, it can be shown that the approximate optimal $\boldsymbol{h}$ is equal to:

$$h^{*}=v_{K}^{-\frac{2}{5}}\left(\int K(t)^{2}\,dt\right)^{\frac{1}{5}}\left(\int f''(x)^{2}dx\right)^{\frac{-1}{5}}n^{\frac{-1}{5}} \qquad (11.1.5)$$

The drawback of this measure is that this “optimal $\boldsymbol{h}$” depends on knowledge of the unknown density **f** ( ) we are trying to estimate. However, (11.1.5) confirms that the optimal window gets smaller as the sample size grows (last term) but it does so slowly, in (inverse proportion to the fifth root of n, and as the degree of fluctuation of the unknown function increases (penultimate term). Note the effect of the absolute size of the second derivative of the density. If there is a large amount of curvature, the band estimates based on averaging in a band will be biased, so the bandwidth should be small; on the other hand, on the approximately linear segments of the density, the bandwidth should be large.

Often an adequate procedure is to plot with different bands and then judge by eye if it is under or over smoothed; though the eye can ignore variability it judges to be artificial or fail to spot the features covered up by oversmoothing. It is therefore, helpful to have a good bandwidth to start the calculations with, and some guidance is available for doing so. Substituting the value of the optimal h back into the formula for the mean integrated squared error and minimizing with respect to k, (11.1.5) results in the Epanechnikov kernel. If both the kernel and the density are Gaussian, then (11.1.5) results in an optimal bandwidth of $1.06\boldsymbol{\sigma N}^{-1/5}$ with $\boldsymbol{\sigma}$ as the standard deviation of the density. Even better results can be obtained by replacing $\boldsymbol{\sigma}$ a robust measure of spread that gives the optimal bandwidth as

$h^{*}=1.06\ \min(\sigma, 0.75IQR)\mathrm{N}^{-1/5}$

where $\boldsymbol{IQR}$ is the difference between the 75th and 25th percentiles, the interquartile range.

For the Epaechnikov, the multiplying factor is 2.34 and with the quartic 2.42. The relative efficiencies of kernels can be shown to be .9512 for the Gaussian kernel, .9939 for the Biweight kernel and .9295 for the rectangular, suggesting that there is little to choose between kernels on efficiency grounds.

For non-parametric purposes that go beyond graphical presentation, we need a more objective method of selecting the window band. The most common among these is *cross-validation*, Silverman (1986, pp. 48-6). Cross-validation obtains different density estimates from different bandwidths, and then selects a bandwidth that minimizes error. There are also alternative non-kernel *adaptive* methods that adapt the bandwidth to the availability of data of points at each region of the density function. A kernel method has a fixed number of observations falling into each bandwidth. By contrast, the ***nearest neighborhood*** method is **adaptive**, it adapts the amount of smoothing to the 'local' density of data controlled by an integer ***k*** that is considerably smaller that the sample size *n*, and usually approximates the square root of ***n***. The number of observations in a bandwidth of the nearest neighborhood box is inversely proportional to the size of the box required to contain a given number of observations. It is therefore, larger in the tails of the density than in its main part in order to reduce the problem of undersmoothing in the tails with sparse observations. Since these alternatives are also employed for nonparametric regression, we examine them next.

11.2 *Nonparametric Kernel Regression*

The nonparametric approach can also overcome the limitation of parametric regression analysis that is defined as conditional expectation $E(y|x)=m(x_i)=\alpha + \beta x_i$. The latter assumes that (y_i, x_i) is a bivariate normal density. The normal distribution then justifies the parametric assumption of linear specification. However, if the true distribution is not normal, then $E(y|x)$ becomes invalid, giving rise to the misspecification problem, making the least square estimator biased and inconsistent. By assuming no functional form for the regression, *nonparametric estimators offer a solution to the misspecification problem*.

The basic idea is, for each *x* point, to *average the* y_i *corresponding to* x_i *in an interval around* *x*, and then to form $\widehat{m} = \frac{\sum_{i=1}^{n} y_i}{n}$. Given the equation error term *u*, we can write $\widehat{m}$

$$\widehat{m} = m + \frac{\sum_{i=1}^{n} u_i}{n} \tag{11.2.1}$$

Then, *E(u_i)=0* as $n \rightarrow \infty$, which makes $\hat{m}$ a consistent estimator of m. We can run a regression function by this method using the sample observation to calculate the average of all y-values corresponding to each x or a vector of x's. Of course, there are no points exactly at each x but we solve this problem by averaging over the points "near" x, where we define nearness in terms of a bandwidth that reduces to zero as the sample size increases; weighted appropriately to avoid discontinuities. This method is effectively a rectangular kernel regression, similar to time-series smoothing by a moving average over a number of adjacent points. y_i's are added accumulatively; weighted by their kernel weight, that is

$$\hat{m}(x) = \sum_{i=1}^{n} y_i \, K\left(\frac{x-x_i}{h}\right) / \sum_{i=1}^{n} K\left(\frac{x-x_i}{h}\right). \qquad (11.2.2)$$

When $h = \infty$, we have $\hat{m} = \sum y_i/n$, and when $h = 0$, $\hat{m}(x_i) = y_i$. The estimated regression function is a weighted average of the y's with the $w_i$weights given by (11.2.2):

$$\hat{m}(x) = \sum_{i=1}^{n} w_i(x)\, y_i$$

Where the weights depend on how far away each x_i is from the point at which we do the calculation.

A common method of choosing the optimal band ***h****for a kernel regression is by the *leave-one-out* ***cross-validation***.

$$CV(h^*)= \sum_{=}^{N} {}_{1}(y_i - \hat{m}_{-i}\,(x_i))^2 \pi(x_i) \qquad (11.2.3)$$

where the weights $\pi(x_i)$. $\pi(x_i)$ is introduced to downweight the end points as the local weights, which can be highly biased at such points, and

$$\hat{m}_{-i}(x_i)= \sum_{i=1}^{n} w_{ji,h}\, y_i / \sum_{i=1}^{n} w_{ji,h} \qquad (11.2.4)$$

(11.2.4) is a leave-one-out estimate of $\hat{m}_i(x_i)$ obtained from the kernel (11.2.2) formula. The procedure validates the ability to *predict the ith observation using all other observations except the ith observation,* otherwise *CV*(h*) would be trivially minimized when $\hat{m}_i(x_i)= y_i$

In practice, there will be difficulties with nonparametric regression whenever x_i does not occur frequently, namely, is close to zero; then it does not make sense to average **y** on its ***x*** any more, and the regression function becomes very imprecise in the vicinity of very small values of ***x***. For this reason, the kernel estimators with a fixed bandwidth are usually truncated at the tails, see Silverman (1986).

i. *Alternative Nonparametric Regression*

The problem of sparseness in data can be overcome by averaging observed values of y appropriately weighted when *x* is close to x_0 where the weights are w_{i0}=1/*N₀* if $x_i = x_0$and qual 0 if $x_i \neq x_0$.

There are alternative to kernel regression that estimate nonparametric functions by a locally weighted average of

$$\widehat{m}(x_0) = \sum_{i=1}^{n} w_{i0,h}\, y_i \qquad (11.2.5)$$

The weights differ with the point of evaluation x_0 and the sample value of x_i and sum up so $\sum_{i=1}^{n} w_{i0,h}$=1. Local regression uses weights that are decreasing in $|x_i - x_0|$.

One method is to order the observations by increasing $x_i$values, from that at the point *0* to points *i*, and then evaluate at $x_0 = x_i$ so

$$\widehat{m}_k(x_i)=\frac{1}{k}(y_{i-\frac{k-1}{2}}+\ldots+y_{i-\frac{k-1}{2}})$$

Then a locally weighted average estimator can be defined based on this ordering with weights as

$$w_{i0,h} = \frac{1}{k}\ 1\ (|i-0| < \frac{k-1}{2}) \qquad (11.2.6)$$

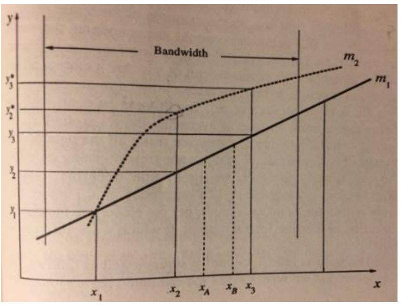
where |*i – 0*| is the distance of *i* observation from the evaluation point 0. (11.2.6) is called the ***k-nearest neighborhood estimator*** (***k-NN***) with **1** as an indicator function, it is a local regression with the equally weighted average of the *y* values for the *k* observations $x_i$closest to x_0. Define as *N*$_k$(x_0) to be the set of *k* such observations. Then

$$\widehat{m}_{k-NN}(x_0)\text{=1/}k\sum_{i=1}^{n} 1\ (x_i \in N_i(x_0))y_i \qquad (11.2.7)$$

Note how the estimator band is defined in terms of *k*, not *h*. This estimator has a uniform weight kernel but with a variable bandwidth; the bandwidth h_0 at x_0 equals the distance between x_0 and the furthest of the k nearest neighbors; more formally $h_0 \cong k/[2Nf(x_0)]$. The k/N is called the **span**.

Another problem is estimation bias arising from employing the same kernel weight function over the entire range of the density function with all data points, or with estimation bias at the band "ends" by fitting a linear function. Figure 11.1 explains the possible sources of kernel bias with a simplified three, equally spaced data points x_1, x_2, and x_3; two regression lines linear m_1 and curved m_2. Start with m_2 estimated at point x_2 with the band defined to cover x_1 and x_3 so all three contribute to the fitted conditional mean as the weighted average of y_1, and y_3* with equal weight, and y_2* with the biggest weight. If the curve is concave, the average has to be less than y_2*, so m_2 regression results in a downward bias (reverse if m_2 is convex), unless m_2 is linear. However, the bias gradually disappears as $N \to \infty$, the bandwidth becomes smaller and at the extreme alone would contribute to the fitted dependent variable. In practice, we can reduce the bias by transforming data logarithmically.

Figure11.1 *nonlinear & linear sources of bias in Kernel Regression*



Even then, linear regression by m_1 can still generate bias even if fitted with a kernel weight function. Figure 11.1 now shows the bandwidth containing five observations; with data points x_A and x_B, they are no longer equally distanced. y_2 still gets the most weight, but y-value corresponding to x_A and x_B also receive some positive weight, so the estimation is biased upward. This biased is most serious at the "ends" bandwidth points. For example, at the smallest value, x_1, the average of nearby points is all to the right of x_1. Therefore, there will be upward, or downward bias (depending on a positive or negative regression slope). The problem will become less acute as the bandwidth shrinks to include only the points that are close to the center but remains a major source of bias at the "end" points. To sum up, with the kernel function close to linearity or linear regression with a suitable kernel weighting function, these sources of estimation bias can be removed. This raises the question of combining the best of ***m₁*** and ***m₂***, namely a series of locally-weighted kernel OLS applied to each bandwidth separately instead of all data points together; that is what the **locally weighted average estimator** does.

The kernel regression estimator is a *local constant estimator* that assumes $m(x)$ is a constant in the local neighborhood of x_0. Instead, we can assume $m(x)$ is linear in the neighborhood of x_0, so $m(x)=a_0+ b_0\,(x - x_0)$ in the neighborhood of x_0. The local linear estimator minimizes

$$\sum_i^N k(\frac{x- x_0}{h})(y_i - a_0 + b_0\,(x - x_0)\,)^2 \qquad (11.2.8)$$

w.r.t $m_0=a_0 + b_0\,((x_i - x_0)$. If the estimate is at exactly x_0, then $\widehat{m}$(x)= $\hat{a}_0$. Therefore, $\hat{b}_0$ provides an estimate of the first derivative $\widehat{m}$'(x_0). The generalized local linear estimator (11.2.8) leads to a **local polynomial estimator of degree *p*** that minimizes

$$\sum_i^N k(\frac{x- x_0}{h})(y_i - a_{0,0} - b_{0,1}(x_i - x_0) - \ldots - a_{0,p}\,(\frac{(x_i- x_0)}{p!})^{\,2} \qquad (11.2.9)$$

Fan and Gijbels (1996) point out several desirable features of (11.2.9). Estimation only requires weighted least-squares regression at each x_0 point expressed as a weighted average of y_i; it has a bias term that does not depend on $\widehat{m}'$(x_0). Therefore, this estimator is particularly useful to deal with the boundary problem. This estimator can be applied with a locally, adaptive variable bandwidth, such as the *k-NN* to deal with the problem of scarcity of data at the tails of the distribution.

Alternative weights to the symmetric *k-NN* result in improved estimates of $m(x)$. A standard version of the (11.2.9) estimator is the locally weighted scatterplot smoothing or ***Lowess*** estimator, Cleveland (1979). *Lowess* regression leads to a much smoother estimates and more precise estimation at the boundaries. This estimator has a variable bandwidth $h_{0,k}$ determined by the distance from x_0 to the nearest neighbor and reduces the weights on observations with large residuals [$y_i - \widehat{m}(x_i)$]. The variable bandwidth of *Lowess* makes it robust against outliers, and its use of the polynomial estimator minimizes boundary problems but the method is computationally intensive.

The **cubic smoothing Spline estimator $\hat{m}_\lambda(x)$** minimizes the penalized residual sum of squares

$$\text{PRSS }(\lambda)= \sum_{=\,1}^{N}(y_i - \; m\,(x_i))^2+\lambda\int(m''(x))^2dx \qquad (11.2.10)$$

where λ is a smoothing parameter; the first term produces a very crude fit because then $\hat{m}_\lambda(x)= y_i$, but the second term penalizes roughness. Cross-validation is used to determine λ as a large λ results in a smoother plot.

However, the main attraction of nonparametric regression is its distribution-free method, allowing the data itself to choose the parameter estimates and the shape of the curve that is best suited to the sample at hand. But there is a cost to this advantage; the major shortcoming of non-parametric regression is its inability handle analysis involving more than a few dimensions; indeed, as the number of variables increase beyond two or three, the sample size required to minimize the ***MISE*** increases phenomenally, Silverman (1986, page 94, table 4.2) shows that required sample size jumps from 67 for three variables to close to a million with ten variables. This is called the ***curse of dimensionality*** of nonparametric analysis. It is a serious limitation for the type of regression required for multiple variables, which typically include more, sometimes many more, than a few variables. The development of semi-parametric econometrics is a response to this limitation.

11.3 *Semiparametric Regression and Tests.*

Semi-parametric econometrics combines the advantages of a nonparametric approach with the abilities of multivariate regression by retaining the linear parametric structure but applying non-parametric methods to a small subset of the principal variables, usually just one. For example, in the analysis of household expenditure surveys, the typical nonparametric component of a multivariate regression is its lead variable, namely total expenditure. A well-known semiparametric regression is Robinson (1988) *partially linear semi-parametric estimator*. The Robinson approach obtains the expected value of the dependent variable regressed nonparametrically on the lead variable with unknown functional form, and, in addition, regresses each of the parametric variables on that lead variable, again nonparametrically, to obtain similar expected values for each. Therefore, it performs as many non-parametric regressions as the number of independent variables in the equation, plus an additional one with the dependent variable. The effect of nonparametric components in the equation at hand are then controlled by subtracting the expected values from each of the corresponding variables. The new equation, with mean-differenced variables thus defined is then estimated by OLS without the bias resulting from the effect of functional form misspecification. Take as an example the budget share equation with the flexible Working-Leser functional form specification

$$y=\alpha+ \gamma.ln\ x +\beta.z +\varepsilon.$$

Its semi-parametric version is written as

$$y= F(ln\ x) +\beta.z +\varepsilon \qquad (11.3.1)$$

where let us say *x* stands for total expenditure and *z* is a vector of all other variables, for example, demographics, regions, *etc.*; and *F* is the unknown function for *x*. In the first step, the conditional means of *E(y|ln x)*, and each *E(z|ln x)* are estimated *non-parametrically*. In the second stage, *E(y|ln x)*, and each *E(z|ln x)* are subtracted from both sides of the equation, giving

$$y - E(y|ln\ x) = \{ z- E(z|ln\ x) \}.\beta +\varepsilon. \qquad (11.3.2)$$

(11.3.2) is the Robinson estimator, with the left-hand, a mean-differenced-variable, regressed on a *vector* of differenced *z* variables; differencing removes the effects of the unknown function form for the lead variables, thereby correcting the misspecification effect due to functional form, in all the model's variables. The OLS can now be applied to this equation to obtain an unbiased parameter estimate. The OLS estimator of β is $\sqrt{N}$ consistent (converges asymptotically to the population mean at a quicker rate of $1/\sqrt{N}$ than the rate of increase is as the sample size $N \rightarrow\infty$), and asymptotically normal with

$$\sqrt{N}\,(\hat{\beta}_{PL} - \beta) \rightarrow^{d} \mathcal{N}\,[0, (plim\tfrac{1}{N}\textstyle\sum_{i=1}^{N}(x_i - E[x_i|z_i\,])(x_i - E[x_i|z_i\,])^{/}\,)^{-1}] \qquad (11.3.3)$$

A variety of nonparametric estimator can be employed for (11.3.3) and the estimator can be trimmed; Robinson used kernel estimates that require convergence at no slower than $N^{-1/4}$. Therefore, oversmoothing or higher order kernels are necessary when the dimension of z is large.

For applications of Robinson's estimator to expenditure share equations in Pakistan and China, see (Bhalotra and Attfield 1998 and Gong et. al 2005).

Robinson’s multiple nonparametric application can be onerous when the equation to be estimated contains a large set of variables, involving a large number of nonparametric regressions. A good alternative is provided by the differencing approach developed by Yatchew (1997, 2003). In this approach, the data on the variable with unknown functional form is sorted in increasing order so that $x_1 \leq x_2 \leq \ldots \leq x_T$. Then Yatchew suggests first differencing the data to obtain:

$$Y_t - y_{t-1} = (z_t - z_{t-1}).\beta + [F\,(x_t) - F(x_{t-1})] + (\varepsilon_t - \varepsilon_{t-1}),\ t=2,\ \ldots, T\,. \tag{11.3.4}$$

Then as sample size increases, the $(x_t - x_{t-1})$ differences shrink at a rate of $1/T$ so that $F(x_{t-1})$ tends to cancel $F\,(x_t)$ and the OLS estimator of β is consistent asymptotically. Once the effect of the variable with an unknown functional form has been removed to obtain consistent estimates for the remaining parameters, Yatchew regresses Y_t on $F(x$t) *nonparamertically*, using a kernel or an adaptive method. This method applies nonparametric regression just once compared to multiple applications of the Robinson method. However, the Yatchew method tends to produce larger standard errors. With first differencing, the Yatchew method achieves about 66.7% efficiency relative to Robinson’s estimator. Yatchew argues that this can be improved substantially through higher order differencing by estimating a generalized version of his first-differenced equation:

$$\sum_{j=o}^{m} dj\; y_{t-j} = (\,\sum_{j=o}^{m} dj\; z_{t-j}\,)\beta + \sum_{j=o}^{m} dj\; f(x_{t-j}) + \sum_{j=o}^{m} dj\; \varepsilon_{t-j} \tag{11.3.5}$$

where m is the order of differencing. To ensure differencing removes the nonparametric effect as sample size increases, Yatchew imposes $\sum_{j=o}^{m} dj = 0$ condition; he also imposes a $\sum_{j=o}^{m}(dj^2) = 1$ condition to ensure the residual of the generalized differenced equation has a constant variance σ_ε^2. Yetchew provides a table of optimal differencing weights (Yetchew page 61) that ensure the higher order differences decline toward zero. Given these, he compares the standard errors of his differencing semi-parametric estimation with those of the partial linear method of Robinson, this indicates after using tenth-order differencing, the differencing method still produces marginally larger standard errors, see (Yatchew 2003, p. 74).

Finally, we are often interested in testing if the parametric and semi-parametric equations differ from each other, expanding the former to contain a quadratic term for its lead term. Two more commonly employed methods for this type of specification test are Hardle and Mammen (1993), and Yatchew (1997, page 703). The Hardle-Mammen test compares the nonparametric and parametric regression fits based on squared deviations between them; the resulting test statistic is

$$T_n = N\sqrt{h}\ \sum_{i=1}^{N}\{(\hat{f}(x_i) - \hat{f}(x_i,\ \beta)\}^2\,\pi\,(.) \qquad (11.3.6)$$

Where $(\hat{f}(x_i)$ & $\hat{f}(x_i,\ \beta)$ are the estimated nonparametric and parametric functions respectively, h is the bandwidth employed and π *(.)* is a weighting function for the squared deviations between fits. To obtain critical confidence interval bands for this test statistic, the Hardel-Mammen method relies on simulated values by *wild bootstrap*, see chapter 12, to generate a standard error density distribution function. The null hypothesis is that the two models are not different, and failing to reject suggests that the polynomial parametric model is at least of a quadratic degree, for an application see Bhalotra and Attfield (1998), or Koohi-Kamali (2019).

Yatchew's specification test statistic is based on comparing two estimates of the residual variance. The first estimate, the residual variance obtained from the partially linear Robinson estimator (11.3.2) after sorting the data by *ln. x*, is

$$s_{diff}^2 = \frac{1}{2n}\sum(\Delta y_i - \hat{\beta}\Delta\, x_i)^2 \qquad (11.3.7)$$

The second residual variance is the average sum of squared residuals of the parametric model, s_{res}^2 Under the null hypothesis that parametric specification is correct, Yatchew's test statistic is

$$V = \mathrm{n}^{1/2}\,(s_{res}^2 - s_{diff}^2\,)\,/(s_{diff}^2)^2 \qquad (11.3.8)$$

Readings

For textbook discussion, see Silverman (1986) on non-parametric methods, Cameron and Trivedi (2005, chapter 9) covers semi-parametric methods. Robinson (1988) proposed the root-*N* consistent semi-parametric estimator; see Gong et. al. (2005) for an interesting application.

Chapter 11 None & Semi Parametric Exercises

Q11.1 Download use *mus02psid92m.dta*, the set contains log of earnings, *lnearns*.

a. Draw histrogram for *lnearns* with bin=40, width=.25 and start =4 set for lower limit of 1st bin

b. Plot the nonparametric density with "Epan" kernel (default, so on need to request) for *lnearns* overlaid on normal kernel density (bin)bwidth=0.2 and comment on the outcome

c. Nonparametric regression: regress *lnearns* on hours by *lowess* with default width=.8 and smoother width=1.6 to compare outcome

d. Nonparametric regression: regress *lnearns* on hours by local polynomial.

e. Plot *lowess* and *lpoly* overlaid and compare the outcomes

f. Regress *lnearns* on hours by *K-NN*, the change bwidth to 10, 100, 500 and comment on outcomes. you must first download user-written "*kernreg*" for this question.

g. Contrast the outcome changing default kercode=4(quartic)above with 3 (Epan) and 6 (Gaussain).

Q11.2 Download *hprice3.dta* containing distance local house prices and their distances from a local incinerator.

a. Fit a semiparametric regression of *lprice* on *ldist larea lland* rooms baths age, treating "*linst*" from incinerator nonparametrically, why there is no estimate for that variable?

b. Test the Robinson semiparametric against a quadratic functional form for *linst-lprice* relationship using H¨ardle and Mammen’s test; why do you need replication for this test?

Chapter 12 Bootstrap

Introduction

Applied econometrics often has no way of obtaining exact finite-sample results and relies on asymptotic theory for approximation. The bootstrap, invented by Efron (1979), provides an alternative approximation method by simulation, applied to resampling from the empirical distribution, with results that are only exact in infinitely large samples. The most common use of the bootstrap is in hypothesis testing, particularly for standard error estimation and approximation of probabilities in the tails of the distribution of interest; other bootstrap applications are to confidence interval and bias reduction estimations. The bootstrap approximations fall into two broad categories; applications for statistical inference when the usual standard error computations are difficult to obtain, and applications that further refine approximation in finite samples.

12.1 *Bootstrap without refinement*

Suppose we wish to estimate the variance of the sample mean $\hat{\mu}=\bar{y}$ of a random variable y_i *iid*[μ, σ^2], but the variance is not known. $V(\ \hat{\mu})$ can be obtained from *S* times resampling from the original sample. A bootstrap treats the actual sample data of size N, $y_1, \ldots, y_N$, as the population and obtains ***B*** random draws with **replacement** of the same size *N* from the "population" to construct B estimates of $\hat{\theta}_b = \bar{y}_b$, *b*=1, . . , *B*, and use those to estimate

$$V(\hat{\theta}) = (B\text{-}1)^{-1}\textstyle\sum_{b=1}^{B}(\ \hat{\theta}_b - \bar{\bar{\boldsymbol{\theta}}}_b)^2 \qquad (12.1.1)$$

where $\bar{\bar{\boldsymbol{\theta}}}_{\boldsymbol{b}}$ is the average of $\sum_{b=1}^{B}(\ \hat{\theta}_b)$ and standard errors computed from $\sqrt{V(\hat{\theta})}$; this is called the **bootstrap estimate of the standard error**. (12.1.1) is the major equation for bootstrap standard error estimation and provide the basis for bias-reduction of the bootstrap estimation, see section v below. The approach replicates the *dgp* of the original sample. In general, we rely on bootstrapping when standard errors are difficult to compute. For example, if a *2sls* estimator expected value of variable from a first stage, based on homoscedasticity, is used to estimate a parameter in a second stage, would be hard to calculate the standard errors because of the additional random component due to the possible first stage heteroskedasticity The estimate may be a **function of other parameters** when the coefficient estimation involves a non-linear function of two parameters $\hat{\theta} = \frac{\hat{\emptyset}}{\hat{\hat{\delta}}}$; or when a robust procedure against heteroskedasticity does not exist.

i. Algorithm

The implementation of a bootstrap follows the general algorithm below:

1-Given a sample of $y_1, \ldots\ldots, y_N$, we draw a bootstrap sample of size N by one of the bootstrap methods discussed below as $y^*_1, \ldots\ldots, y^*_N$.

2-Calculate the statistic of interest from the bootstrap sample in 1, such as $\hat{\theta}$, or $S_{\hat{\theta}}$, or

$t^*=(\hat{\theta}^* - \hat{\theta})/S_{\hat{\theta}}$.

3-Repeat steps 1 & 2 a large number of *B* times to obtain *B* replications of $\hat{\theta}_1, \ldots, \hat{\theta}_B$, or

$S_{\hat{\theta}\,1}, \ldots, S_{\hat{\theta}\,B,}$ or $t_1^*, \ldots, t_B^*$.

4-Use the replications to obtain a bootstrap version of the statistic of interest.

This procedure is similar to a Monte Carlo simulation treatment of initial sample parameter estimates as the "true" values with the actual values of the explanatory variables as the "fixed in repeated samples". However, to complete such a simulation, the errors must be drawn from a known distribution such as the normal, and this is a principal weakness of the Monte Carlo simulation. The bootstrap simulation offers a method of avoiding this problem because it does not assume a known error distribution. The simplest bootstrap is to assume that the unknown **error term** distribution can be reasonably well-approximated by a discrete distribution that gives equal weights to *each residual* of the original estimation. Given an additive *iid* error, we obtain fitted residuals $\hat{u}_1, \ldots, \hat{\mu}_N$ where $\hat{\mu}_i = y_i - g(\mathbf{x}_i, \hat{\beta})$. Step 1 treats the fitted values as a new draw of residuals $\hat{u}_1^*, \ldots, \hat{u}_N^*$, leading to a bootstrap sample of $y_i^* = g(\mathbf{x}_i, \hat{\beta}) + \hat{u}_i^*$. This is called a **residual bootstrap** and it can be employed if the error term has a distribution that does not depend on *unknown* parameters. With a reasonable sample size, most of the residuals will have small absolute values, even though each receives equal weight, so, repeated random draws produce small values more frequently than large values. Nonetheless, in some circumstances, the residual bootstrap assumption of *exchangeable errors* with equal likelihood of occurring may not be true. For example, with heteroskedastic errors, if larger error variances are correlated with larger explanatory variables, then larger errors are more likely to occur with larger values of the explanatory variable. Another bootstrap sampling method, based on random draws from the pair of $w_i = g(y_i, x_i)$ can overcome this problem.

ii. Bootstrap Methods

The bootstrap from $w_i = g(y_i, x_i)$ leads to $w_1^*, \ldots, w_N^*$obtained by sampling with replacement from the original sample $w_1, \ldots, w_N$ randomly; therefore, some of the original points may appear more than once, some none at all. This is an **empirical or nonparametric bootstrap**; it is also called a **paired or complete bootstrap** since here both y_i & x_i are resampled. The paired bootstrap has three new features. First, it implicitly pairs the true, unknown errors as a part of the dependent variable, with the original explanatory variables uncorrelated with errors. Second, it does not use estimates of the unknown parameters but implicitly employs the true parameter values and the true functional form. Third, it abandons the assumption of explanatory variable values as fixed in repeated sampling; instead, the method assumes that the values are drawn from a distribution sufficiently approximated by a discrete distribution, giving equal weight to each observed vector of explanatory variables, as opposed to equal weight to each error value given by the residual method. The paired bootstrap is the most popular and the most frequently used bootstrap method because its simplicity, applicability to a wide range of nonlinear models, and use of weak distributional assumptions.

If the conditional distribution of the data is specified for example as $y|\mathbf{x} \sim \boldsymbol{F}(\mathbf{x}, \theta_0)$ and $\hat{\theta} \rightarrow^{p} \theta_0$ is available, then step 1 becomes a bootstrap from the original $\mathbf{x_i}$ to produce random draws from $\boldsymbol{F}(\mathbf{x}, \hat{\theta})$; corresponding to fixed regressors in repeated samples. This is a **parametric bootstrap** applicable to fully parametric models. A parametric bootstrap assumes the error distribution is known and therefore, it is a bootstrap that is in fact a Monte Carlo simulation.

iii. *How many Bootstraps?*

The bootstrap can be invalid for low ***B*** replications since it relies on $N\rightarrow\infty$. A sufficiently large B varies with tolerance for bootstrap-induced simulation error, or its relative discrepancy from the ideal bootstrap with B=∞, and with the reason for the bootstrap application. Let λ be the quantity of interest, namely, a standard error critical value, $\hat{\lambda}_\infty$ represents the desired bootstrap estimate with $B=\infty$, and $\hat{\lambda}_B$ be the estimate with B bootstrap replications. Then it can be shown that

$$\sqrt{B}\,(\hat{\lambda}_B - \hat{\lambda}_\infty)/\hat{\lambda}_\infty) \rightarrow^{d} \mathcal{N}[0, \omega] \tag{12.1.2}$$

and the relative discrepancy caused by only B replications is $\delta=|\hat{\lambda}_B - \hat{\lambda}_\infty|/\hat{\lambda}_\infty$; securing the relative discrepancy requires $B \geq \omega\, z^2_{\tau/2}/\delta^2$ where z stands for the standard normal random variable for symmetric, confidence interval tests. The recommended rule of thumb for a practical application is $B=384\omega$; therefore, for discrepancy < 10% with probability of 95% with $z=1.907$, we have $z^2_{0.025}/0.1^2=384$. The main problem, however, is that ω varies in each application depending on the estimation purposes. For standard error estimation, $\omega=(2+\gamma_4)/4$ where γ_4 is the coefficient of excess kurtosis for the bootstrap estimator; fatter tails of the mean's distribution distort standard error estimation. Thus, if $\gamma_4=0$, then B=384*1/2=192 replications is sufficient. But if γ_4=4, then for a symmetric two-sided test or confidence interval at 95%, larger replications are needed. For p-value=0.05 test, ω=(1-p)/p =19, therefore B=7296.

A different approach is to focus on the loss of power due to bootstrapping with finite *B* (no loss if *B*=∞). Simulations based on this approach recommend *B*=399 at α=0.05%. For hypothesis testing choose *B* so that α(*B*+1) is an integer, namely, at α=0.05%, choose *B*=399 rather than 400; with *B*=400, the 20th and 21st largest bootstrap *t*-statistic are the critical values: 399*0.05=19.95, and 400*0.05=20.5, therefore, it is unclear with B=400 which is the largest on an upper one-sided test, see MacKinnon (2002).

12.2 *Asymptotic Refinements*

Bootstrap procedures without asymptotic refinement are only exact in infinitely large samples; so their applications depend on asymptotic theory; moreover, the bootstrap approximation discussed above will offer no improvement in finite-sample performance. For that, we need the asymptotically *refinement* methods we examine now. The bootstrap methos is most commonly used to estimate standard errors when an analytical solution is hard to obtain. The second most common use of the bootstrap method is for adjustment made to hypothesis tests for type I error (probability of rejecting a true hypothesis). However, there is an *inconsistency* between the two common uses of the bootstrap method, as the tests are based on the critical values that assume normally distributed errors, or are valid asymptotically, but not in small samples. However, the main point of the bootstrap applications is to avoid results that depend on the normality assumption. This suggests bootstrap standard errors should not use the normal table critical values if this assumption is in doubt. Similarly, that restriction should be observed for confidence interval estimation, that is, not to use the bootstrap method standard error estimates with the usual critical values; instead, we should find the critical values applicable to the problem at hand. For example, if we have 1000 values of *t* statistic ordered from the smallest to the largest, a two-sided 90% test requires the 50th (1st half) and 95Th (10/2%) values; the bootstrap confidence interval is formed by subtracting the 50th and adding the 95Th values to the estimated coefficient. Since the *t*-distribution differs from the normal at the tails, the interval can be *asymmetrically* around the coefficient estimate, unlike the normal confidence interval that is symmetric. This is an example of an **asymptotic refinement** that leads to a better bootstrap method approximation in a small, finite-sample rather than in a procedure based on the conventional asymptotic theory.

Asymptotic theory relies on the result of $\sqrt{N}$ consistent estimator[11] $\sqrt{N}\,(\hat{\theta} - \theta_0) \rightarrow^{d} \mathcal{N}[0, \boldsymbol{\delta}^2]$, hence

[11] A consistent estimator converges in probability to the true distribution of the parameter being estimated as the data points (n) increase indefinitely, while a root *N* consistent estimator additionally addresses the speed of the convergence, how quickly it converges to the true value. The latter estimator is expressed as

$$\Pr[\sqrt{N}(\hat{\theta} - \theta_0)/\boldsymbol{\delta} \leq z] = \varphi(z) + R_1 \qquad (12.2.1)$$

where φ(*z*) is the standard normal *cdf* and R_1 is a residual term that approaches zero as N→∞, a result based on the application of CLT, see Appendix. In particular, the *Edgeworth expansion* provides a better approximation method by including one additional term in the expansion as

$$\Pr[\sqrt{N}(\hat{\theta} - \theta_0)/\boldsymbol{\delta} \leq z] = \varphi(z) + \frac{g_{1(z)}\,\Phi(z)}{\sqrt{N}} + R_2 \qquad (12.2.2)$$

Where $g_1(.)$ is a bounded function, and R_2 a residual term that disappears as *N*→∞. Since R_1 is of the *order of magnitude* $R_1=O(N^{-1/2})$, and, since it is divided by $\sqrt{N}$, and $R_2= O(N^{-1})$, then asymptotically $R_2< R_1$, produces a better approximation as *N*→∞[12]. A bootstrap method replication with asymptotic refinement provides an empirical method of obtaining the Edgeworth expansion that is hard to do duplicate analytically.

However, for asymptotic refinement to occur, the bootstrapped statistic must be an **asymptotically pivotal statistic**, meaning, that it must have a limit distribution independent of unknown parameters. For example, sampling from $y_i \sim [\mu, \sigma^2]$ to estimate $\hat{\mu}=\bar{y} \sim^a \mathrm{N}[\mu, \sigma^2]$ is not asymptotically pivotal since its distribution depends on the unknown σ^2, but the studentized statistic $t= \widehat{(\mu} - \mu_0)/S_{\hat{\mu}} \sim^a \mathrm{N}[0, 1]$is asymptotically pivotal; other examples are chi-squared, Wald, etc.

ii. Hypothesis Testing

We can employ upper one-sided, two-side, and other ~~t~~ests based on bootstrapping when the statistic of interest is difficult to obtain analytically; instead, we employ both asymptotically refined and without refinement bootstrap methods.

. The usual statistic T_N provides the potential for asymptotic refinement since its asymptotic normal distribution is independent of unknown parameters. The empirical distribution of $t^*_1, \ldots, t^*_B$, obtained from *B* resampling and ordered from smallest to largest, can be used to approximate the distribution of T_N . For an upper one-sided test, for example at α=0.05% with *B*=999, the **bootstrap critical value** is the 950th largest value of t^* since then (*B*+1)(1- α)=950 (when t^* is ordered from low to high). A **bootstrap *p*-value,** as the largest value at which we still fail to reject the null, can also be computed from the replicated values. For example, if the original *t* statistic lies between the 914th and 915th largest values of 999 bootstrap replicates, then an upper one-sided test if *z*=(1-914)/(999+1)=- 0.913 corresponding to critical *p* =0.086 (at

being bounded in probability in big *O* notation (see below) by $O_p(1)$, or in terms of variance of T_n as *O*(1/*n*).

[12] In a finite sample it is possible that $R_2>R_1$.

.05% for α/2). For a two-sided **nonsymmetrical test**, the bootstrap critical values are the lowest α/2 and upper α/2 quintiles of the *ordered* t^* *; the null is rejected if the original t-statistic is outside this range.* For a symmetrical test, the bootstrap critical value is the upper α quantile of the order $|t^*|$**;** rejection is if $|t|>|t^*|$. These methods use the **percentile-*t* method** for asymptotic refinement, and have the advantage of not requiring computation of $S_{\widehat{\theta}}$.

iii. Tests Without Refinement

These tests compute $t = (\widehat{\theta} - \theta_0)/S_{\widehat{\theta},\text{boot}}$ and compare this test statistic to critical values from the *standard normal distribution*. For a two-sided test, find the lower α/2 and upper α/2, and the null is rejected if $\theta_0$falls outside this range. This is called the **percentile method**.

iv. Confidence Interval

The *percentile-t method* 100(1- α) percent confidence interval is $(\widehat{\theta} - t^*_{[1-\alpha/2]} . S_{\widehat{\theta}} , \widehat{\theta} + t^*_{[1-\alpha/2]} . S_{\widehat{\theta}})$ where $\widehat{\theta}$ and $S_{\widehat{\theta}}$ are the estimate and standard error from the original sample. An alternative is the **bias-corrected and accelerated** (**BCa**) method that can offer asymptotic approximations for a broader range of problems than the percentile-*t* method.

v. Bias Reduction

The bias is computed as the distance between the expected or population average value and the data generated parameter value $E\,[\widehat{\theta}]$-θ. Bootstrap generated parameter averages are $\bar{\bar{\theta}}_b$over the bootstraps. The bootstrap estimate of the bias is then

$$\text{Bias}\widehat{\theta} = (\bar{\bar{\theta}}_b - \widehat{\theta}) \qquad (12.2.3)$$

where $\bar{\bar{\theta}}_b$is defined in (12.1.1). If for example, $\bar{\bar{\theta}}_b$=5 and $\widehat{\theta}$=4, then there is an upward bias of 1; as a result, bias correction requires subtracting 1 from $\widehat{\theta}$. In general, the **bootstrap bias-corrected estimator** of θ is

$$\widehat{\theta}_{\text{Boot}} = \widehat{\theta} - \left(\bar{\bar{\theta}}_b - \widehat{\theta}\right) = 2\widehat{\theta} - \bar{\bar{\theta}}_b \qquad (12.2.4)$$

In practice bias correction is not used for √*N* consistent estimators since the bias is small relative to the standard error of the estimation. Bootstrap bias-correction is also employed for estimators that converge at a rate less than √*N*, in particular, in nonparametric density estimation and nonparametric regression.

Example: Consider the data generated from a two-regressor exponential function from a sample 50 observations; ML estimates are $\widehat{\beta}_1$=-2.192 (intercept) and $\widehat{\beta}_2$=0.267, s_2=1.417, and *t2*=0.188; and $\widehat{\beta}_3$=**4.664**, s_3=1.741, and t_3=2.679. Table 12.1 presents the results of implementing bootstrap method statistical inference focused on $\hat{\beta}_3$ based on the *paired bootstrap* jointly resampled from (y_i, x_{2i}, x_{3i}), with replacement *B*=999 times; for this bootstrap replication, the estimated mean and standard deviation of $\hat{\beta}_3$are 4.716 and 1.939.

Standard errors: the bootstrap estimate without refinement is 1.939 compared to the conventional asymptotic estimate of 1.741.

Testing with refinement: based on the asymptotically *t* statistic and computed from each bootstrap is

T_3^*=($\hat{\beta}_3^*$ **- 4.664**)/$s_{\hat{\beta}_3^*}$. The bootstrap critical values for a α=0.05% nonsymmetrical test are the lower and upper 2.5 percentiles of the 999 values of t_3^*; from table 1 corresponding to the 2.5th lowest=-2.183 and 2.5th highest =2.066 values. We reject the null since the original sample t_3=(4.066-0)/1.741=2.679>2.066. The critical value for a symmetric test instead using the upper 5% of | t_3^*|is 2.078, again rejecting H_0.

Testing without refinement: using the bootstrap rather than asymptotic standard errors,

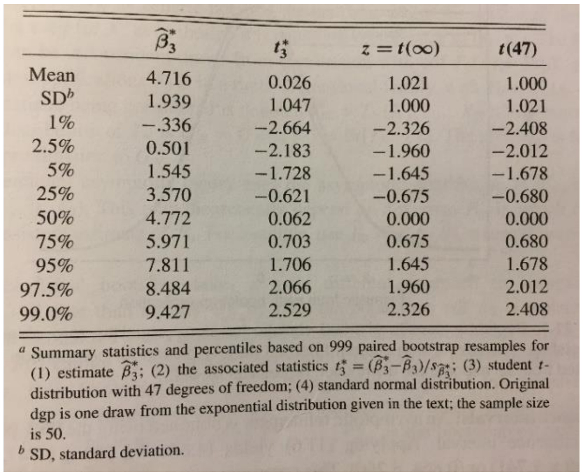
t_3=(4.066-0)/1.939=2.405, leads again to rejecting H_0 at either *standard normal or t* (with *df*=47) critical values.

Confidence Intervals: Applying an asymptotic refinement at the 95% *t* results in

(4.664 – 2.183x1.741, 4.664+2.066x1.741)=(0.864, 8.260) compared to a usual 95% asymptotic confidence interval 4.664∓1.960x1.939=(1.25, 8.08). The percentile method without refinement using the lower and upper 2.5 percentiles of the 999 bootstrap estimates leads to (0.501, 8.484) intervals.

Bias correction: The estimated bias=(4.716-4.664)=0.052, small compared to S_3=1.714; therefore, bias-corrected β_3 = 4.664-0.052=4.612.

Table 12.1 Paired Bootstrap with B=999



	$\widehat{\beta}_3^*$	t_3^*	$z = t(\infty)$	$t(47)$
Mean	4.716	0.026	1.021	1.000
SD[b]	1.939	1.047	1.000	1.021
1%	−.336	−2.664	−2.326	−2.408
2.5%	0.501	−2.183	−1.960	−2.012
5%	1.545	−1.728	−1.645	−1.678
25%	3.570	−0.621	−0.675	−0.680
50%	4.772	0.062	0.000	0.000
75%	5.971	0.703	0.675	0.680
95%	7.811	1.706	1.645	1.678
97.5%	8.484	2.066	1.960	2.012
99.0%	9.427	2.529	2.326	2.408

[a] Summary statistics and percentiles based on 999 paired bootstrap resamples for (1) estimate $\widehat{\beta}_3^*$; (2) the associated statistics $t_3^* = (\widehat{\beta}_3^* - \widehat{\beta}_3)/s_{\widehat{\beta}_3^*}$; (3) student *t*-distribution with 47 degrees of freedom; (4) standard normal distribution. Original dgp is one draw from the exponential distribution given in the text; the sample size is 50.

[b] SD, standard deviation.

Note that the bootstrap method is based in asymptotic theory and may lead to worse finite-sample approximation than that of conventional methods. To decide if the bootstrap is an improvement, a full Monte Carlo simulation is needed to obtain first 999 samples of size 50 drawn from the exponential *dgp*, then again, another bootstrap 999 times for each of these samples.

v. Extensions

A broader range of bootstrap methods beyond the $\sqrt{N}$ consistent asymptotically normal estimators are also possible.

Block Bootstrap: the **moving block bootstrap** method applies to the data that are dependent rather than independent by splitting the data into *r* non-overlapping blocks of length *l*, where $r\,l \cong N$. The method treats randomly drawn blocks as independent of each other but permits dependence inside each block. The process requires $r \to \infty$ as $N \to \infty$ to ensure consecutive blocks are uncorrelated with each other; it also requires $t \to \infty$ as $N \to \infty$.

Nested Bootstrap: The **nested bootstrap** method is a bootstrap within a bootstrap, usually employed when the bootstrap statistic is not asymptotically pivotal; it is especially useful when the standard errors are difficult to compute. Then, we can first bootstrap from the current sample to obtain $S *_{\hat{\theta},\text{boot}}$ used to form $t^* = (\hat{\theta} - \theta_0)/S *_{\hat{\theta},\text{boot}}$; then apply the percentile-*t* method to the bootstrap replications $t^*_1, \dots, t^*_B$ that provide an asymptotic refinement univariable from a single round of bootstrap. This type of **iterated bootstrap** method corrects for bias by improving bootstrap performance estimates that arise from a single bootstrap; each further iteration reduces bias by a factor of N^{-1} if the statistic is asymptotically pivotal and by a factor of $N^{-1/2}$ otherwise.

The Jackknife : The *delete-one* **Jackknife** is a method that forms *N* resamples of size (N -1) by *sequentially* deleting each observation and then estimating θ in each resample, thus it is *not* a randomly drawn resampling method. Let $\hat{\theta}$ be the original sample estimate with i=1, . . . , *N*, and the average of the *N* Jackknife estimates be $\hat{\theta} = N^{-1} \sum_{i=1}^{N} \hat{\theta}i$. The Jackknife *BC* (bias-corrected) of θ is

$$\hat{\theta}_{\text{jack}} = N\hat{\theta} - \left(N-1\right)\bar{\bar{\theta}} = (1/N \sum_{i=1}^{N} [N\,\bar{\bar{\theta}} - (N-1)\hat{\theta}i]\Big) \qquad (12.2.5)$$

The *BCa* method for a bootstrap with asymptotic refinement can also use the Jackknife. The bias appears large since it is scaled by (N -1) but note that the differences $[\hat{\theta}_{(-i)} - \hat{\hat{\theta}}]$ are much smaller than the bootstrap case because a Jackknife differs from the original sample by just one observation.

The Jackknife estimation of standard errors of $\hat{\theta}$ is obtained from

$$\widehat{se}_{\text{Jack}}[\hat{\theta}] = \{\frac{1}{N(N-1)}\sum_{\mathrm{i}=1}^{N}[N\,[\hat{\theta}_{(-i)} - \hat{\hat{\theta}}]\}^{1/2} \qquad (12.2.6)$$

The squared (12.2.6) provides the Jackknife estimate of the *VCE* that is now largely replaced by the bootstrap. The Jackknife method requires less computation than the bootstrap in small samples but is computationally intensive if *N* is large.

vi. *Applications*

Wild Bootstrap for Heteroskedasticity: White's heteroskedasticity-consistent covariance matrix performance has a poor performance in small samples; the bootstrap method can improve that. However, the residual bootstrap method assumes homoskedasticity, while the paired bootstrap method does not offer an asymptotic refinement. The **Wild bootstrap** method, see Mammon (1993), provides asymptotic refinement without imposing any structure on heteroskedasticity by replacing the OLS residual $\hat{u}_t$by the following residual:

$$\hat{u}_i^* = \frac{1-\sqrt{5}}{2}\hat{u}_i \cong \text{-0.6180}\hat{u}_i \qquad \text{with probability } \frac{1-\sqrt{5}}{2\sqrt{5}} \cong 0.7236$$

$$\hat{u}_i^* = [1\text{-}\frac{1-\sqrt{5}}{2}]\hat{u}_i \cong 1.6180\hat{u}_i \qquad \text{with probability } 1\text{-}\frac{1-\sqrt{5}}{2\sqrt{5}} \cong 0.2764$$

The evidence shows this bootstrap method works much better with heteroskedasticity compared to other bootstraps.

Panel and Cluster Data: Assume that the errors $\hat{u}_{it}$ of a linear panel regression are independent over *i*, though they may be heteroskedastic and serially corelated over *t* for given *i*. For a short panel, *T* is finite and asymptotic theory relies on *N*→∞, then consistent standard errors can be obtained from a paired bootstrap that resamples over *i* but not over *t*. This is called the **panel or block bootstrap** method, based on the assumption of a short panel, and the data independent over *i*. This bootstrap can also be applied to clustered data provided the number of clusters tends to infinity. The bootstrap methods also employed in a panel data model with *AR* long time-series dimensions, see chapter 14 on a large heterogenous panel, and Pesaran (2015, 28.11.7-8).

Appendix: *Bootstrap Asymptotic Theory*

Consider the data $X_1, \ldots, X_N$ that are independently drawn with *cdf* $F_0 = F_0\,(x, \theta_0)$ to estimate the statistic of interest $T_N = T_N(X_1, \ldots, X_N)$, with its exact infinite sample distribution given as $G_N = G_N\,(t, F_0)$. The asymptotic theory uses the asymptotic distribution of T_N, $G_\infty = G_\infty\,(t, F_0)$. When $G_N = G_N\,(.\,, F_0)$ cannot be determined analytically, bootstrap approximation of it is used by replacing the population *cdf* F_0 with a consistent estimator F_N obtained from the empirical distribution of the sample. One bootstrap re-sample renders the statistic T^*_N and its repetition $\boldsymbol{B}$ independent times leads to replications $T^*_{N,\,b}$; the empirical *cdf* of $T^*_{N,\,b}$ leads to the bootstrap estimate of the distribution of T, namely, the proportion of the bootstrap resample for the realized $T^*_N \leq t$:

$$\hat{G}_N(t,\, F_N) = \frac{1}{B}\sum_{b=1}^{B}\mathbf{1}(T^*_{N,b} \leq t)$$

where 1(.) is an indicator equal to 1 if the event occurs, zero otherwise; consistency of the bootstrap estimate requires that $G_N\,(t,\, F_N) \rightarrow^{p} G_N\,(t,\, F_0\,)$.

One advantage of the bootstrap is that it allows asymptotic refinement, that is, it can produce better approximation than possible with the conventional asymptotic theory. The proof for this statement uses Edgeworth expansions.

Consider X_i random variables standardized as $Z_N = \sum_{i=1} X_i / \sqrt{N}\ \ iid[0,\,1]$. The application of a CLT leads to *cdf* of Z_N as

$$G_N(z) = \Pr[Z_N \leq z] = \varphi(z) + O(N^{-1/2}) \qquad (12.1a)$$

where O stands for the *order of magnitude*[13] that approximates $G_N(z)$ by $\varphi(z)$ and ignores the reminder term[14].

A better approximation is possible based on the cumulants[15] of the *Edgeworth expansion* that adds two additional terms to (12.2.1a) resulting in

$$G_N(z) = \Pr[Z_N \leq z] = \varphi(z) + \frac{g_{1(z)}}{\sqrt{N}} + \frac{g_{2(z)}}{N} + O(N^{-3/2}) \qquad (12.2.2a)$$

[13] Oder of magnitude for sequences of variables (expressed in $(O,\, _o)$ notation), for a non-stochastic real number a_N is $O(g(N))$ if $lim(a_N/g(N))$ is finite nonzero, and $o(g(N))$, if $lim(a_N/g(N))$ is zero. We often put $g(N)=N^{-c}$ for some constant $c \geq 0$. That implies that $a_N = O(N^{-c})$ is of the same order of magnitude as the function N^{-c}, and $a_N = o(N^{-c})$ if it is of a smaller order of magnitude of N^{-c}. For example, $(3/N + 5/N)^2$ is $O(N^{-1})$ if for a large N; it behaves like a constant times N^{-1}, and is $o(N^{-1/2}) > o(N^{-1})$ otherwise.

[14] $\hat{\theta} = \theta_0 + o_p(1)$ is consistent estimator θ_0 since the second term has zero probability. Additionally, $\hat{\theta}$ is root-N consistent for θ_0 if $\hat{\theta} = \theta_0 + O_p(N^{-1/2})$ since then $N^{-1/2}(\hat{\theta} - \theta_0) = o_p(1)$ as data points increase.

[15] The cumulants of a probability distribution are a set of quantities that determine the moments of the distribution. The first cumulant is the mean, the second cumulant is the variance, and the third cumulant is the same as the third central moment, but fourth and higher-order cumulants are not equal to central moments. Moreover, the third and higher-order cumulants of a normal distribution are zero; it is the only distribution with this property.

Where $g_{1(z)}$depends on z, φ(*z*) and the third cumulant of Z_N, and $g_{2(.)}$ is another lengthy term. An Edgeworth expansion ignores the last term in (12.2.2a) and approximates G_N (*z*, F_0) by φ(*z*)+$\frac{g_{1(z)}}{\sqrt{N}}+\frac{g_{2(z)}}{N}$; that can be used to compute critical values and *p*-values. The problem is the cumulants can be very difficult to solve analytically. However, the bootstrap method offers a numerical method for the application of the Edgeworth expansion (12.2.2a) without the need to solve for the cumulants:

$$G_N(t, F_N)=\varphi(z)+\frac{g_{1(t,F_N)}}{\sqrt{N}}+\frac{g_{2(t,F_N)}}{N}+O(N^{-3/2}) \qquad (12.2.3a)$$

Then the bootstrap estimator of $G_N(t, F_N)$ is used to approximate the finite-sample cdf of $G_N(t, F_0)$. Subtracting the latter from (12.2.3a) leads to

$$G_N(t, F_N) - G_N(t, F_0)= G_\infty(t, F_N)- G_\infty(t, F_0)+\frac{g_{1(t,F_N)}- g_{2(t,F_0)}}{\sqrt{N}} + O(N^{-1}) \qquad (12.2.4a)$$

Assuming F_N is √*N* consistent for the true *cdf* F_0, then $F_N - F_0= O(N^{-1/2})$, (see the note and the text for (12.2.2) above); therefore, both $G_N(t, F_N) - G_N(t, F_0)$ and $G_\infty(t, F_N)- G_\infty(t, F_0)$ are equal to $O(N^{-1/2})$, so the bootstrap approximation is asymptotically no closer to $G_N(t, F_0)$ than the standard asymptotic approximation $G_\infty(t, F_0)$. However, suppose the statistic of interest T_N is *asymptotically pivotal* as defined in the text, so G_∞ does not depend on unknown parameters, then standardized T_N has the standard normal limit distribution. Then $G_\infty(t, F_N)=G_\infty(t, F_0)$; (12.2.4a) simplifies to

$$G_N(t, F_N) - G_N(t, F_0)=N^{-1/2}[\, g_{1(t,F_N)} - g_{2(t,F_0)}] + O(N^{-1}) \qquad (12.2.5a)$$

Since $F_N - F_0=O(N^{-1/2})$, we also have $[\, g_{1(t,F_N)} - g_{2(t,F_0)}]=O(N^{-1/2})$; therefore,

$$G_N(t, F_N)= G_N(t, F_0) + O(N^{-1}) \qquad (12.2.6a)$$

Which is an improvement over the conventional approximation $G_N(t, F_N)= G_\infty(t, F_0) + O(N^{-1/2})$ as long as T_N is asymptotically pivotal.

Readings

For textbook discussion, see Cameron and Trivedi (2005, chapter 11); Efron and Tibshirani (1993) for an introduction to the bootstrap method. Efron (1979) invented the bootstrap method.

Chapter 12 Bootstrap Exercises

Q12.1 Download *bootdata.dta*, this is a data set for annual number of doctor's visits.

a. Use *bsample* command for simulation to write a program for one *N* re-sample that estimates a Poisson regression of *docvis* as a function of *chronic* with robust standard errors

b. Now simulate **a.** B=400 times

c. Use 400 bootstrap values of $\boldsymbol{t_j^*}$, $\boldsymbol{j}$=1, …,400 to estimate the p-value statistic.

d. Employ the nonprametric (unconditional)bootstrap pairs method, using bsample, to estimate standard error with B=400

e. Employ parametric (conditional) bootstrap, essentially, a Monte Carlo simulation, to re-run step ***d.*** Use *nbreg* to deal with overdispersion estimation; employing *rgamma* for the gamma function

f. Check the above outcome in **e.** using *bootparametric* command, bootstrap B=400 times.

g. Now employ the residual bootstrap method; note that the bootstrap is from the original sample residuals (assumed to be iid), not the regressors

h. Apply the wild bootstrap method to account for heteroskedasticity

i. Apply the jackknife method as an alternative to the bootstrap method.

Chapter 13 Duration Models analysis

Introduction

Some economic models require an analysis of a duration of time collapsing before a certain event comes into effect, for example, a period of unemployment before entering, or re-entering, into employment, or a period of treatment before gaining full health. This chapter examines a range of duration models for different research purposes.

13.1 *Survivor & Hazard Functions*

Denote the duration of an event by $T \geq 0$, with a distribution among the population, and denote its particular value as *t*. In biostatistics, *T* is the length of time a subject lives before extinction, for instance, the useful life of a lightbulb, or, a patient's period of treatment before leaving hospital. In economics *T* is usually the time at which a person leaves the initial economic state, for example, if the initial state is unemployment, *T* would be the time spent in that state before becoming employed, or re-employed. We define the cumulative distribution function (*cdf*) of *T* by

$$F(t)=P(T\leq t),\ t\geq 0 \tag{13.1.1}$$

We define the **survivor function** *S*(.)as the *complement* function to the *cdf*, i.e. periods of time taken to come out of the initial state, that is

$$S(t)=P[T>t]=1 - F(t) \tag{13.1.2}$$

In other words, (13.1.2) is the probability of "surviving" past time *t*, a cancer patient successfully completing the course of treatment, or workers ending the period of strike. Thus, as *F*(*t*) measures the probability that the duration is less than or equal to *t*, *S*(*t*) measures the probability that the duration *T* is greater than *t*. We assume that *T* is continuous and its *pdf* density represented by

$$f(t) = \frac{dF}{dt}(t) \tag{13.1.3}$$

Then with a change in time $t+h$ for a "small" *h*>0, the probability of exiting the initial state in the time interval [$t, t+h$): is given by

$$P(t\leq T< t+h|T \ \geq t) \tag{13.1.4}$$

For example, the probability of leaving a hospital treatment given survival up to time *t*, or entering employment given being in the state of unemployment up to time *t*.

For each ***t***, we define the **hazard function** $\lambda(t)$ as the instantaneous exit probability (instantaneous rate of exit) per unit of time; given a small increase in *t* by *h*. The hazard function provides an approximation for the conditional probability (13.1.4) by

$$P(t \leq T < t+h|T \geq t) \approx \lambda(t)h \qquad (13.1.5)$$

Examples: For unemployment, if *T* is length of time unemployed, measured in weeks, then $\lambda(20)$ is approximately the *probability of becoming employed between weeks 20 and 21 after 20 weeks of unemployment*, or *conditional* on having been unemployed through week 20. For exampole, gun crime, suppose T stands for the number of months before a U S state experiences a major gun violence event. Then $\lambda(12)$ is roughly the probability of the US state experiencing gun violence during the 13th month, conditional not having seen such an event during the previous year. This is an example of *recidivism duration* as with, for example, $\lambda(6)$ for a 6-month course of a cancer eradication treatment before the patient has to terminate the program some time during the 7th month if there is relapse.

The role of the hazard function in survival/duration analysis is to define a distribution function for *t*, and to do so we must obtain the density of *T* by driving the relationship of the hazard function density and *cdf* in a few steps.

First, we note

$$P(t \leq T < t+h|T \geq t) = \frac{F(t+h) - F(t)}{1-F(t)} \qquad (13.1.6)$$

The instantaneous transition conditional on survival to time t defines (13.1.6) for Δ_t. We have

$$\frac{P(t \leq T < t + \Delta t|T \geq t)}{\Delta t}$$

If Δt is a small change *h*, then this ratio is equal to differentiating *cdf* with respect to *dt*=*h*, for a small *h*>0, that is equal to *f*(*t*) given by (13.1.3). More formally,

$$f(t) = lim_{h\to 0}\frac{F(t+h) - F(t)}{h}. \qquad (13.1.7)$$

To obtain an approximate form for the hazard function, start from (13.1.6) as given by $\frac{f(t)}{(1-F(t))}$ and substitute the denominator by (13.1.7) for approximation. Then, we can write $\frac{f(t)}{S(t)}$ as the product of two separate terms, one of which is divided by *h* as *h* approaches zero from above, that is by:

$$\lambda(t) = \frac{f(t)}{S(t)} = lim_{h\to 0} \frac{F(t+h) - F(t)}{h} . \frac{1}{1-F(t)} \tag{13.1.8}$$

(13.1.8) is the hazard function. The numerator of (13.1.8) for a small change *h* in *t* is, by (13.1.3) *f*(*t*) or by (13.1.7), while the denominator uses (13.1.2); $\frac{f(t)}{S(t)}$ is also known aa the **hazard rate.** Next, we note the derivative of *S*(*t*) for continuous by a small change in *t* is

$$\frac{dS(t)}{dt} = \frac{d(1-F(t)]}{dt} = - \frac{dF(t)}{dt} = - f(t).$$

When change in the component of (13.1.7) that depends on small *h*=*dt*, is expressed in log term, so we have

$$\lambda(t) = - \frac{dlogS(t)}{dt} \tag{13.1.9}$$

That is, the hazard function is equal to the change in the log-survivor function. By exponentiating (13.1.9), the right-hand side becomes *S*(*t*); we further obtain

$$S(t) = e^{-\int_0^t \lambda(s)dt} \tag{13.1.10}$$

Next, we note for *t* in (0, *t*), the boundary for *S*(0)=1, namely no exit during *t*=0 for sure. By using (13.1.2) as *F*(*t*)=1-*S*(*t*), , we can integrate to obtain

$$F(t) = 1 - \exp\left[- \int_0^t \lambda(s)ds\right],\ t \geq 0 \tag{13.1.11}$$

Finally, differentiating (13.1.11) leads to the density of *T* given by

$$f(t) = \lambda(t) \exp\left[- \int_0^t \lambda(s)ds\right],\ t \geq 0 \tag{13.1.12}$$

We finally note another function related to the hazard function known as the **cumulative hazard function** and defined by the term inside the exp [.] in (13.1.12):

$$\Lambda(t) = -\int_0^t \lambda(s)ds = -ln\, S(t) \tag{13.1.13}$$

The cumulative hazard function provides an alternative to the hazard function because it can be more precisely estimated than the hazard function.

The relationship between the survivor and hazard functions allows examining how the distribution of *T* affects the probability of survival by the specification of the hazard function $\lambda(t)$. Therefore, the next question is choosing suitable functional form for the distribution of *T*. Before addressing that question, however, we must point out another complication common in duration analysis, namely, that the data are subject to different types of censoring.

13.2 *Censoring*

The discussion of survival data here is confined to **single-spell data**. That is, if the individual *re-enters* unemployment again after transition from a period of unemployment to employment status in the interval [0, a], the sample disregards that data on re-entry. Moreover, we examine here two types of duration data, *time-invariant & time-varying covariates*; the former is easier to deal with, while the latter, more complicated, is often employed with discrete time data. We first examine single-spell time-invariant data. Duration data may be right-censored from above, or left-censored from below, or interval-censored. For **right-censored** data, we observe spells from time 0 until a censoring time ***c***; some spells are *completed* by the time c, while for other spells all we know if that they will end at some time in the interval (***c*, **∞). For **left-censored** data, spells are known to end at some time in the interval (***c***, 0) but the exact time is unknown, as with the data for the classical Tobit model. **Interval-censored** data are only observed for the compomitting sleted spell length in interval form [t1, t2); ")" indicates open-ended, top side.

The right-censoring arises when **flow sampling**. This kind of sampling, individuals enter the sample sometime~~s~~ during [0, a], we record their covariates at the time of entry, and the period of time each remain in the initial state before transition. An example is a random sample of women unemployed at any time during 2020 with individual records on last jobs, number of children under five, years of education, etc. at the start of the unemployment spell. Right-censoring is a common feature of flow data because, after a certain amount of time, the sample stops tracking the subjects. The only known duration data for those still in the initial state is that the duration lasted as long as the tracking period. For example, if the sample weekly data duration was for ~~a~~ six-month period and stopped tracking unemployed women after 26 weeks, then we would have right-censoring at week 26 for those who still remained unemployed at that week. Tracking can also be based on a fixed calendar date, such as the last week of June 2020 in which case, right-censoring differs across women in the sample because they would have become unemployed any time during January-June 2020. With **Stock sampling**, by contrast, the data have also a left-censoring problem. Rather than observe a random sample of people flowing into the initial state, stock data during a specific interval [0, a] records a random sample of individuals at the initial state at time a, that is, at least some of the *starting times*, a_j, *are not observed*, therefore, are left-censored. Left-censoring causes sample-selection bias because the sample data excludes individuals with longer initial spell than the sample tracking period and it is not permissible to assume missing observations are random. However, since flow data are the most common duration data, and right-censoring can only occur with flow data and are the usual kind employed in economics, we focus on this type of duration data.

More generally, if duration is not censored, the density of $t_i=t_i^*$, given (x_i, a_i, c_i), is $f(t|\mathrm{x_i};\,\theta)$. Therefore, the probability that t_i is censored is

$$P(t_i^* \geq c_i|x_i)=1-F(c_i|\,\mathrm{x_i};\,\theta)$$

where t_i is the observed duration time for individual i, $t_i^*$is the time spent in the initial state, c_i is the censoring time, a_i in the interval [0, b] for b as the length of the interval, is the starting point of entry into the initial state, and $F(t|\mathrm{x_i};\,\theta)$ is the conditional *cdf* of t_i^* given x_i. Let d_i be a censoring indicator (d_i=1 if uncensored, d_i=0 if censored); the conditional likelihood for observation i can be written as

$$f(t|\mathrm{x_i};\,\theta)^{di}[\,1-F(c_i|\,\mathrm{x_i};\,\theta)]^{(1-di)} \qquad (13.2.1)$$

Note that the length of the interval b plays no role in the analysis because *the true duration is* assumed to be independent from a_i, therefore, make it also irrelevant to the analysis. Given a random sample of size N, the maximum likelihood estimator for θ is obtained by maximizing

$$\sum_{i=2}^{N}\{d_i\ [f(t_i|\mathrm{x_i};\ \theta)d_i)]+(1-\ d_i)\log[\,1-F(t_i|\,\mathrm{x_i};\ \theta)]\} \qquad (13.2.2)$$

If there is no censoring, d_i=1, the second term in (13.2.2) is dropped.

The survival models have typically focused on right-censoring from above, but even that leaves a variety of censoring encountered when modeling duration data. **Random censoring** is when an individual observation has completed duration T^* and censoring time c^* that are *independent* of each other, namely, individuals randomly dropping out of the study. *Type I censoring* occurs from above with a certain fixed and known censoring time. This is a special case of random sampling with $c^*=T_c$. The standard survival analysis with censoring is valid only if the censoring mechanism is **independent** or a **noninformative censoring** type. This means the parameters of the censoring distribution are not informative about the parameters of the duration distribution. Then we may treat the censoring indicator as exogenous. Given censored data (t, δ), the probability of the uncensored observations is

$$\mathrm{P}[T=t, \delta=1]= \mathrm{P}[T = t \mid \delta=1] \; x \; P[\delta=1] \; ;$$

With independent censoring, we have P[$T=t$, $\delta=1$]= P[$T=t$], and if the censoring is noninformative, P[$\delta=1$] can be dropped from the likelihood function since it contains no parameter of the duration distribution. Similarly, for censored observations

$$\mathrm{P}[T=t, \delta=0]= \mathrm{P}[T \geq t \mid \delta=0] \; x \; P[\delta=0]$$

With P[$T=t$, $\delta=0$]= P[$T \geq t$] under independent censoring, $P[\delta=0]$ is disregarded under noninformative censoring. The combination of the two reduces the density to $P[T=t]$ when $\delta=1$, and to P[$T \geq t$ whenδ=0. It is possible to allow for T and C to vary with the same regressors; what matters is that the C parameters are not informative about the T parameters. With *Type II censoring*, only the p shortest spells are completely observed. For example, a clinical vaccine trail may end after p infected patients experience deteriorating conditions. In economics, random censoring from above is the usual type of duration data, so censoring differs randomly for different individuals.

13.3 *Specification of the Hazard Function*

The shape of the hazard function is a central issue in survival analysis; there are several distributions employed for that purpose and the choice among them largely depends on the features of the data and the type of research questions addressed. However, the two main distributions most frequently employed, which also provide a benchmark for comparison with all other hazard distributions, are the **exponential duration distribution** and the **Weibull distribution**. We focus on these two distributions because they are the two most frequently employed in the econometrics of duration data. The exponential distribution is the simplest because its parameter is constant, that is

$$\lambda(t) = \lambda \text{ , for all } \quad t \geq o \qquad (13.3.1)$$

(13.3.1) makes the hazard function *memoryless*; the probability of exit in the duration period does *not depend on how much time has been spent in the initial state*. Therefore, (13.1.10) becomes

$S(t)$=$exp\left(-\int_0^t \lambda(t)dt\right) = exp(-\lambda(t))$; since the *cdf* (13.1.11) is written as $F(t)$=1$-\ \lambda(t)$.

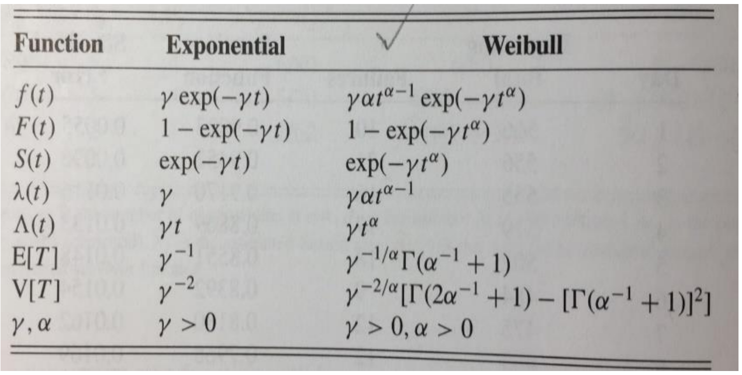
The exponential function is a one-parameter distribution, and that makes it too restrictive in many applications. A more flexible alternative based on the generalization of the exponential distribution commonly employed for duration analysis is the Weibull distribution. When the hazard function is not constant, the process displays **duration dependence**; with *positive duration dependence*, $\frac{d\lambda}{dt} > 0$, the probability of exiting the initial state increases the longer one is in the initial state, for example, long-run unemployed more likely to be employed. With a negative derivative and reverse exit probability, we have *negative duration dependence*.

If T has a Weibull distribution, its *cdf* is presented by $F(t)$=1$-\ exp(-\gamma t^{\alpha})$; its density is given by $f(t) = \gamma\alpha t^{\alpha-1}exp(-\gamma t^{\alpha})$, and its hazard function, by (13.1.10) and (13.1.12), is

$$\lambda(t) = \frac{f(t)}{S(t)} = \gamma\alpha t^{\alpha-1} \qquad (13.3.2)$$

The values of the nonnegative parameters $\boldsymbol{\gamma}$ and $\boldsymbol{\alpha}$ determine the exact shape of the Weibull distribution. The Weibull distribution reduces to the exponential distribution as its special case when $\boldsymbol{\alpha}$=1, demonstrating the Weibull as a more flexible generalization of the exponential. If $\boldsymbol{\alpha}$>1, the hazard is monotonically increasing, so it displays positive duration dependence everywhere; if $\boldsymbol{\alpha}$<1, the hazard is monotonically decreasing. If we know the hazard to be increasing or decreasing, then the Weibull distribution is suitable for capturing duration dependence. Table 13.1 shows the survivor and hazard functions for exponential and Weibull distributions.

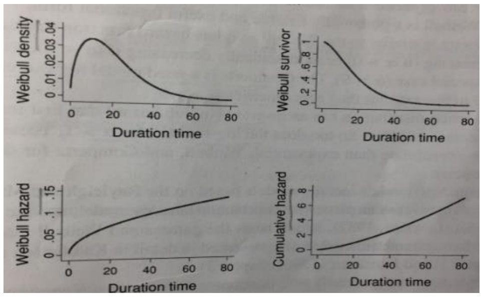
Table 13.1 *Distributional features of Exponential and Weibull functions*



Function	Exponential	Weibull
$f(t)$	$\gamma \exp(-\gamma t)$	$\gamma \alpha t^{\alpha-1} \exp(-\gamma t^{\alpha})$
$F(t)$	$1 - \exp(-\gamma t)$	$1 - \exp(-\gamma t^{\alpha})$
$S(t)$	$\exp(-\gamma t)$	$\exp(-\gamma t^{\alpha})$
$\lambda(t)$	γ	$\gamma \alpha t^{\alpha-1}$
$\Lambda(t)$	γt	γt^{α}
$\mathrm{E}[T]$	γ^{-1}	$\gamma^{-1/\alpha}\Gamma(\alpha^{-1}+1)$
$\mathrm{V}[T]$	γ^{-2}	$\gamma^{-2/\alpha}[\Gamma(2\alpha^{-1}+1) - [\Gamma(\alpha^{-1}+1)]^2]$
γ, α	$\gamma > 0$	$\gamma > 0, \alpha > 0$

As an example, Figure 13.1 shows the distributional shapes for a family of different Weibull distribution functions for the particular values of $\boldsymbol{\alpha}$=1.5 and $\boldsymbol{\gamma}$ **= 0.01**.

Figure 13.1-Weibull Distribution



13.4 *Estimation*

We consider fully parametric estimation with independent or noninformative censoring by ML or OLS. Moreover, since the duration distributions we employ are parametric, we discuss only the continuous duration estimation with the regressors assumed to be time-invariant as this is the more standard type of duration data. Then Estimation with time-variable regressors is briefly examined later. Estimation of duration models is complicated by the presence of censoring as the observed *t* is the length of a possible incomplete spell; therefore, we augment the data by the introduction of a variable indicating the presence of right-censored observations, that is, we know only that the duration exceeded *t*. The contribution of uncensored observations to the likelihood function is given as $F(t|\mathrm{x}, \theta)$. The probability distribution of *T*, in general terms with its particular distribution left unspecified, and an error term u; plus θ (q x 1) vector of parameter and x vector of regressors varying across subjects, is given by:

$$P[T > t] = \int_t^{\infty} f(\mathrm{u}|\mathrm{x},\theta)d\mathrm{u} = 1 - F(t|\mathrm{x},\theta) = S(t|\mathrm{x},\theta)$$

where S(.) is the survivor function with density for the *ith* observation given by $f(t_i|\mathrm{x_i},\theta)^{\delta_i}S(t_i|\mathrm{x_i},\theta)^{1-\,\delta_i}$ for which we have introduced into the model a right-censoring indicator δ_i*= 1 if no censoring;* δ_i*= 0 if right-censoring*. Taking logs and summing up, we have the *MLE* $\hat{\theta}$ obtained from maximization of the following log-likelihood

$$\ln \mathrm{L}\ (\theta) = \sum_{i=1}^{N}[\delta_i \ln f(t_i|\mathrm{x_i},\theta\ ) + (1-\delta_i)\ln S(t_i|\mathrm{x_i},\theta\ ) \qquad (13.4.1)$$

The first term in (13.4.1) corresponds to the completed spells, and the second term to right-censored spells, where we also assume independence over $\boldsymbol{i}$ observations. The application of (13.4.1) to duration data produces asymptotically consistent estimates if the density is correctly specified. If the density is incorrectly specified, then (13.4.1) estimation will in general be inconsistent with the exception of the exponential duration and in the absence of censoring, since the exponential function only requires correct specification of the conditional mean. With censoring, even the exponential distribution is inconsistent. Also note that many economic duration data have interval-censoring, so the data are often known to lie in an interval. For example, unemployment durations may be grouped in weeks or months, yet the parametric model applied has a continuous distribution, such as the exponential or the Weibull. In practice, it is common to assume the effect of interval-censoring is sufficiently minor to be disregarded. For example, a person unemployed for two months who becomes employed in the third month is assumed to have had an unemployed spell of exactly three months rather than a spell in the range of two to three months.

For application, we need to specify a hazard function for (13.4.1). The hazard function (13.3.2) for the Weibull distribution is $\boldsymbol{\lambda}(t) = \gamma\alpha t^{\alpha-1}$ where γ >0 and α>0. The regressors can be introduced in different ways, but the most common specification is to let $\gamma = exp(x'\beta)$ that ensures γ >0 while γ does not vary with the regressors. Then the uncensored contribution to the log LM function (see table 3.1, 1st row)

$$\ln\ f(t|x,\beta,\alpha) = \ln\,[\exp(\mathrm{x}'\beta)\,\alpha t^{\alpha-1}\exp(-\exp(\mathrm{x}'\beta)\,t^{\alpha})$$

$$= x'\beta + \ln\alpha + (\alpha-1)\ln t - \exp\,(\mathrm{x}'\beta)t^{\alpha}\ ,$$

and for the Weibull survivor function (see table 13.1, 3rd row), we have

$$\ln\ S(t|x,\beta,\alpha) = \ln\ [\exp(-\exp(\mathrm{x}'\beta)\ t^{\alpha}) = -\exp(\mathrm{x}'\beta)\ t^{\alpha}$$

With the above specification, (13.4.1) for the Weibull distribution becomes

$$\ln \mathrm{L} = \sum_{i=1}^{N}[\delta_i\{\mathrm{x}_\mathrm{i}'\beta + \ln\alpha + (\alpha-1)\ln t_i\ \exp(x_i'\beta)t_i^{\alpha}\} - (1-\delta_i)\exp(\mathrm{x}'\beta)\ t_i^{\alpha}] \quad (13.4.2)$$

The differentiation of (13.4.2) with respect to β and α leads to the first-order conditions that solve for their estimates. The interpretation of the parameter estimates of the duration models usually focuses on the hazard rate $\lambda(t) = \frac{f(t)}{S(t)} = \gamma\alpha t^{\alpha-1}$, how it changes over time, with changes in the distribution function, and with a method selecting between the two competing models. For changes in regressors

$$\frac{d\lambda(t)}{d\mathrm{x}} = \exp(\mathrm{x}'\beta)\ \alpha t^{\alpha-1}\beta = \lambda(t)\beta \quad (13.4.3)$$

showing that changes in regressors have a multiplicative effect on the hazard function. A positive β_i implies an increase in the hazard rate as a regressor in *x* increases, that is if $\beta_i > 0$, an increase in x_i leads to a rise in the hazard of failure and hence a decrease in the expected duration. Fully parametric estimation can also be obtained by least squares. In practice, this method must still have correct specification of the density and yet is less efficient than the MLE.

13.5 *Two Important parametric Duration Models*

Duration models are estimated with two formulations. Both the exponential and Weibull distributions are applicable in either of the two formulations, but some duration data must be conducted by density and hazard distributions such as the log-normal that are only applicable in one formulation. Although we cover only the two principal duration models that are most frequently employed distributions in econometrics, it is helpful to work with data using both formulations.

In a **proportional hazard** (**PH**) model, the conditional hazard rate can be analyzed as the product of two separate functions:

$$\lambda(t|\mathbf{x}) = \lambda_0(t,\alpha)\ \varphi(\mathbf{x}\beta) \quad (13.5.1)$$

where $\lambda_0(t, \alpha)$, a function of ***t*** alone, is called the **baseline hazard,** while φ(**x**β) is a function of **x** alone, and usually φ($x\beta$)=exp(**x**'β). All hazard functions of form (13.5.1) are proportional to the baseline hazard with a scaling factor φ(**x**β) that is not an explicit function of ***t***. The PH model is the most widely used in setting up a duration model regression analysis because then the model has the advantage of providing consistent estimate of regression parameter vector β without specification of the functional form for λ_0 by employing non-parametric methods. Both the exponential distribution and the Weibull distribution are PH models since their hazards are, respectively, exp(x' β) and exp(x' β)$\alpha t^{\alpha-1}$(see below for an empirical example); note that the Weibull has only one baseline hazard parameter, though there may be other Weibull models with more parameters to estimate. Finally, we note that since the baseline is scaled by the vector of covariates, if all covariates are zero, then $\lambda_0(t, \alpha)\beta_0$ where β_0 the intercept constant becomes a part of the baseline.

The other alternative duration formulation arises by modeling duration in ln *t* rather than *t*, thus the regression model specification is

$$\ln t = x'\beta + \mathrm{u} \qquad (13.5.2)$$

Therefore, this alternative formulation has an explicit stochastic error term u , with density *f* (.); the distributional form of *f* (.) determines the regression model; for example, with *f* (.) as the normal density, we obtain the log-normal hazard. This alternative formulation of duration is known as the **accelerated failure time** (**AFT**) model; different distributions for u lead to different AFT models. Exponentiating (13.5.2), we have the survival models in AFT metric with its *t* as

$$t=exp(x'\beta).v \; ; \; v = \mathrm{e}^{\mathrm{u}}$$

In the AFT metric, $\lambda(t|\mathrm{x}) = \lambda_0(v)$ φ(xβ) is the hazard rate, with $\lambda_0(v)$ as its baseline hazard that does *not* dependent on *t*. Substituting v= *t.exp*(- x'β) leads to the AFT hazard as

$$\lambda(t|\mathrm{x}) = \lambda_0(t.exp(-\mathrm{x}'\beta))\ \varphi(\mathrm{x}\beta) \qquad (13.5.3)$$

(13.5.3) is an acceleration of the baseline hazard $\lambda_0(t)$ if $exp(-\mathrm{x}'\beta) > 1$, and deceleration if $exp(-\mathrm{x}'\beta) < 1$. The Weibull and exponential hazard are the only models in being of both PH and AFT forms. Setting f (.) to the extreme-value density[16] results in the exponential and the Weibull regression models. The effect of the AFT model is to change the *time scale* by *t*=exp(x'β)e ^{u}, depending on whether this exp(x'β)>1, or exp(x'β)<1; time is either accelerated or decelerated (degraded). Then a subject with covariates x_i would have a probability of survival *S(t*) past time *t* that is evaluated at the point exp(- x_i ***β****)t*. Therefore, the AFT does not imply a positive acceleration of time with the increase of a covariate but a deceleration of time, that is, (time scale) increases in the expected waiting time for failure. Why employ a duration model based on the AFT? Because not all duration processes can be analyzed by the exponential or Weibull PH models; some, like the log-normal, can be formulated as an AFT model only, while others, like the generalized Weibull, by the PH model alone. However, the exponential and the Weibull distributions are the only two duration functions that can be applied in both the PH and the AFT formulations.

13.6 *Semi-parametric Cox function*

Fully parametric models produce inconsistent estimates if any part of the parametric model is misspecified. In such cases, the proportional hazard (PH) models are known to provide consistent estimate of regression parameter vector by non-semiparametric methods without specification of the functional form for λ_0. We employ a PH Cox semi-parametric survival function that does not require estimating the base hazard at the same time. The Cox model is semi-parametric in that the hazard rate λ_0 drops out of the estimation of β as a consequence of the PH assumption, though λ_0 estimate can be recovered, once β is estimated. Hence, the Cox model with the exponential covariant specification leaves λ_0 in (13.1.8) unspecified, now denoted as $h_o(t)$

$$\lambda(t|\mathbf{x}) = \lambda_0\ (t)\ \exp(\mathbf{x}\beta) \qquad (13.6.1)$$

The Cox proportional hazards regression model states that the survival function for the *i*th person in the data is estimated by: β, a vector of unknown parameters to be estimated from the data

[16] Extreme-value distributions are the limiting distributions for the minimum or the maximum of a very large collection of random observations from the same arbitrary distribution. For example, the normal density function with μ=0 and β=1reduces to the (minimized) distribution $e^{x}e^{-e^{x}}$.

$\lambda_0(t)$ = baseline hazard function at time t, that is hazard function when all predictors are equal to zero; x = independent vector of predictor variables.

Given (13.6.1), consider the probability of a spell of unemployment for example ending at t_i for an individual i. The survivor function is then the conditional probability of ending the spell for i divided by the conditional probability of any individual spell inding at t_i; the latter being the sum of the conditional probability i individual. Then

$$\mathrm{P}[t_i]=\frac{\lambda_i(ti\ |xi\ ,\beta)}{\sum \lambda_l(ti\ |x_l\ ,\beta)} = \frac{\emptyset(xi\ ,\beta)}{\sum \emptyset(x_l\ ,\beta)}$$

where λ drops from the last ratio beause of the model's PH assumption. This is broadly the same strategy employed by other semi-parametric methods examined in chapter 11 except that those remove the non-parametric component by differencing rather than division; and just like the models in chapter 11, the non-parametric component of the baseline hazard can be recovered once we obtain the slope estimates $\hat{\beta}$.

Note that the effect of the base hazard function is unspecified and removed semi-parametrically from the regression equation based only on the individual covariates. That may still leave unaccounted the direct recession effects on the covariates. If there are doubts regarding the presence of such direct effects on the covariates, we can also move a covariate of interest inside the hazard function, then re-estimate the model[17]separately in terms of that covariate; tha is, the covariate in question has a similar scaling effect on all the other individual corariates.

13.7 *Unobserved Heterogeneity*

We sometimes wish to test for duration conditional on both observed covariates *and* unobserved individual heterogeneity; in biostatistics, the unobserved heterogeneity is called **frailty**, for example, probability of death conditional on surviving up to *t* for a patient with a given frailty. The expansion of the above models for unobserved heterogeneity requires the following assumptions: a. the heterogeneity is *independent* of the observed covariates, starting and censoring times (strong assumption), b. the distribution of heterogeneity variable is known; c. the heterogeneity enters the hazard function *multiplicatively*. A survival model with an explicit

[17] This can be done in Stata by using the *ancillary* code for the covariate suspected of being directly affected by the base.

heterogeneity introduced into a hazard function is called a **mixture model**. A Weibull hazard function conditional on observed covariates **x**$_i$ and unobserved heterogeneity $\boldsymbol{v_i}$ is

$$\lambda(t|x_i, v_i) = v_i exp(x_i\beta)\alpha t^{\alpha-1} \qquad (13.7.1)$$

where x_{i1}≡1 for the intercept, and v_i > 0 enters (13.7.1) multiplicatively; the identification of β & α requires conditioning on normalization for the distribution of frailty v_i, usually $E(v_i)$=1. This implies the average hazard, given a vector x, is $exp(x_i\beta)\alpha t^{\alpha-1}$. Many applications adopt a **gamma-distributed heterogeneity**, v_i~ *gamma* (δ, δ), with $E(v_i)$=1 and Var=1/δ. More generally, suppose the hazard function is $\lambda(t|x_i, v_i) = v_i k(t; x_i)$ & $k(t; x_i)$>0 ; for simplicity, we disregard the dependence of $k(.\,;.)$ on parameters, and assume the density of v_i is continuous. We can then re-write (13.1.11) for the cdf with heterogeneity as

$$F(t|x_i, v_i) = 1 - \exp\left[-v_i \int_0^t k(s;\ x_i)ds\right] \qquad (13.7.2)$$

The density of v_i is $h(\mathbf{v})$= $\boldsymbol{\delta^{\delta}}$ $\mathbf{v}^{\boldsymbol{\delta}-1}$*exp*(-$\boldsymbol{\delta v}$)/$\boldsymbol{\Gamma}(\boldsymbol{\delta})$, *Var*($v_i$)=1/$\boldsymbol{\delta}$ *and* $\boldsymbol{\Gamma}$(.) is the gamma function. Let $\boldsymbol{\xi}_i \equiv \xi i(t; x_i) = \int_0^t k(s; x_i)ds$ and because the *gamma* (δ, δ+$\boldsymbol{\xi}_i$) density must integrate to unity, it can be shown its *cdf* to be

$$G(t|x_i) = 1 - [1 + \xi(t; x_i)/\boldsymbol{\delta}]^{-\boldsymbol{\delta}} \qquad (13.7.3)$$

The derivative of (13.7.3) with respect to *t*, given that $k(t; x_i)$ is the derivative of $\boldsymbol{\xi}_i$; results in the density of *t* conditional on x_i as

$$g(t|x_i) = k(t; x_i).[1 + \xi(t; x_i)/\boldsymbol{\delta}]^{-\boldsymbol{\delta}-1} \qquad (`13.7.4)$$

(13.7.4) is called **unconditional hazard** because it is not conditioned on the unobserved heterogeneity any more. When the hazard function has the Weibull distribution, $\xi i\ (t; x_i) = exp(\mathrm{x}\beta)t^{\alpha}$, the result is called the **Burr distribution** and its hazard, namely the unconditional hazard function when the conditional hazard is Weibull and heterogeneity has a *gamma* distribution, is

$$\lambda(t|x_i, v_i) = exp(\mathrm{x}\beta)\alpha t^{\alpha-1}[1 + \frac{exp(\mathrm{x}\beta)t^{\alpha}}{\delta}]^{-1}$$

Now reparametrize the Burr distribution with $\boldsymbol{\eta}$=1/ δ, known as *precision* parameter, as

$$\lambda(t|x_i, v_i) = exp(\mathrm{x}\beta)\alpha t^{\alpha-1}/[1 + \eta . exp(\mathrm{x}\beta)t^{\alpha}] \qquad (13.7.5)$$

If η=0, that is $Var(v_i)$=0, the result is the Weibull hazard but if η=1, that is $Var(v_i)$= E (v_i), then (13.7.5) can be shown to lead to the log-logistic hazard.

The main interest in a hazard model focuses on testing for the dependence of duration on heterogeneity, but a more careful look at (13.7.1) suggests that the assumption of independence of heterogeneity from observable covariates with only a single cycle data for each individual, is a strong assumption. In practice it may be hard to distinguish between duration dependence parameter $\boldsymbol{\alpha}$ and heterogeneity $\boldsymbol{\nu}$: once the *T* distribution is estimated conditional on **x**, we cannot uncover the distribution given (**x**, $\boldsymbol{\nu}$) without extra assumptions (with more than one cycle for each individual, then unobservable heterogeneity will be present in all cycles, hence allowing us to isolate the effect of flexibility parameter $\boldsymbol{\alpha}$ on the duration function). An interesting hypothesis to test in this context if H_0 : α=1; if confirmed, the test can remove the effects of flexibility and isolate that of heterogeneity on the duration function. However, when the hazard has the PH form, then it is possible to identify $\boldsymbol{k}(.)$ and the baseline hazard. Moreover, if research interest is on how covariates affect the mean duration, then modeling heterogeneity is not critical since its addition changes the error distribution but not the mean effects.

13.8 *Time-Varying regressors*

The standard duration models have regressors that change across individual observation units but not over time. However, some duration data have individual units that are observed at several stages during a spell, and the relevant regressors may take different values over the spell. For example, in a survival medical study, the dosage levels may vary over time for the same individuals, or unemployment benefits may change during an unemployment spell. Time-varying covariates pose two kinds of problems for duration model estimation. First, the misspecification that results from treating a time-varying covariate as a fixed variable. Second, a time-varying covariate may have feedback effects, for example, duration of unemployment may depend on job search but the search level may fall the longer is the duration, or the medical dosage may change in response to the patient’s improving conditions. Then the regressors may not be strictly exogenous. However, the standard duration analysis deals with only the first problem by assuming the covariates are **strictly exogenous**. To define this concept, let x(t +h) denote the covariate path from t to t+h , then strict exogeneity for the conditional hazard function at time t requires that for each t, x(t+h) remains constant for all small changes h to t. As a result, the probability distribution for the hazard function with strict exogeneity is defined as

$$P[\mathrm{x}(t, t+h)|T \geq t+h, \mathrm{x}(t)] = P[\mathrm{x}(t, t+h)|\mathrm{x}(t)]$$

That is, the probability depends not on the distribution of t, but only on x(t).

The strict exogeneity is important when duration data are discrete. The usual **grouped duration data** in economics, given in monthly or weekly, provide information only on individuals falling into the time-intervals of the data. An application of survival analysis to grouped data summarizes the information in a sequence of binary outcomes; producing in effect a panel data set where each cross-section observation is a vector of binary responses plus covariates. The advantages of this approach are: first, ease of estimating flexible hazard functions with a PH specification; second, ease of introducing observable time-varying covariates due to the sequential nature of the data. To simplify, we assume flow sampling data to exclude sample-selectivity bias with stock sampling data, and divide the time into M+1 intervals: [0, a_1), [0, a_2), . . ., [a_{M-1}, a_M), [a_M, ∞), where a_m are known constants, for example, a_1=1, a_2=2, *etc.*; and any duration falling into the last interval, [a_M, ∞), is censored, hence no observed durations are longer than a_M . Let d_m be a binary variable equal to one if the duration is censored in interval m; zero otherwise; similarly, y_m is a binary indicator equal to one if the duration ends in the mth interval; zero otherwise. Now allow individuals enter the initial state at different calendar times. Since starting time is unimportant with flow data, we assume starting times are independent of any covariate and unobserved heterogeneity. For each person i, we observe (y_{i1}, d_{i1}), . . .,(y_{iM}, d_{iM}) which is a balanced panel data set as a string of binary indicators that must be a string of zeros followed by a string of ones for any individual.

If d_i is a censoring indicator equal to one if duration i is uncensored, the log likelihood for observation i over (m-1) discrete periods is

$$\sum_{h=1}^{m_i-1} \log\,[\alpha_h(\mathrm{x_i}\,, \theta)] + di\,\log\,[1-\alpha_{m_i}(\mathrm{x_i}, \theta)] \qquad (13.8.1)$$

Summing up (13.8.1) over i=1, . . ., N leads to the log-likelihood for the entire sample. The application of (13.8.1) requires an specification for the hazard function and because of its flexibility, a **piecewise-constant proportional hazard** is popular: for m=1, . . . , M,

$$\lambda(t|\text{x},\theta) = k(\text{x}\beta)\lambda_m \ \ a_{m\text{-}1} \leq 1 < a_m \qquad (13.8.2)$$

Where $k(\text{x}\beta)$>0 and usually specified by $exp(\text{x}\beta)$, allowing different constant hazard over each time interval, though the hazard over [a_M, ∞) cannot be estimated. Hence, we have

$$\alpha_m(\text{x}_\text{i},\theta) \equiv exp\ [-exp(\text{x}\beta)\lambda_{m(\alpha_m-\alpha_{m-1})}] \qquad (13.8.3)$$

where the a_m are constant, not parameters; usually a_m=m. Therefore, the duration distribution is discontinuous at the endpoints. We can also add unobserved heterogeneity to hazards specified with grouped data, although the assumptions stated above play an even more critical role for such applications.

We note one further issue not discussed above. We confined the discussion to fully parametric regression models. It is also possible to estimate the parameters in a PH model *nonparametrically* without specifying the baseline hazard, and that provides more flexibility by avoiding arbitrary imposition of a particular function form on the data for the hazard function in advance of the investigation. Such nonparametric models are inapplicable when covariates are not strictly exogenous. In any case, this issue must rely on additional duration model distributions besides the two main distributions examined above.

13.9 *Competing Risks Model*

Competing risk is a class of survival models designed to multiple transitional exit destinations. A type of cancer treatment can have exit destinations of cleared or relapsed, see the end of chapter exercise; McCall (1996) developed a model in which workers could be reemployed either in full-time or part-time jobs, and for each outcome the considered factors were allowed to have varying effects.

We expand the above models to account for the probability survival when there are more than one exit destinations, for instance, moving from unemployment to either full-time or part-time employment. A survival function with this type of multivariant transitional probabilities will involve estimating a joint distribution of durations. The **competing risk model** (**CRM**) estimates a multiple hazards version of the single-spell model where each exit destination provides one complete duration m, and m -1 censored durations, thus competing risks determine the destination state. We denote destination-specific covariates by $\text{x}_j(j = 1,2,\ldots,m)$, and only one duration, the shortest, is observed at the end of the spell τ=*min_j* (t_j), t_j>0. If the risks are independent, then the multiple-spell survival function is

$$S_\tau(t)\text{=Pr}[\tau > t]\text{= Pr}[t_1 > t]* \text{ Pr}[t_2 > t]* \ldots *\text{Pr}[t_m > t].$$

Let $g_j(t)dt$ be the probability of risk *j* materializing over a small interval change by *dt*, then the total hazard rate of all durations is $\lambda_\tau(t)$≡-*d/dt* ln $S_\tau(t)$=$\sum_{j=1}^{m} g_j(t)$. That is, the probability of exit remains the same regardless of *j* being one of the risks or the only risk.

Given hazard functions independence, the PH Cox duration model provides probability estimates for the integrated hazard over different destinations by a PH model of the form

$$\lambda_j(t;\mathbf{x}) = \lambda_{0j}(t) \text{ exp}[(\mathbf{x}'(t)\beta_j], j\text{=1}, \ldots, m \qquad (13.9.1)$$

where both the base hazard and parameter are specific to *j*-type hazard $t_{j1} < \ldots < t_{jk_j}$;

$k_j$denotes the ordered destination of type *j*. For instance, with *m*=2, k_1=full-time work and k_2=part-time work.

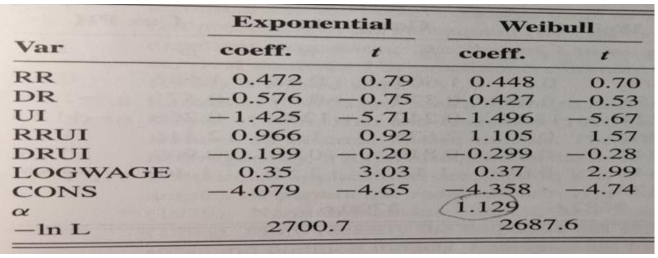
13.10 *Examples*

Example 1: Duration of Unemployment: Data from McCall (1996) Displaced Workers Supplements (DWS) for 1986, 88, 90, & 92; application requires information on part or full-time status of the first post-displacement job. Unemployment durations have been measured in two-week intervals and four binary censors are employed to indicate the status of the first post-displacement job, here CENSOR 1 is used, meaning a spell is complete if person is re-employed at a full-time job. *UI* is an indicator for the person filling an unemployment claim, *RR*=the ratio of weekly benefit to weekly earnings in the job lost, and disregard is the threshold earnable amount in a part-time job without losing unemployment benefits and its rate (*DR*)is the ratio of the amount to weekly earnings in the lost job.

Consider two parametric regression models in table 13.2 The formulation is the *PH*, though the outcome is also interpretable as an *AFT*. The Weibull model provides the best fit, with positive state dependence (α=1.129), meaning the probability of the spell terminating increases the longer the spell lasts. *UI* is negative in both models, implying that the joblessness of those who claim unemployment insurance terminates more slowly. *LOGWAGE* is positive in both models with little variation across models.

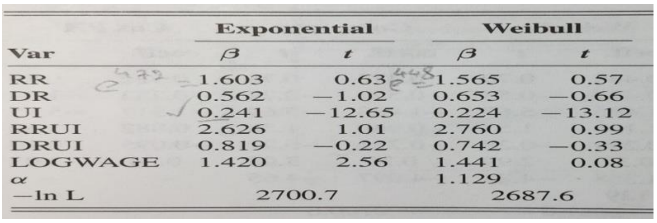
Table 13.3 shows the estimates for the exponentiated coefficients corresponding to those in table 13.2. The *UI* hazard ratio is 0.241, meaning claiming unemployment insurance decreases the hazard by nearly 76% =[(1-0.241)*100] over the baseline hazard. Similarly, for the Weibull function, the Hazard decreases by about 78%. Here the results from both models are similar, the relatively few variables that are significant, indicate large unexplained variation, possibly caused by unobservable heterogeneity.

Table 13.2 Exponential & Weibull Coefficient Estimates of Unemployment Distributions



	Exponential		Weibull	
Var	coeff.	*t*	coeff.	*t*
RR	0.472	0.79	0.448	0.70
DR	−0.576	−0.75	−0.427	−0.53
UI	−1.425	−5.71	−1.496	−5.67
RRUI	0.966	0.92	1.105	1.57
DRUI	−0.199	−0.20	−0.299	−0.28
LOGWAGE	0.35	3.03	0.37	2.99
CONS	−4.079	−4.65	−4.358	−4.74
α			1.129	
−ln L	2700.7		2687.6	

Table 13.3 Exponential & Weibull Hazard Ratios of Unemployment Distributions



	Exponential		Weibull	
Var	β	*t*	β	*t*
RR	$e^{.472}$ = 1.603	0.63	$e^{.448}$ = 1.565	0.57
DR	0.562	−1.02	0.653	−0.66
UI	✓ 0.241	−12.65	0.224	−13.12
RRUI	2.626	1.01	2.760	0.99
DRUI	0.819	−0.22	0.742	−0.33
LOGWAGE	1.420	2.56	1.441	0.08
α			1.129	
−ln L	2700.7		2687.6	

Example 2: Duration of Strikes

Kennan (1985, J of Econometrics) examined the duration of official (contract) strikes in the US manufacturing industries. The study employs functional forms for hazard and unobserved heterogeneity that fall outside those examined above; the brief discussion included here is intended to show the range of topics analyzed in labor economics with the application of survival models. Kennan specifies a logit hazard function and a beta function for unobserved heterogeneity variable conditional on a parameter α that determines its precise distribution; absence of heterogeneity is indicated by α=0. The beta-logit duration model is applied to the BLS from US department of labor to the age of strikes in days relation to monthly industrial production in order to test the prevailing view that strikes are procyclical.

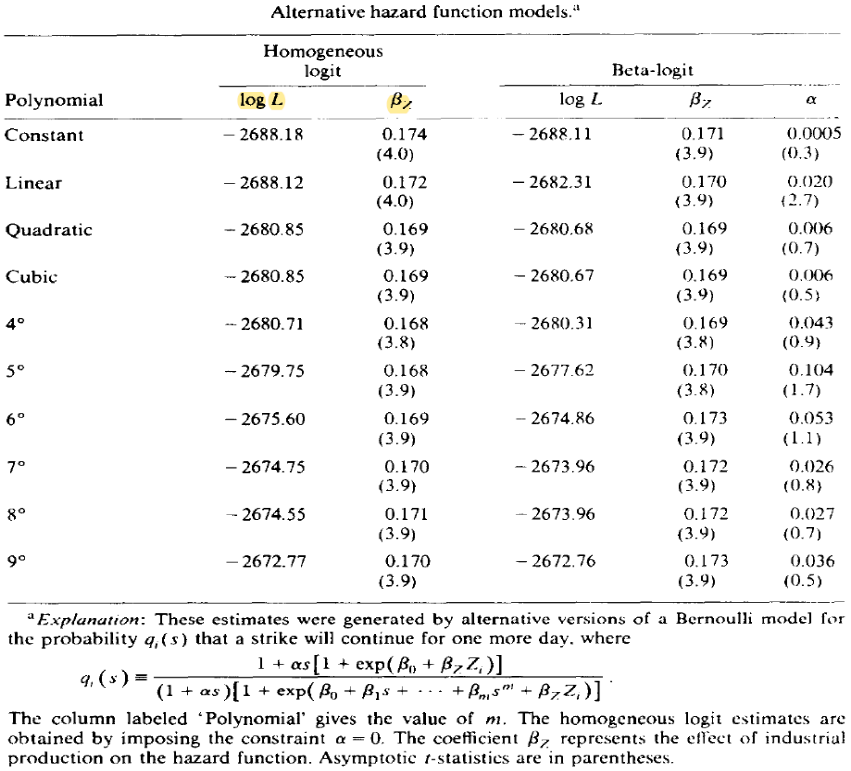
The author suggests an alternative hypothesis that duration of strikes is a function of the cost of strikes to both firms and workers, predicting that when production is near its peak, the cost of strike (to both parties) is relatively high, so the number of strikes should be reduced; the strikes, conditional on the age of strikes, should be *countercyclically* affected by business cycle fluctuations. The vector of explanatory variables for a strike ***i*** after *s* days is

$$X_i\,(s)=(1\ s\ s^2\ s^3\ .\ .\ .\ s^m\ Z_i\ )$$

where *m* is the order of the polynomial in the age of the strike*s*, namely, linear, quadratic, etc., and Z_i denotes the value of industrial production in the month when strike ***i*** began.

Table 13.4 shows the main ML estimates of the study comparing the models without heterogeneity (logit hazard, first two columns) and with heterogeneity (logit-beta hazard, third-to fifth columns); both estimated with a ninth-order polynomial of ***s***. Note that production positively affects strikes hazard function. Moreover, α is close to zero and insignificant for the logit-beta model, suggesting the simple logic model captures the main feature of the hazard function of this study. The key empirical result, however, is the consistently negative effects of the age of strikes on hazard function, that is strikes are counter-cyclically related by the business cycle fluctuations.

Table 13.4 Hazard function for duration of Strikes



Alternative hazard function models.[a]

	Homogeneous logit		Beta-logit		
Polynomial	log L	β_Z	log L	β_Z	α
Constant	−2688.18	0.174 (4.0)	−2688.11	0.171 (3.9)	0.0005 (0.3)
Linear	−2688.12	0.172 (4.0)	−2682.31	0.170 (3.9)	0.020 (2.7)
Quadratic	−2680.85	0.169 (3.9)	−2680.68	0.169 (3.9)	0.006 (0.7)
Cubic	−2680.85	0.169 (3.9)	−2680.67	0.169 (3.9)	0.006 (0.5)
4°	−2680.71	0.168 (3.8)	−2680.31	0.169 (3.8)	0.043 (0.9)
5°	−2679.75	0.168 (3.9)	−2677.62	0.170 (3.8)	0.104 (1.7)
6°	−2675.60	0.169 (3.9)	−2674.86	0.173 (3.9)	0.053 (1.1)
7°	−2674.75	0.170 (3.9)	−2673.96	0.172 (3.9)	0.026 (0.8)
8°	−2674.55	0.171 (3.9)	−2673.96	0.172 (3.9)	0.027 (0.7)
9°	−2672.77	0.170 (3.9)	−2672.76	0.173 (3.9)	0.036 (0.5)

[a] *Explanation*: These estimates were generated by alternative versions of a Bernoulli model for the probability $q_i(s)$ that a strike will continue for one more day, where

$$q_i(s) = \frac{1 + \alpha s[1 + \exp(\beta_0 + \beta_Z Z_i)]}{(1 + \alpha s)[1 + \exp(\beta_0 + \beta_1 s + \cdots + \beta_m s^m + \beta_Z Z_i)]}.$$

The column labeled 'Polynomial' gives the value of m. The homogeneous logit estimates are obtained by imposing the constraint $\alpha = 0$. The coefficient β_Z represents the effect of industrial production on the hazard function. Asymptotic t-statistics are in parentheses.

Readings

For textbook discussion, see Cameron and Trivedi (2005, chapters 17, 18 and 19), Wooldridge (2010, chapter 22). McCall (1996) and Kenan (1985) are well-known applications.

Chapter 13 Duration Models Exercises

Q13.1 Downloads *kva.dta*, the data are about the ability of a new type of generators to stand overload

a. Fit a Weibull model with exponential Coefficients (Hazard ratios) in PH metric, and Obtain un-exponentiated coefficients, and also fit the model in corresponding AFT metric; and use *p* to convert previous un-exponentiated coefficients to AFT metric

Q13.2 Download *mfail3.dta*, more complicated patient survival data that have repeated occurrences

a. describe data, fit Weibull, interpret estimates, fit exponential, compare models; why use robust standard errors; are the standard errors noninformative?

. Q13.3 Download *cancer.dta*, patient survival data in drug trial for model selection, use LR or Wald test if models are nested; if not use Akiake with the smallest AIC

a. Fit Weibull, test if it differs from exponential, fit exponential for comparison

Q13.4 Download *hip3.dta* data on hips fracture data set.

a. Employ *streg* commands to allow explanatory variables to have hazard effects that depend on the model's research question. The research question is based on the hazard (13.3.2) scaled by sex & protective device but hazard for both sexes have different shapes using (13.4.1), that is age and *protect* remain the same for both sexes but differently scaled for each sex (*male*).

Q13.5 Download *cancer.dta.*

a. Fit the model in Q13.3 with the coefficients constrained to be same across strata while allow intercept and ancillary to vary.

Q13.6 Download *cancer.dta.*

a. Sort the data by individual *id*, censor time and transition time, then estimate a Discrete Survival model with Unobserved Heterogeneity/Frailty by *glm*, defining the duration sequences in log terms.

b. Show the same results are obtained by the discrete Weibull hazard, using the *pgmhaz* command function.

c. Compare the results in ***b.*** with those obtained from the Weibull function without unobserved heterogeneity, and those with the continuous Weibull. How different are the results?

Q13.7 Download *hypoxia.dta*, the data set for 109 cervical cancer patients followed over 1994-2000 by the time (in years) after treatment loss or relapse, the latter further divided into *local* if condition relapse was in pelvis, or *distant* if elsewhere.

a. Fit a Competing Risks model for *ifp* as a function of *tumsizxer* & *pelnodde*, with *local* as the event of interest and distant as competing event. Obtain the coefficient estimates for this model.

Chapter 14 Long Panel Cointegration Test and *VEC* Estimation

Introduction

This chapter discusses two issues related to testing and estimating of cointegrated *long* panel data sets, both problems arise when we extend time-series analysis to a panel data context with a long time dimension. The long time series are important when we test for slow converging series that are in fact cointegrated but the week testing poor of the Dicky-Fuller or Johansen cannot correctly identify cointegrated stationarity, as discussed in chapter 8. Moreover, for panels of short time span, much of variation across observations are from time-invariate data units; therefore, by necessity we assumed slope homogengeity across time periods in static and dynamic pael data analysis of chapters 4 and 5. That assumption can no longer be plausibly maintained if panel time spane is long, and the problem of slope heterogeneity over time must be addressed to avoid inconsistent estimation. We examine Im, Pesaran and Smith (2003), the IPS unit-root cointegration test as a solution to the weak power of the DF and Johanen cointegration tests by exploiting additional variation available when combining cross-section, countries for example, and time-series into a panel data set. For cointegration we must employ a more general multiple series testing procedure that can capture potential co-movement among panel data variables. We examine such a test in section 14.1 specifically formulated to relax I(0) verses I(1) of the DF and Johansen tests and allows for cointegrated models that contain I(0) *and* I(1) in single-equation multiple panel series. Moreover, with long panel data sets, typically with $T > N$, we can estimate a separate equation for N units in each time period, but then the standard panel data estimators ignore the new estimation problem of slope heterogeneity. We can of course allow slope to change for each time period, but then estimation of a very large number of parameters would seriously lower the degrees of freedom and hence the esimatin accuracy. We discuss the more recent Mean Group and Pooled Mean Group estimators in section 14.2 that can estimate cointegrated ARDL models by imposing plausible parameter restrictions and yet allow and test for heterogeneity hypothesis.

14.1 *Unit Root Bounds Test for Cointegration*

ARDL provides the basis of a more powerful unit-root test by exploiting the greater variation of co-variables offered by the panel data structure in order to overcome the weak power of the residual-based and the Johansen test procedures. There are several problems with the Engel Granger residual-based approaches to test of cointegration stationarity. First, the test results can change depending on the left-hand variable selected in the first step estimation of error correction. Second, it does not allow for more than one cointegrated relation. Furthermore, the test has low power. The co-integration stationary test developed by Johansen addresses the first two problems. However, low test power remains a problem. Since these tests apply to a *I*(1) spurious null hypothesis, they are based on the assumption that underlying regressors are *I*(1). With a low co-integration test power, the true order of integration may well be unknown to the investigator, despite rejection of stationarity by the DF test, for eample, with very slow converging series. Lack of certainty regarding I(1) regressors in the model adds uncertainty as a new problem to the standard unit root tests. The Pesaran, Shin and Smith (2001) **Bounds test** addresses this problem and obtains co-integration test results on whether the model regressors are I(0) or I(1) when the the order of cointegration is unknown. The test employs an ADRL model that involves a number of steps discussed below.

Testing for cointegrated unit roots among several series would be clear if we knew: a) I(0), then apply OLS to the series in levels, b) all series are I(1), then apply the *OLS* to the VAR series defined in first differences, and c) all the series are integrated of the *same order*, and also cointegrated; apply *ECM* by the *OLS* if the test employed is based on the residual-based EC. Th~~a~~t leaves the possibility that only *some* of the variables may be *I*(0) , and *some I*(1), even integrated but not in integers but only *fractionally*; in addition to other I(1) series. Based on an *ADRL* model, the bounds test is designed to provide an answer when it is unknown, whether the series are of the same order of integration and produces a test outcome that is robust to the possibility that the series may be cointegrated with different orders of integration of I (0) and I(1). This method has some advantages over the standard methods of cointegration unit root tests. First, it can be applied to a combination of I(0) and I(1) series, second, it is conducted with just a single, equation-by-equation application, so it is simple to apply; finally, it allows *different lag structures* for different series because it generates asymmetries by weak exogeneity.

The bounds test is developed from *ARDL* model and involves several steps. Start from the general *ADRL* (*p, q*) model with a serially independent error term:

$$Y_t=\beta_0+ \beta_1 y_{t-1}+\ldots+ \beta_p y_{t-p}+\alpha_o x_t+\alpha_1 x_{t-1}+\ldots+\alpha_q x_{t-q}+\varepsilon_t$$

Summing over 1 to *p* for y_t series and over 0 to *q* for x_t series. Assume we test for stationarity among three time series of y_t, X_{1t}, *and* X_{2t}.

Step 1: Estimate the error correction model for the three series:

The conventional *ECM* would be

$$\Delta Y_t=\beta_0+\beta_1 y_{t-}+\gamma_1\Delta x_{1t-1}+\gamma_2\Delta x_{2t-1}+\varphi z_{t-}+\varepsilon_t$$

with the cointegrated long-run series z_{t-1} are obtained from a first step regression

$$Y_t=\alpha_o+\alpha_1 x_t+\alpha_2 x_{t-}+\varepsilon_t$$

Estimating the *ECM* in a one-step, single-equation employing the *ARDL* model would result in

$$\Delta Y_t=\beta_0+\sum_{i=1}^{p}\beta_1 y_{t-1}+\sum_{0=1}^{q}\gamma_1 x_{t-1}+\sum_{0=1}^{q}\gamma_2 x_{t-2}\,x+(\varphi_0 y_{t-1}+\varphi_1 x_{1t-1}+\varphi_2 x_{2t-1})+\varepsilon_t \quad (14.1.1)$$

(14.1.1) is similar to the conventional *ECM*; the terms inside the brackets represent the EC term. However, the difference is that now the coefficients on the lagged structure of the EC term e_{t-1} (equal to the expression inside the brackets) in eq. (14.1.1) have no *a priori* restrictions because the new model is estimated equation-by-equation; ~~hence~~ so, the coefficients of the *EC* term are *not* restricted across equations

Step 2: Compute the F or Wald statistics for testing the null hypothesis

$$H_0\colon \varphi_0=\varphi_1=\varphi_2=0$$

Rejection of H_0 suggests a long-run relationship between y, x_1 and x_2. The exact critical values are not available for a different mix of I(0) and I(1) and the distribution of this test statistic is non-standard. However, the critical values obtainable from Pesaran, Shin and Smith (2001) provide bounds on the critical values for the asymptotic distribution of the F-statistic; the lower bound F_L is based on the assumption that all variables are *I*(0), and the upper bound F_U on the assumption that all variables are *I*(1); and the critical bound values depend on whether a trend is included or excluded in (1)

Step 3: Compare the computed statistic from step 2 with critical F_L and F_U.

-If $F_b > F_U$, we reject *Ho*, and conclude that there is a potential long-run relationship between y_t, x_{1t} *and* x_{2t}.

-If $F_b < F_L$, we fail to reject Ho and conclude that there is no long-run relationship between y_t, x_{1t} *and* x_{2t}.

-If $F_L > F_b > F_U$, we conclude that the test is inconclusive, somewhat similar to the Durbin-Watson test.

Using this method there is no need to know *a priori* the order of integration of the underlying variables when the computed F-statistic is above or below the critical bounds to conduct a co-integration test. As a cross-check, we should also perform the bounds *t*-tests on the coefficient of the lagged dependent variable, testing *Ho*: φ_0=0 against H_A: $\varphi_0 < 0$; such *t*-tests are also non-standard though Pesaran, Shin and Smith (2001) supply the critical bounds values. If the t-statistic is larger than the tabulated "I(1) bound", we conclude the existence of a long-run relation between the variables; if smaller than the "I(0) bound" variables are all non-stationary. The test is not valid in presence of I(2) series, hence the implementation of the bounds test requires to ascertain first that there are no I(2) series among the variables.

14.2 *Slope heterogeneity and Mean-Group Cointegration Test*

We examined the asymptotic of large *N* and small *T* with the limited information on time-varying observation. That limitation makes the assumption of homogenous slopes for cross-sectional units necessary for both fixed and random effects models. When both *N* and *T* are large, that assumption is no longer plausible; ignoring cross-sectional slope heterogeneity results in inconsistent estimates. With relatively large *T*, the individual equations can be estimated for each unit separately for static and dynamic panels. In this section, we examine consistent estimators with models of cross-sectional erogeneity as $T \rightarrow \infty$. However, allowing for slope heterogeneity does not necessary rule out the shared features that cross-sectional units have in common, for example shared industry, or geography, economic and financial climate. In particular, while the dynamics of adjustment towards equilibrium differ among the cross-sectional units, they could converge to the same equilibrium in the very long-run; therefore, it may be useful to employ a panel model of slope heterogeneity also applicable to the dynamics of the error correction model. In this section, we examine consistent estimators with models of cross-sectional heterogeneity as $T \rightarrow \infty$ with fully different cross-section slopes, and a mixture of homogenous and heterogenous slopes.

First, let's examine the consequences of ignoring heterogeneity. Assuming a common fixed effects variable, the equation for a slope homogenous model is

$$y_{it} = \mu_i + \beta_i x_{it} + u_i$$

where β_i=β+η_i for η_i cross-sectional intercept (fixed or random). We can define the slope heterogeneity in terms of the η_i features. Suppose we wish to examine systematic dependence between η_i , the regressors x_{it} , and say another set of variables z_{it} , generated as new linear or nonlinear functions from x_{it} . We could formulate a model that ignores slope heterogeneity

$$y_{it}= \alpha_i +\delta_x x_{it} + \delta_z z_{it} + v_{it}$$

Let w_{it}= (x_{it}, z_{it})' independently distributed over time, with the covariance matrix as

$$\Omega_i=\begin{bmatrix} w_{ixx} & w_{ixz} \\ w_{izx} & w_{izz} \end{bmatrix}$$

Since the β_i are assumed fixed over time, the dependence of μ_i on w_{it} is ruled out. Then the FE estimations δ_x and δ_z are consistent if

$$\text{Cov } (w_{ixz}, \mu_i)= \text{Cov } (w_{ixx} , \mu_i)=0$$

Suppose δ_z are spurious, incorrectly included as regressors. The FE estimator δ_z is robust to slope heterogeneity if the included z_{it} are, on average, orthogonal to x_{it}. However, given slope heterogeneity, the FE estimator of δ_x are inconsistent even if z_{it} and x_{it} are orthogonal. The bias of estimated $\delta_{x,\, FE}$ is positive if Cov (w_{ixx} , μ_i) > 0 and vice versa. More generally, if E (w_{ixz}) $\neq$ 0 and Cov (w_{ixz}, μ_i) $\neq$ 0, and/or when Cov (w_{ixx} , μ_i) $\neq$ 0, the FE estimators of δ_{xFE} and δ_{zFE} are inconsistent. For example, consider spurious, incorrectly included RH quadratic terms in the above model by setting z_{it}=x_{it}^2. It is possible to reject linear slope heterogeneity not because of valid non-lineari~~l~~ties, but also because cross-sectional heterogeneity is disregarded. With the β_i assumed fixed over time, the nonlinear specification

$$y_{it}= \alpha_i +\delta_x x_{it} + \delta_z x_{it}^2 + v_{it}$$

cannot meet the above covariance conditions for consistency unless β_i varies proportionally with x_{it} . However, the variation can be systematically related to some other aspect of cross-sectional distribution of x_{it} unrelated to proportionality; for example

$$\beta_i =\Upsilon_0+ \Upsilon_1\bar{x}_{it}$$

where $\bar{x}_{it} = T^{-1}\sum_{t=1}^{T} x_{it}$. If linear slope heterogeneity is disregarded, it is possible to obtain a statistically ~~a~~ significant quadratic effect as explained in Figure 14.1 that shows three cross-sectional units (countries) with slopes that differ systematically with $\bar{x}_{it}$. The pooled regression based on the scatter points from all three would display pronounced non-linearities even though the unit-specific regression is linear.

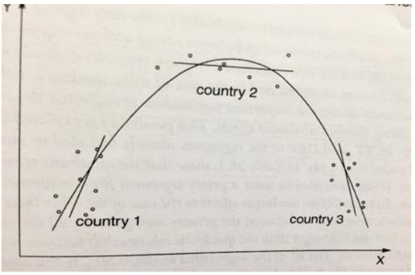


Figure 14.1 *Slope Heterogeneity*

The Mean-Group and Panel Mean-Group Estimators

We turn to an examination of the two alternative estimators among several that allow for slope heterogeneity. The estimators rely on *N* time-series regressions and averages coefficients. Assume an autoregressive distributed lag (*ARDL*) dynamic panel specification of the following type

$$Y_{it}=\sum_{j=1}^{p} \lambda_{ij} y_{it-j} + \sum_{j=0}^{q} \delta'_{ij} x_{it-j} + \mu_i + \varepsilon_{it} \qquad (14.2.1)$$

Where the number of groups $_i$=1, 2, …, *N*, and the number of time periods *t*=1, 2, …, *T*; x_{it} is a (*k*.1) vector of explanatory variables, δ_{ij} are the (k.1) coefficient vectors, λ_{ij} are scalars, and μ_i are group-specific fixed effect. The model is applicable to panels with large *T* because it requires fitting a separate regression for each group, and may include a time trend and other fixed regressors. If the variables are *I*(1), then the error term is *I*(0) if the variables are co-integrated and adjust to any deviation from long-run equilibrium. If so, it is common to re-parametrize the model as the error correlation equation:

$$\Delta Y_{it}=\phi_i (y_{it-1} - \theta_i' X_{it}) + \sum_{j=1}^{p} \lambda *_{ij} \Delta y_{it-j} + \sum_{j=0}^{q} \delta' *_{ij} \Delta x_{it-j} + \mu_i + \varepsilon_{it} \qquad (14.2.2)$$

Where subscripted * parameters indicate group mean avarages for each groupi *i*, ϕ_I are the error correction terms that measure the speed of adjustment to deviations from the equilibrium. For example, with the *ARDL* (1, 1, 1) dynamic panel specification, we have

$$\phi_i = -(1- \lambda_i), \ \theta_{0i} = \frac{\mu_i}{1 - \lambda_i}, \ \theta_{it} = \frac{\delta_{10i} + \delta_{11i}}{1 - \lambda_i}, \ \theta_{2i} = \frac{\delta_{20i} + \delta 21_i}{1 - \lambda_i};$$

the main parameters of interest are the error correction speed of adjustment coefficients ϕ_I, with θ_{1i} and θ_{2i} as the long-run coefficients.

One approach to the estimation of the above panel error correction model is to use the fixed effects method; all group time-series data are pooled and only the intercept is allowed to differ across groups. As discussed above, the method can lead to inconsistency. Pesaran and Smith (1995) proposed an alternative ***mean group*** **(*MG*)** estimator employing a simple arithmetic average of the coefficients that allows the intercept, slope coefficients, and error variances to change across groups (or panels). The *MG* is defined as the average of the OLS estimators, $\hat{\beta}_i$

$$\hat{\beta}_{MG} = \frac{1}{N} \sum_{i=1}^{N} \hat{\beta}_i$$

where $\hat{\beta}_i=(X_i'X_i)^{-1}X_i'y_i$. More generally, for the heterogenous slopes

$$\hat{\psi}_{MG} = \frac{1}{N} \sum_{i=1}^{N} \hat{\psi}_i$$

where the individual OLS slopes are $\hat{\psi}_i=(W_i'W_i)^{-1}W_i'y_i$. Moreover, the variance of $\hat{\psi}_{MGE}$ is also consistently estimated by

$$\widehat{Var}\,(\hat{\psi}_i)=\frac{1}{N(N-1)} \sum_{i=1}^{N}(\,\hat{\psi}_{MG} - \,\hat{\psi}_i)^2.$$

The *MG* parameters are the simple unweighted means of the individual coefficients. For large *N* and *T*, the *MG* estimator is asymptotically normal as long as $\sqrt{N/T}\rightarrow 0$ as $T \rightarrow\infty$. However, the MG estimator is biased when *T* is small; it is unlikely to be effective if either *N* or *T* is small.

However, slope heterogeneity may be false in the presence of long-run cointegration since that implies long-run homogenous parameters. Pesaran, Shin and Smith (1999), ***PSS***, proposed an extension of the MG estimator that permits a mixed long-run homogeneity for the error correction terms, and heterogenous parameter lag structure for short-term dynamics. That is, the assumption that the long-run coefficient of X_{it} , defined by $\Theta_i= -\beta_i/\varphi_i$, is the same across the cross-sectional unit equations:

$$\Theta_i= \Theta,\ i=1, 2, \ldots, N.$$

This estimator is known as the **Pooled Mean Group** (***PMG***) estimator, an intermediate estimator between *FE* and *MG* estimators. The *PMG* models error correction long-run homogeneity by *pooling* the observation over time. The error correction model under the *PMG* homogeneity assumption can be compactly written as

$$\Delta y_i = \varphi_i \check{\mathfrak{Z}}_i(\Theta) + W_i k_i + \varepsilon_i \qquad (14.2.3)$$

where * indicates mean group avarages and pooling has removed the variation by *t* (compare 14.2.2 with 14.2.2), $k_i = (\lambda^*_{i1}, \lambda^*_{i2}, \ldots, \lambda^*_{ip-1}; \delta^{*\prime}_{i0}, \delta^{*\prime}_{i1}, \ldots, \delta^{*\prime}_{iq-1})^{/}$; $W_i=(\Delta y_{i-1}, \Delta y_{i-2}, \ldots, \Delta y_{i-p+1}; \Delta X_i, \Delta X_{i-1}, \ldots, \Delta y_{i-q+1})$, the error correction compound term is $\check{\mathfrak{Z}}_i(\Theta)=y_{i-1} - X_i\Theta_i$. Three characteristics should be noted about this estimator. First, it imposes the cross-equation restriction for the long-run homogeneity assumption. Second, error variances differ across cross-sectional units. Finally, the regression equations for each unit are non-linear in Θ and φ parameters. To deal with the last issue for non-linear estimation, ***PSS*** (1999) develop a maximum likelihood method of estimation, combining both pooling and averaging, that expresses the likelihood as the product of each cross-section's likelihood, and after taking log results as:

$$\ell_T(\Theta', \varphi', \sigma') = -\frac{T}{2}\sum_{i=1}^{N}\ln(2\pi\sigma_i^2) - \frac{1}{2}\sum_{i=1}^{N}\frac{1}{\sigma_i^2}\{\Delta y_i - \emptyset_i\check{\mathfrak{Z}}_i(\theta)\}'\,\mathrm{H}\{\Delta y_i - \emptyset_i\check{\mathfrak{Z}}_i(\theta)\}$$

where $\varphi=(\varphi_1, \varphi_2, \ldots, \varphi_N)$, $\sigma = (\sigma_1^2, \sigma_2^2, \ldots, \sigma_N^2)$, $\check{\mathfrak{Z}}_i(\Theta)=(y_{it-1} - X_i\Theta_i)$ for the error correction parameters, and $H_i=I_T - W_i(W_i{'}W_i)W_i$ and I_T is an identity matrix of order *T*. The *PMG* estimator highlights the pooling effect of the homogeneity assumption using group averages to obtain group-wide mean estimates of the long-run error correction coefficients restricted to be the same across equations and the other short-run parameters. Note that for small *T*, the error correction standard errors of the *PGM* will be downward biased due to limited variation over time, and it may then be necessary to improve estimation accuracy by employing a bootstap biased-reduction procedure, see chapter 12.

14.3 *Hausman test of GM against PGM.*

Testing for cointegration can be achived by a test comparion of *MG* againt *PMG* estimates. *MG* has no imposed slope homogeneity assumption and due to its averaging method has contistent estimates. However, the consistency of the *PMG* estimator depends on whether the long-run slope elasticities are equal across all panels; otherwise, the *PMG* estimates are inconsistent. A *Hausman* model selection test, applied in chapter 4 to test *FE* against *RE* model is employed to decide the mpdel selection. The homogeneity hypothesis is rejected if the true model is heterogenous, while the *MG* estimates remain consistent in either case. A Hausman test determines whether *PMG* estimates are consistent if they do not significantly differ from the common parameters of the *MG* model estimates; otherwise, the *PMG* is rejected; that is, there is no coinegration and the error correction *PMG* model estimator is inapplicable.

Readings

For textbook discussion, see Pesaran (2015, chapters 28 and 22) for long *T* panel data with slope heterogeneity, full or mixed with homogeneity; and bounds test respectively. See Pesaran, Shin and Smith (1999) for mean group large *T* panel estimator; Pesaran et. al. (2003) developed the bounds test of panel data cointegration.

Chapter 14 *ARDL* Long Panel Cointegration Test & *VEC* Estimation Exercises

Q14.1 Download *davegiles-naturalgasprices.dta,* time-series of European and us gas prices.

***a.** Fit an ARDL autoregressive model of *eur* on *us*; select the number of *eur* lags by AIC and BIC, and test for co-integration by the *PSS* bounds test. Explain and comment on your test~~y~~ results.

Q14.2 Download *lutkepoh12.dta* contains quarterly series of investment, income and consumption in levels and log levels for W. Germany, 1960-82.

a. Fit an ARDL model for ln_inv as a linear function of ln_inc and ln_consump

b. Re-run the model in a. as an ARDL error correction model; identify the error correction estimate, and explain its interpretation within the ARDL model.

c. Test the model in **b.** for cointegration by PSS bounds test procedure. Explain how the outcome of PSS test is determined. Briefly explain the difference between the *PSS* test and the *IPS* (Im, Pesaran & Shin) test of the unit-root of chapter 9.

Q14.3 Download *SamieiOECD.dta* on saving/consumption data file for 21 OECD countries examined in Masson, Bayoumi and Samiei (1998, WBER).

a. Obtain PMG estimate for a differenced model of consumption regressed on income and inflation.

b. Test the theory that Ho: income elasticity=1, state the test result.

c. Since each group has its own estimated equation, predict the variable *id*, and apply cross-equation restriction for *id* when its value 111 and 112.

d. Obtain the MG estimates based on unweighted mean of N individual regression coefficients; compare the outcomes for PMG in a. and MG here.

e. Test *PMG's Ho: L-R* elasticities are equal across all panels, state the outcome.

Ch 15 Spectral Analysis of Time-series

Introduction

Time-series analysis discussed so far are all based on examining data in a *time-domain*; for example, we discuss the evaluation of autoregressive processes $\boldsymbol{Y_t}$ in terms of its autocovariance, or autocorrelation function, based on distinct time and displacement t & τ. We now turn to a complementary approach to time-series based on cycles of different frequencies; this *frequency-domain* approach to time-series is known as ***spectral analysis.*** Its aim is to decide how important are cycles of different frequencies in explaining the behavior of a time-series by employing an alternative function called the ***spectral density function.*** The two types of time-series analyses are complementary rather than mutually exclusive; application depends on which type offers a simpler representation of the key features of the data at hand.

15.1 *Modeling Time-series by cycles*

Instead of modeling the *AR, MA or ARMA* defined in terms of time-lags, spectral analysis describes the behavior of a time-series $\boldsymbol{Y_t}$ as a weighted sum of trigonometric periodic functions *cos*(ω) and *sin*(ω) for a particular frequency denoted by ω, that is by

$$Y_t = \mu + \int_0^{\pi} \alpha(\omega).con(\omega t)d\omega + \int_0^{\pi} \delta(\omega).sin(\omega t)d\omega \qquad (15.1.1)$$

We start by defining the **spectral distribution function**. Suppose a time-series contains a periodic sinusoidal component with a known wavelength modeled as

$$Y_t=R\ cos\ (\omega t+\varphi)+Z_t \qquad (15.1.2)$$

where Z_t ~ (0, 1) is dandom error, $\boldsymbol{\omega}$ is the frequency of the sinusoidal variation, $\boldsymbol{R}$ gives the **amplitude** of the variation (with a maximum at $+\boldsymbol{R}$ and a minimum at $-\boldsymbol{R}$), φ determines its **phase**, that is where in the cycle, $\boldsymbol{Y_t}$ would be at time $\boldsymbol{t}$=0. $\boldsymbol{\omega}$ measures how quickly $\boldsymbol{Y_t}$ cycles and is indicated by either of two following measures. The **period** or **wavelength** is the length of time required for the process to repeat a full cycle around the unit-circle's circumference of $2\boldsymbol{\pi}$, namely, if $\boldsymbol{\omega}$=1, then $\boldsymbol{Y_t}$ repeats itself every $2\boldsymbol{\pi}$ periods; if $\boldsymbol{\omega}$=2, then $\boldsymbol{Y_t}$ repeats itself every $\boldsymbol{\pi}$ periods. The **frequency** measures the number of cycles completed compared to the simple cos($\boldsymbol{t}$) wave during $2\boldsymbol{\pi}$ periods (equal to completing the circumference of a circle with a unit radius). For instance, if $\boldsymbol{\omega}$=2, the cycles are completed twice as fast as those for cos(t). There is a simple relationship between the two measures cyclical speed, that is equal to *wavelength*= $2\pi/\omega$. Chapman & Xing (2019) call (15.1.2) the **spectral distribution function**, but the function is also known by other names (see Figure 15.1 and below). A more convenient formulation of (15.1.2) is by writing the cycle as a combination of sine and cosine waves, replacing the amplitude and phase by two parameters **α** & **β** as

$$Y_t=\alpha\, cos\, (\omega t)+\beta\, sin\, (\omega t)$$

where $R=(\alpha^2 + \beta^2)^{1/2}$, and $\varphi=\tan^{-1}(\beta/\alpha)$

Looking at the models like (15.1.1) shows that they are not stationary if parameters are all fixed constants because $\boldsymbol{E(y_t)}$ will then change with time. The application of (15.1.1) to present stationary processes requires additionally assumptions that $[\boldsymbol{R_j}\}$ be uncorrelated random variables with mean 0, and $\{\boldsymbol{\varphi}_j\}$ random variables mean zero with a uniform distribution on $(0, 2\boldsymbol{\pi})$, in order to treat time-series as stationary processes.

As an example of (15.1.2) with $\boldsymbol{k}$=3 periods or cycles, we first generate three series, then construct an aggregate fourth combined series as follows:

$$X_{t1}=cos(10\frac{\pi t}{150}+\frac{\pi}{8}),\ X_{t2}= 3cos(30\frac{\pi t}{150}+\frac{3\pi}{8}),\ X_{t3}= 5cos(60\frac{\pi t}{150}+\frac{5\pi}{8}),\ Z_t \sim N(0, 1)\ \text{ and}$$

$$X_t= X_{t1}+ X_{t2}+ X_{t3}+Z_t$$

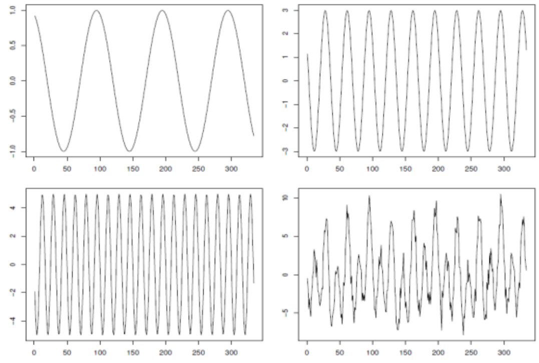


Figure 15.1 *Periodic components and their sum*

(top left: $\boldsymbol{X_{t1}}$*; Top right:* $\boldsymbol{X_{t2}}$*; Bottom left* $\boldsymbol{X_{t3}}$*; Bottom right:* $\boldsymbol{X_t}$*)*

Given the covariance-stationary process for $\boldsymbol{Y_t}$, we define its mean $\boldsymbol{E(Y_t)=\mu}$ and $\boldsymbol{j}$th autocovariance as $E(Y_t - \mu)(Y_{t-1} - \mu)=\gamma_j$. Assuming these autocovariances are well-behaved, namely, absolutely summable, the autocovariance function defined as a function of $\boldsymbol{\omega}$ is given by

$$g_Y(\mathrm{z})=\sum_{j=-\infty}^{\infty}\gamma_j z_j \qquad (15.1.3)$$

z denotes a complex scalar. If (15.1.3) is divided by $2\boldsymbol{\pi}$ and evaluated value of $\boldsymbol{z = e^{-i\omega}}$ where $\boldsymbol{e^{-i\omega}}$ is a complex exponential function with $\boldsymbol{i=\sqrt{-1}}$, see Mathematical Appendix, then Hamilton (1994) calls (15.1.3) the **population spectrum** of $\boldsymbol{Y}$ (though simply the **spectrum** is a more common name for this function)

$$s_Y(\omega) = \frac{1}{2\pi} g_Y\left(e^{-i\omega}\right) = \frac{1}{2\pi}\sum_{j=-\infty}^{\infty}\gamma_j\, e^{-i\omega j} \qquad (15.1.4)$$

According to (15.1.4), the spectrum is a function of $\boldsymbol{\omega}$ and can be calculated at a particular value of $\boldsymbol{\omega}$ for a sequence of its autocovariances. Using De Moivre’s theorem based on Euler’s rule, see exercise Q15.1, we can rewrite $\boldsymbol{e^{-i\omega j}}$ in terms of sinusoidal functions as

$$e^{-i\omega j} = \cos(\omega \mathrm{j}) - i.sin\,(\omega \mathrm{j}) \qquad (15.1.5)$$

Substituting for $\boldsymbol{e^{-i\alpha j}}$ in (15.1.3) leads to the equivalent form of the population spectrum in terms of cos and sin functions of $\boldsymbol{\omega}$:

$$s_Y(\omega) = \frac{1}{2\pi}\sum_{j=-\infty}^{\infty}\gamma_j\,[\cos(\omega \mathrm{j}) - i.sin\,(\omega \mathrm{j})] \qquad (15.1.6)$$

Using the symmetry of a covariance-stationary process, $\Upsilon_j=\Upsilon_{-j}$ to write out Υ_j & Υ_{-j} separately in terms of $(\omega \mathrm{j})$ & $(-\,\omega \mathrm{j})$ leads to

$$s_Y(\omega) = \frac{1}{2\pi}\sum_{j=-\infty}^{\infty}\gamma_0\,[\cos(0) - i.\sin\,(0)] + \frac{1}{2\pi}\sum_{j=-\infty}^{\infty}\gamma_j\,[\cos(\omega \mathrm{j}) + [\cos(\omega \mathrm{j}) - i.sin\,(\omega \mathrm{j}) - i.sin\,(-\,\omega \mathrm{j})] \qquad (15.1.7)$$

(15.1.6) and (15.1.7) are equivalent ways of writing the population spectrum but (15.1.6) allows using trigonometric results to simplify (15.1.4). Using the rules for

$$Cos(0)=1,\ sin(0)=0,\ sin(-\,\Theta)= -\,sin(\,\Theta);\ con(-\,\Theta)=con(\,\Theta),$$

the first square bracket equals 1, and the second square bracket is equal to (15.1.4) but has equal terms added to con and sin on either side of minus sign in the middle, resulting in:

$$s_Y(\omega) = \frac{1}{2\pi}\left\{\gamma_0 + 2.\sum_{j=-\infty}^{\infty} \gamma_j \, con\ (\omega \mathrm{j})\right\} \qquad (15.1.8)$$

This spectrum, with ωj independent of *N*, is also known as the *Fourier Transformation* of the autocovariance function (**acv.f**), also equivalently expressed as an expotental function of ω, see section 15.2.

Since $con(\omega j + 2\pi)$=$con(\omega j)$, then it follows from (15.1.8) that $s_Y(\omega_k + 2\pi)$)= $s_Y(\omega)$ for any integer ***K***. Therefore, the spectrum is a periodic function of $\boldsymbol{\omega}$; if we know the value of $\boldsymbol{s_Y(\omega)}$ for all $\boldsymbol{\omega}$ between 0 and π, we can calculate the value of $\boldsymbol{s_Y(\omega)}$ for any $\boldsymbol{\omega}$. However, the reverse is also true; given a spectrum and a value of $\boldsymbol{\omega}$, we can calculate the corresponding covariance by inversion of (15.1.8) to express the autocovariance $\boldsymbol{\gamma_j}$ as a function of $\boldsymbol{s_Y(\omega)}$. The autocovariance generating function obtained by the inversion of (15.1.8) is known as the **power population spectrum** of ***Y***. The frequency-based autocovariance function measures the contribution of every $\boldsymbol{\omega}$ with different frequency in the 0 to $\boldsymbol{2\pi}$ to the variance of the time-series.

*An example: calculation of the population spectrum for **AR**(1) process:* $Y_t - \mu = (1 - \varphi L)^{-1}\boldsymbol{\varepsilon}_t$. According to (15.1.3) and (15.1.4), spectrum *scales* the series ***z*** by covariance γ_j =E(y_t-μ)($y_{t-\boldsymbol{\tau}}$ - μ), where the brackets consist of an infinite geometrical series of the residuals, see chapter 4. The series by ***z*** is defined exponentially as $\boldsymbol{z = e^{-i\omega}}$. This autocovariance-generating function can be written as (see Hamilton p.62):

$$s_Y(z) = \sigma^2(1+ \varphi_1 z+ \varphi_2 z^2+. . . + \varphi_q z^q) \times (1+ \varphi_1 z+ \varphi_2 z^{-2}+. . . + \varphi_q z^{-q}) \qquad (15.1.9)$$

Summarizing the autocovariance is through such a scalar-value function called the **autocovariance-generating function**. In general, a covariance-generating function with $\boldsymbol{z = e^{-i\omega}}$ is given by:

$$g_Y(z)=\sigma^2\psi(z)\ \psi(z^{-1}) \qquad (15.1.10)$$

where the infinite residual series in (15.1.9) is approximated by $\psi(z)$= 1/(1 – φ) equation whose solutions lead to the characteristic roots of the series that depend on the φ values and as long as |φ| < 1. Defining z as $\boldsymbol{e^{-i\omega}}$, (15.1.4) and (15.1.10) can also be expressed as:

$$g_Y(z)=(2\pi)^{-1}\sigma^2\psi(\boldsymbol{e^{i\omega}})\ \psi(\boldsymbol{e^{-i\omega}}) \qquad (15.1.11)$$

Using (15.1.3) and (15.1.4) with $\boldsymbol{z = e^{-i\omega}}$, the application of (15.1.10) and (15.1.11) leads to the spectrums of *AR*(1) process for the *MA*, *AR*, and *ARMA* processes (see end of chapter exercises).

15.2 *Fourier Analysis*

The autocovariance $f(\omega)$ as the inverse of the spectrum is a continuous function in [0, π] interval with the derivative obtained from:

$$f(\omega) = \frac{dF(\omega)}{d\omega} \qquad (15.2.1)$$

(15.2.1) can be expressed in the form

$$\gamma\,(k) = \int_0^{\pi} cos\ \omega_k f(\omega) d\omega \qquad (15.2.2)$$

Putting *k*=0, we have

$$\gamma\,(0) = \sigma_x^2 = \int_0^{\pi} f(\omega) d\omega = F(\pi) \qquad (15.2.3)$$

$f(\omega)d\omega$ represents the contribution of the spectrum to variance of the components with frequencies in $[\omega, \omega + d\omega]$ range; (15.2.3) indicates that the total area under the spectrum curve is equal to the variance of the process. It is important to note that the *spectrum and the autocovariance function (**acv.f.**) are equivalent ways of describing a stationary stochastic process*; they complement each other, expressing the same information in different ways.

(15.2.2) expresses γ (k) in terms of $f(\omega)$ as a cosine transformation. The corresponding inverse relationship can be shown to be

$$f(\omega) = \frac{1}{\pi} \sum_{k=-\infty}^{\infty} \gamma(k) e^{-i\omega_k} \qquad (15.2.4)$$

(15.2.4) spectrum is known as the **Discrete Fourier transform** (***DFT***) of the ***acv.f*** . The DFT establishes a representational relationship between the discrete time domine signals and their transformation into the equivalent in frequency domine; in practice such a transformation would be too slow to be helpful; instead, a much faster equivalent procedure is relied upon to implement the transformation, see section 15.4.2 below. (15.2.3) & (15.2.4) together are called the Fourier transform ***pair***. The transform is usually written in the equivalent form as:

$$f(\omega) = \frac{1}{\pi}\ [\gamma(0) + 2 \sum_{k=-\infty}^{\infty} \gamma(k) cos\omega_k] \qquad (15.2.5)$$

It sometimes appears in normalized form given by

$$f^*(\omega) = \frac{f(\omega)}{\sigma_x^2} = \frac{dF^*(\omega)}{d\omega} \tag{15.2.6}$$

where $f^*(\omega)$ is the *Fourier transform of the* ***acv.f***.; written equivalently

$$f^*(\omega)=\frac{1}{\pi}\ [1 + 2\sum_{k=-\infty}^{\infty} \rho(k) cos\ \omega_k \tag{15.2.7}$$

Where $f^*(\omega)d\omega$ is the *proportion of variance* in the interval $[\omega, \omega+d\omega]$.

Fourier analysis, also known as **Harmonic analysis**,[18] plays an important role in spectral time-series models in providing approximation for a function by the sum of its sine and cosine terms called the **Fourier series representation**. Suppose $\boldsymbol{f(t)}$ is defined on (- $\boldsymbol{\pi, \pi}$] (different shape brackets indicate the lower limit - $\boldsymbol{\pi}$ is excluded from the interval). Then, given a finite number of discontinuity and number of maxima and minima, the ***f(t)*** function can be approximated by the Fourier series over an interval of ***r***=1, 2, …***k*** of different wavelengths.

$$\frac{a_0}{2} + \sum_{r=1}^{k} (a_r\ cos(rt) +\ b_r\ sin(rt) \tag{15.2.8}$$

where $a_0\ = \frac{1}{\pi} \int_{-\pi}^{\pi} f(t)dt$, $a_r\ = \frac{1}{\pi} \int_{-\pi}^{\pi} f(t)\ cos(rt)\ dt$, $b_r\ = \frac{1}{\pi} \int_{-\pi}^{\pi} f(t)\ sin(rt)\ dt.$

The Fourier application to a time-series partitions the total sum of squares into a residual component and an explained sum of squares by the periodic component at frequency ω (similar to the ANOVA analysis of variance); the latter component is given as

$$\sum_{j=1}^{M} \delta_j^2\ [(\hat{\beta} cos^2(\omega jt) + \hat{\beta} sin^2(\omega jt)]$$

Since the trm inside the squared brackets is unity, that provides the aggregate variance for a simple deterministic sinusoidal at a known **ω**, X_t=*μ*+*α.conωt*+β.*sinωt*+Z_t. The last equation can be decomposed by (15.2.8) into the different wavelengths contributing to the variance.

The upper bound $\boldsymbol{\pi}$ called the **Nyquist frequency** is the highest frequency about which the data can provide meaningful information; it is expressed as cycles per unit time as f_N=ω_N /2 $\boldsymbol{\pi}$.

[18] A harmonic series is the sum og its positive unit-fractions: $\sum_{=1}^{\infty} \frac{1}{n} = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} +\ldots$

The lowest frequency is called the **fundamental Fourier frequency** because the Fourier representation of the data is normally evaluated at the frequencies that are all integer multiplies of the fundamental frequency; the integers are called **harmonics**. When ***f(t)*** is a *periodic* with ***T*** period so that ***f(t)=f(t + nT)***, then ***f=1/T*** or $\boldsymbol{\omega} = 2\,\boldsymbol{\pi}/\boldsymbol{T}$ becomes the fundamental and the Fourier representation of ***f(t)*** is the sum of the over integer multiples, or harmonics. Basically, the fundamental frequency of a time-series is its first frequency, while its harmonics are all remaining frequencies. Note that the highest frequency does not depend on *N* while the lowest frequency *does* depend on N. This raises the important question about the consistency of the spectral estimation based on the Fourier series discussed in section 15.5.

15.3 *Calculation of Covariances from Population Spectrum*

If we know the sequence of autocovariances $\{\boldsymbol{\gamma_j}\}_{j=-\infty}^{\infty}$, then (15.1.2) and (15.1.7) shows we can calculate the value of time-series as a function of its periodic functions for a given value of $\boldsymbol{\omega}$, and conversely, given the value of $s_Y(\boldsymbol{\omega})$ for all $\boldsymbol{\omega}$ in [0, $\boldsymbol{\pi}$], we can calculate the value of any autocovariance $\boldsymbol{\gamma_k}$ for any displacement value ***k***.

i. *Proposition 1*

The formula for calculating any autocovariance from the population spectrum is given by the following proposition:

Proposition 1: Let $\{\boldsymbol{\gamma_j}\}_{j=-\infty}^{\infty}$ be an absolutely summable sequence of autocovariances, and (15.1.3) its autocovariance generating function, then

$$\int_{-\pi}^{\pi} s_Y(\omega) e^{iwk}\ d\omega = \gamma_k \tag{15.3.1}$$

Using De Moivre’s theorem (15.1.5) and (15.1.3) applied to the sum of trigonometric functions, (15.1.8) can be equivalently ~~be~~ written as:

$$\int_{-\pi}^{\pi} s_Y(\omega) \cos\ (\omega k) d\omega = \gamma_k \tag{15.3.2}$$

Let’s first obtain the result of the proposition for the variance of *y*, i.e. with displacement ***k***=0.

$$\int_{-\pi}^{\pi} s_Y(\omega)\, d\omega = \gamma_0 \tag{15.3.3}$$

Hence, the area under the population spectrum between $[-\pi, +\pi]$ is equal to the variance of Y_t, γ_0. Next, rewrite (15.3.2) for $k\neq 0$ to account for nonnegative autocovariances between

[t, t+k] periods Y

$$\int_{-\omega 1}^{\omega} s_Y(\omega)\, d\omega = \gamma_\omega \qquad (15.3.3)$$

(15.3.3) would be a positive number for any frequency ω_1 between [0, π], we can take that value as the proportion of the variance of Y_t due to frequencies ω less than ω_1 in absolute value; as $s_Y(\omega)$ is a symmetric function, the value of (15.3.3) between [- ω_1 , 0] $=$ [0, $+$ ω_1], we have

$$2. \int_{0}^{\omega_1} s_Y(\omega)\, d\omega = \gamma_\omega \qquad (15.3.4)$$

To understand the reason that (15.3.3) & (15.3.4) measure the periodic random components less than $\boldsymbol{\omega_1}$ rather than $\omega 1$ itself, consider a portion of the variance of a special $\mathbf{Y_t}$ stochastic process attributed to cycles with frequencies $\leq \omega_1$; suppose the value of Y at time t for $\boldsymbol{M}$ different frequencies is, using (15.1.1), given by:

$$Y_t = \sum_{j=1}^{M}[\alpha_j.\cos(\omega_j t) + \delta_j.\sin(\omega_j t)] \qquad (15.3.5)$$

For (15.3.5), α_j & $\delta_j$are two mean-zero random variables, therefore, $E(Y_t)$=0 for all t, and $\{\alpha_j\}_{j=1}^{M}$ & $\{\delta_j\}_{j=1}^{M}$ are serially and mutually uncorrelated; thus the variance σ_j^2 remains unchanged for all j & k , so $E(Y_t)$= 0, independent of t. The variance of Y_t can be simplified to:

$$E(Y_t^2) = \sum_{j=1}^{M}[(\alpha_j^2).cos^2(\omega_j t) + (\delta_j^2).sin^2(\omega_j t)]$$

$$= \sum_{j=1}^{M}\sigma_j^2\, [cos^2(\omega_j t) + sin^2(\omega_j t)]$$

$$= \sum_{j=1}^{M}\sigma_j^2 \qquad (15.3.6)$$

cycles of ω_j frequency are equal $to\ \sigma_j^2$. The portion of the variance of Y due to cycles $\leq \omega_j$, given the ordered frequencies $0 < \omega_1 < \omega_2 < \ldots < \omega_M < \pi$, is equal to $\sum_{j=1}^{j}\sigma_j^2$. Then the kth autocovariance of Y becomes

$$E(Y_t\, Y_{t-k}) = \sum_{j=1}^{M}\{(\alpha_j^2).cos(\omega_j t).cos[\omega_j(t-k] + (\delta_j^2).sin(\omega_j t).\, sin[\omega_j(t-k)]\}$$

$$= \sum_{j=1}^{M}\sigma_j^2\, .\{cos(\omega_j t).cos[\omega_j(t-k)] + sin(\omega_j t).\, sin[\omega_j(t-k)]\}$$

Employing the trigonometric identity cos(A – B)=cos(A).cos(B)+sin(A).sin(B) with A=$\omega_j t, B = \omega_j(t-k)$, & (A – B)= ω_{jk}, the ***k***th autocovariance of Y simplifies to a function independent of time as

$$E(Y_t\, Y_{t-k}) = \sum_{j=1}^{M} \sigma_j^2\, .cos(\omega_j k) \qquad (15.3.7)$$

Since neither (15.3.6), and nor (15.3.7) are functions of time, (15.3.5) is a covariance-stationary process, namely, depends only on the displacement k.

This outcome is contingent on the special covariance-stationary nature of (15.3.5), but why should there be a finite sum of frequencies involved in (15.3.6) and (15.3.7)? However, a similar general result, known as the **spectral representation theorem,** states that as $\boldsymbol{j} \rightarrow \infty$**,** the same results hold for *any* covariance-stationary process. Given any fixed frequency $\boldsymbol{\omega}$ in [0, $\boldsymbol{\pi}$], with defined random variables α(ω) & δ(ω) , a stationary process with summable autocovariances can be written as:

$$Y_t = \mu + \int_0^{\pi}[\alpha(\omega).con(\omega t) + \int_0^{\pi}\delta(\omega).sin(\omega t)]d\omega \qquad (15.3.8)$$

(15.3.8) has the properties that the random variables $\alpha(.)$ & $\delta(.)$ are serially uncorrelated over time and also uncorrelated with each other; one can, therefore, calculate the portion of the variance of Y_t due to cycles less than or equal to some specified ω_1 by (15.3.8) the generalization of (15.3.7).

Summarizing, we invert the spectrum of a time series as a function of ω and autocovariance to obtain variance and covariance; given orthogonality of cosine and sine terms, we then obtain the spectrum representation theorem as the linear sum of cosine and sine (ωt) over the interval [0, **π**] for a pair of functions of the time series and its corresponding autocovariance.

ii. *Sample Periodogram*

The analog of (15.1.1) estimated from sample data is known as the **sample periodogram**; the same calculation employed to obtain (15.1.8) applied to data shows that the area under the periodogram is the sample variance of y. Moreover, the finite sample analog of (15.1.1) for a time-series $\boldsymbol{Y_t}$ as a weighted sum of trigonometric periodic functions, given sample orthogonality of each periodic function and orthogonality between them, provides ~~a~~ finite sample estimates of the partitioned portions of the population variance for cycles with frequencies $\boldsymbol{\omega}_{\mathbf{j}}$ from the sample periodogram. The periodogram as a sample analog for the spectrum is expressed as:

$$\hat{s}_Y(\omega) = \frac{1}{2\pi}\left\{\gamma_0 + 2.\sum_{j=-\infty}^{\infty} \hat{\gamma}_j\, con\ (\omega\ \mathrm{j})\right\}$$

The same calculation as before also leads to the area under the periodogram as being the analog for the population variance and given its symmetry around **ω**=0, is equal to

$$\hat{\gamma}_j = 2\int_0^{\pi} \hat{s}_Y(\omega)d\omega$$

The sample variance of y is T-1=$\sum_{t=1}^{T}(y_t - \bar{y})^2$, and the portion of this variance from cycles ω_j can be obtained from the sample periodogram $\hat{s}_Y(\omega_j)$. If the sample size T is an odd number and the number of periodic functions with different frequencies as M≡(T-1)/2, then ω1=$(2\pi/T)$ with M=1, ω_2=$(4\pi/T)$ with M=2, . . . , ω_M = $(2M\pi/T)$ with M periodic functions; and the highest frequency is ω_M={$[2(T-1)\pi]/2T$}<$\boldsymbol{\pi}$. Therefore, the constant factor of proportionality for all **ω**j cycles is $2\pi/T$. Now consider an *OLS* regression of y_t on a constant and on the various cosine and sine terms as:

$$Y_t = \mu + \sum_{j=1}^{M}\{\alpha_j.\cos[\omega_j(t-1)] + \delta_j.\sin[\omega_j(t-1)]\} + u_t$$

Run as a usual *OLS* regression. The coefficients of this model have the property that $\frac{1}{2}(\hat{\alpha}_j^2 + \hat{\delta}_j^2)$ represents the portion of the sample variance attributed to cycles with frequency ω_j; this quantity is also proportional to the sample periodogram assessed at $\omega_j$frequency.

The proof of the above claim that the sample periodogram measures the part of the sample variance of y that results from cycles of different frequencies is rather long, see Hamilton (1994) Appendix 6.2(a)-(c). Here, we comment on the critical components of this proposition for the case of an *odd* number of sample observations

Proposition 2: Let T denote an odd integer, M ≡(T-1)/2, ω_j =2 πj /T for j=1, 2, . . . , M; and let

T observations on a process be {$y_1, y_2, \ldots, y_T$}. Then the following are true:

(a) The value of $\boldsymbol{y_t}$ can be expressed as:

$$y_t = \hat{\mu} + \{\ \hat{\alpha}_j.\cos\ [\omega_j\,(t\text{-}1)] + \hat{\delta}_j.\sin\ [\omega_j\,(t\text{-}1)]\} \qquad (15.3.9)$$

with the sample mean $\boldsymbol{\bar{y}}$=$\boldsymbol{\hat{\mu}}$ and for $\boldsymbol{j}$=1, 2, . . . , $\boldsymbol{M}$

$\widehat{\alpha_J}$ = (2/T)$\sum_{t=1}^{T} y_t . cos[\,\omega_{\text{j}}\,(\text{t-1})]$ (15.3.10)

$\widehat{\delta_J}$ = (2/T)$\sum_{t=1}^{T} y_t . sin[\,\omega_{\text{j}}\,(\text{t-1})]$ (15.3.11)

(b) The sample variance of ***y_t*** can be expressed as

$$\left(\frac{1}{T}\right)\sum_{t=1}^{T}(y_t - \bar{y}\,)^2 = (\tfrac{1}{2})\sum_{j=1}^{M}[(\,\widehat{\alpha_J}\,)^2 + (\,\widehat{\delta_J}\,\,)^2] \qquad (15.3.12)$$

and the portion of the sample variance of ***y*** due to cycles of frequency ***ω_{j}*** is given by:

$$\frac{1}{2}[(\,\widehat{\alpha_J}\,)^2 + (\,\widehat{\delta_J}\,\,)^2]$$

(c) the portion of the sample variance of *y* due to cycles of frequency ω_{j} can be equivalently expressed as:

$$(\tfrac{1}{2})\sum_{j=1}^{M}[(\,\widehat{\alpha_J}\,)^2 + (\,\widehat{\delta_J}\,\,)^2] = (4\pi/T).\,\hat{s}_y\,(\omega_{\text{j}}) \qquad (15.3.13)$$

where $\hat{s}_y\,(\omega_{\text{j}})$ is the sample periodogram at frequency ω_{j} and where $4\pi/T$ is the constant factor of proportionality.

Regarding (a), y_t in (15.3.9) has *M con* & *sin* cycles plus a constant, therefore, (*2M+1*)=*T* elements, so the number of variables is equal to the number of observations. Given the linearly independent elements, a least square regression of (15.3.9) produces a perfect fit with no error term.

Moreover, the OLS coefficients in this case have the property that the magnitude of $\frac{1}{2}[(\,\widehat{\alpha_J}\,)^2 + (\,\widehat{\delta_J}\,\,)^2]$ represents the portion of the sample variance due to cycles with frequency ω_j; the magnitude turns out to be proportional to the sample periodogram evaluated at ω_j. Therefore, we can find the portion of the sample variance due to cycles with frequency ω_j from the sample periodogram. Moreover, note that the proposition excludes negative ωj; it also confines ω_{j} to the [0, π] range, that is, ω_{j} must not be larger than π. As for ω < 0, consider a special case of the process in (15.3.6)

$$y_t = \boldsymbol{\alpha}.\cos(-\omega t) + \delta.\sin(-\omega)$$

for zero-mean $\boldsymbol{\alpha}$ & δ random variables. However, because $\cos(-\omega t)$= $\cos(\omega t)$ and $\sin(-\omega t)$= $-\sin(\omega t)$, this function is observationally equivalent to

$$y_t = \boldsymbol{\alpha}.\cos(\ \omega t)\ -\delta.\sin(\omega)$$

Thus, one cannot tell if the data was generated by a ω>0 cycle or by a ω <0 cycle; by convention we focus only on ω>0 cycles. Moreover, the largest frequency that can be considered in (a) is $\omega = \pi$. Consider if the data were generated by a process with frequency of $\omega > \pi$, for example ω= 3π/2. Then we have:

$$y_t = \boldsymbol{\alpha}.\cos[(3\pi/2)\mathrm{t}] + \delta.\sin[(3\pi/2)t]$$

$$y_t = \boldsymbol{\alpha}.\cos[(-\pi/2)\mathrm{t}] + \delta.\sin[(-\pi/2)t]$$

The shortest-period cycle observable is one repeating itself every $2\frac{\pi}{\pi}$=2; if ω= 3π/2, the cycle repeats itself every $(2\,\pi/1)/(3\pi/2)$=4/3 periods. However, the data is observed only at integer dates, the sampled data will exhibit cycles every four periods with frequency ω= π/2. Once again, cycles of frequency ω= 3π/2 cannot be observationally distinguished from cycles with frequency π/2.

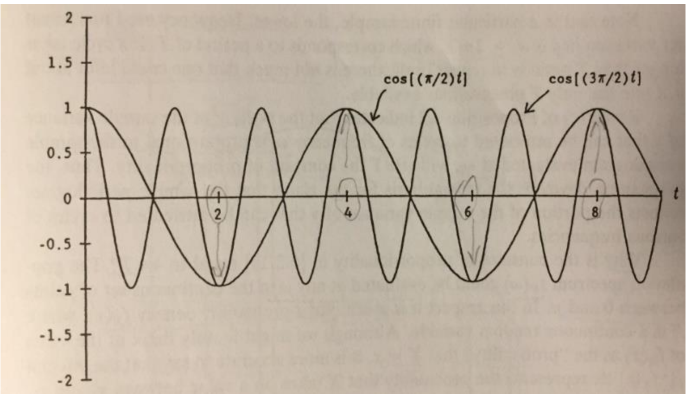


Figure 15.2 Aliasing: plots of cos($\boldsymbol{\pi}$/2)/*t* & cos(3$\boldsymbol{\pi}$/2)/*t* as functions of *t*

Summarizing, if the data generating process includes the two periodic functions such that $\omega > \pi$ and $\omega < 0$ are observationally indistinguishable from their corresponding $\omega = \pi/2$; these cycles will be imputed to those with frequencies between [0, π]. This is known as aliasing. Figure 15.2 explains this problem with the plots of these two functions of *t*. Although the function *cos*[(3π/2)*t*] repeats itself whenever *t* increases by 4/3, one would only observe y_t at four distinct dates (y_t, y_{t+1}, y_{t+2}, y_{t+3}) before seeing the value of *cos*[(3π/2)*t*] repeating itself for an integer value of *t.*

The more general result is that sampling will have an effect, in that variation at frequencies above the Nyquist frequency will be `folded back' to produce apparent variation in the sampled series at a frequency lower than the Nyquist frequency. If we denote the Nyquist by the portion of the sample due to cycles with frequency $\boldsymbol{\omega_j}$**,** then ω, $(2\omega_N - \omega)$, $(2\omega_N + \omega)$, $(4\omega_N - \omega)$, **, . .** are **aliases** of each other, that is they are observationally indistinguishable; variations at all these frequencies will appear as variation at frequency $\boldsymbol{\omega}$ in the sampled data.

Note that given a finite sample, the lowest frequency used in describing the variation in y is ω_1=$2\,\pi/T$. Proposition 2_c maintains that the portion of the sample variance due to cycles of frequency ω_j is proportional to the sample periodogram evaluated at ω_j ; while proposition 2_b maintains that the constant of proportionality is equal to $2\pi/T$. This is the basis of the claim that the periodogram measures the portion of the sample variance of y due to different cycles. However, it would be misleading to interpret that to mean that the value of $s_Y(\omega)$ represents the contribution of cycles with frequency ω to the variance of Y, and more accurate to mean it represents the contribution of cycles between ω_1 & ω_2. Assuming $s_Y(\omega)$ is continuous, the contribution of a cycle with a particular value is zero. One should therefore interpret $\frac{1}{2}[(\,\widehat{\alpha_J}\,)^2 + (\,\widehat{\delta_J}\,\,)^2]$ as the portion of the sample variance due to cycles with frequency near ω_j rather than exactly at ω_j; in other words, (15.3.3) is not an estimate of the height of the population spectrum but of the area under the population spectrum. Figure 2 illustrates this issue.

Suppose $\frac{1}{2}[(\,\widehat{\alpha_J}\,)^2 + (\,\widehat{\delta_J}\,\,)^2]$ is an estimate of the portion of the variance with frequency between ω_j & ω_{j-1} , equal to twice the area under $s_Y(\omega)$ between ω_j & ω_{j-1}. Since ω_j= $2\,\frac{\pi j}{T}$;

(ω_j - ω_{j-1})= $2\,\frac{\pi}{T}$. Then the area under $s_Y(\omega)$ between ω_j & ω_{j-1} would be approximated by the area of a rectangle with width of $\frac{2\pi}{T}$ and height of $\widehat{s_Y}(\omega)$; therefore $\frac{1}{2}[(\,\widehat{\alpha_J}\,)^2 + (\,\widehat{\delta_J}\,\,)^2]$= $\widehat{s_Y}(\omega).\,(2\,\pi/T)$ as ~~the~~ proposition (c) maintains.

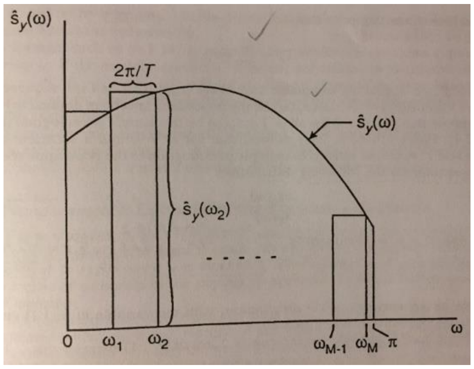


Figure 15.3 *variance by the area under the sample periodogram due to cycles of different frequencies*

Finally, the proposition provides a calculation method for the value of the periodogram at frequency $\omega_j = 2\,\pi j/T$ for j=1, 2, . . . , (T -1)/2:

$$\hat{s}_y\,(\omega_\mathrm{j}) = \Big[\frac{T}{8\pi}\Big]\big[\big(\,\hat{\alpha}_j\,\big)^2 + \big(\,\hat{\delta}_j\,\big)^2\big]$$

using (15.3.13), and (15.3.10)-(15.3.11), then

$$\hat{s}_y\,(\omega_\mathrm{j}) = \Big[\frac{1}{2\pi T}\Big]\{[\,\textstyle\sum_{t=1}^{T} y_t.\, cos[\,\omega_\mathrm{j}\,(t\text{-}1)]^2 + [\sum_{t=1}^{T} y_t.\, sin[\,\omega_\mathrm{j}\,(t\text{-}1)]^2$$

because (2/T) in front of (15.3.10)-(15.3.11) is squared and (T/8π).(2/T)2=1/2 πT.

15.4.1 ***Estimating the Population Spectrum***

We examine the large sample properties of the periodogram by estimation of the population spectrum $s_Y(\omega)$ by sample periodogram $\widehat{s_Y}(\omega)$, given an observed sample for y_t. It can be shown that with a sufficiently large sample size T, for $\widehat{s_Y}(\omega)$ with $\omega \neq 0$, twice the ratio of the sample periodogram to the population spectrum is approximately χ^2 distributed as

$$2.\,\widehat{s_Y}(\omega)/\, s_Y(\omega) \approx \chi^2_{(2)} \qquad (15.4.1)$$

Therefore, the expected value of (15.4.1) is n=2 (the mean and variance of a χ^2 distribution with *d.f.=n* equal n and $2n$ respectively). Therefore:

$$E[2.\,\widehat{s_Y}(\omega)/\, s_Y(\omega)] \cong 2$$

Since $s_Y(\omega)$ is a population spectrum; E[*2* $\widehat{s_Y}(\omega)$.]=*2.* $s_Y(\omega)$; thus

$$E[\,\widehat{s_Y}(\omega)] \cong s_Y(\omega) \qquad (15.4.2)$$

(15.4.2) shows that the sample periodogram provides an approximately unbiased estimate of the population spectrum, given a sufficiently large sample size. However, a 95% confidence interval for $\chi^2_{(2)}$ falls between 0.05 and 7.4, or (for n=2) 0.025 and 3.7. This means $\widehat{s_Y}(\omega)$would have to be as small as 0.025 times $s_Y(\omega)$ and larger than 3.7 times $s_Y(\omega)$to remain statistically significant at 5%; an unlikely outcome for $\widehat{s_Y}(\omega)$. Given such a large confidence interval, $\widehat{s_Y}(\omega)$ is not a satisfactory estimate for $s_Y(\omega)$. Can we deal with this problem by increasing the sample size? The answer is no. As already pointed out, (15.3.8) using *N* observations for different cycles per unit time to estimate and equal number of parameters; its estimation produces a perfect fit. Therefore, with sufficiently large *N* we can obtain an unbiased estimate of the population spectrum by the periodogram, but as *N* increases, so do the number of parameters estimates. Thus, even with a large sample size, an increase in size does not generate additional degrees of freedom, with the paradoxical outcome that $\widehat{s_Y}(\omega)$ *is an unbiased but inconsistent estimator of* $s_Y(\omega)$. In short, these limitations suggest that to produce a good estimate of a spectrum, the periodogram must be appropriately modified in application, a task to which we turn now.

There are several such smoothing methods of the periodogram estimates. We discuss one that has gained popularity in recent times. The periodogram provides estimates of the variance and autocovariance parameters $\boldsymbol{c_0}$ and $\boldsymbol{c_j}$, so one can write the periodogram equivalent of (15.1.8) as

$$\widehat{s_Y}(\omega) = \frac{1}{2\pi}\left\{\lambda_0 c_0 \ + 2.\sum_{j=1}^{M} \lambda_j \, c_j \, con \ (\omega \ \mathrm{j})\right\} \qquad (15.4.3)$$

Here $\{\boldsymbol{\lambda_j}\}$ are a set of weights called the **lag window** and the number of cycles *M*< *T* the *truncation point*. (15.4.3) shows that c_j for M<J<T are not used, while the values of c_j for $J \leq M$ are weighted by a factor of λ_j; the weights chosen so as to get smaller as *j* approaches *M*, that is, as the number of frequencies becomes larger. In order to use this procedure, one must employ an appropriate lag window and truncation point. The choice of the truncation point is subjectively based on the *balancing bias against variance* (see below).

15.4.2*The Fast Fourier Transform*

The periodogram is the finite Discrete Fourier Transform representation, it demonstrates the conversion of a time-series from time domine into frequency domine. The conversion involves a multiplication of data of a column vector of signals in time-domine by a matrix containing conversion values that leads to a corresponding column of signals in frequency domine. For a sample size of *N*, this task would require N^2

matrix operations; the conversion would be of the order of N^2, or O (N^2). If N is a large number, the application of the *DFT* becomes a cumebersome, slow, non-linear procedure and effectively infeasible. One popular alternative that simplifies the number of operations is based on an algorithm known as the **Fast Fourier Transform** (***FFT***). The *FFT* is computationally complex but more efficient and much less time-consuming than the *DFT* as long as the algorithm depends on the factorization of N as a *composite number*[19] to the power of 2; expressed as O ($N \log_2$ (N)). The procedure divides the column factor of time-series data two groups of even and odd observations stacked on top of each other to carry out the matrix operations, and further divide each group in turn into even and odd observations so as to simplify the matrix operation. For example, if 2^{10}=1024, this "divide and conquer" strategy reduces, in repeated steps, a huge matrix of 1024 elements until it becomes a simple 2 by 2 matrix, $F_{1024} \rightarrow F_{512} \rightarrow F_{254} \rightarrow ... \rightarrow F_4 \rightarrow F_2$, so as to approximately linearize the series into O ($N \log_2$ (N)). This method therefore requires T not to be a prime number and therefore can be factorized; that is, if T is even, in T=$r.s$ form at least one of the factors, say r, will be even; T is then a *composite* number. What if the N is not a composite number to a power of 2? Then one can always add a series of zeros to the end of the observation number to make it so, a procedure called **Tapering** or **data windowing**, and then apply the *FFT*. The application of *FFT* then requires increasing the length of the data by creating a highly composite number for T with additional zero~~s~~ observations to T so as to make the number of the form 2^k; after removing any linear trend from the data. For example, T=382 observations is *not* highly composite, but one can increase the length of the data beyond that number by making it equal to 2^9=512 and then add 512-382=130 zeros at one end. The application of the FFT first calculates the Fourier coefficients of the mean-corrected and average their squared values $[(a_p)^2 + (b_q)^2]$ in groups of around 10. To obtain most benefit from the FFT, the data should be large and in many thousands.

15.5 *Estimation of the spectrum*

One alternative for periodogram estimation is to fit an autoregressive ***AR*** or ***ARMA***, called **autoregressive spectrum estimation**, to the data. Suppose the data can be modeled by an ***ARMA*(*p, q*)** with ~~a~~ white noise and variance ε_t and σ^2. Then, one can for instance, first estimate

[19] A composite number is formed by multiplication of two positive integers, it is a prime number or 1, hence it also a positive integer with at least one divisor other than 1 and itself. Hence, 14 is a composite number because it is mutple of 2*7, while the prime numbers 2 and 3 are not.

the AR (1) parameters for y_t=c+ϕy_{t-1}+ε_t, by maximum likelihood, and insert theϕ estimates into the $s_Y(\omega)$ formula for *ARMA* (*p*, *q*), we examined this approach to a full ARFIMA model estimation in chapter 9, see also exercise Q9.4.

The alternative approach is to assume that $s_Y(\omega)$ will be close to $s_Y(\lambda)$ with $\{\lambda_j\}$ as lag window weights when ω is close to, or in the 'neighborhood' of, λ. This assumption is the basis of **nonparametric or kernel estimation.** The assumption implies an estimation approach based on a weighted average of the values of $\widehat{\boldsymbol{s_Y}}(\boldsymbol{\lambda})$for values **λ** in the neighborhood around ω with the weights depending on the distance between values λ and ω. Let ω_j=2 $\pi j/T$; the implication is that

$$\hat{s}_{y_Y}(\omega_j)= \sum_{m=-h}^{h} k(\, \omega_{j+m},\ \omega_j).\ \hat{s}_y\,(\omega_{j+m}) \qquad (15.5.1)$$

Where *m* is the distance between λ and ω in the vicinity of ω, *h* is a bandwidth parameter indicating how many different frequencies are considered as helpful to the estimation of $s_Y(\omega_j)$; the *kernel* $k(\omega_{j+m},\ \omega_j)$ indicates how much weight each frequency receives; the kernel weights sum up to unity:

$$\sum_{m=-h}^{h} k(\, \omega_{j+m},\ \omega_{j)\ =\ 1}$$

One method is to make $k(\omega_{j+m},\ \omega_j)$ proportional to [h +1 - |m|], then one can show, see Hamilton section 6.3, that

$$\sum_{m=-h}^{h}[h +1 - |m|]=(h +1)^2$$

Since the weights must sum up to unity, the suggested kernel is

$$k(\omega_{j+m},\ \omega_j)= \frac{h+1-|m|}{(h+1)^2} \qquad (15.5.2)$$

and the estimator (15.5.2) becomes

$$\hat{s}_{y_Y}\,(\omega_j)= \sum_{m=-h}^{h}[\frac{h+1-|m|}{(h+1)^2}].\ \hat{s}_y\,(\omega_{j+m}) \qquad (15.5.3)$$

For instance, if ***h***=2, the estimation by (15.5.2), with ***m***=4, 5, 6, leads to

$$\hat{s}_{y_Y}\,(\omega_j)= \frac{1}{9}\,\hat{s}_y\,(\omega_{j-2}) +\frac{2}{9}\,\hat{s}_y\,(\omega_{j-1})+ \frac{3}{9}\,\hat{s}_y\,(\omega_j)+ \frac{2}{9}\,\hat{s}_y\,(\omega_{j+1})+ \frac{1}{9}\,\hat{s}_y\,(\omega_{j+2}).$$

We learned that periodogram is asymptotically unbiased but has a large variance. If one employs an estimate based on averaging the periodogram at different frequencies, that will reduce the variance but at the cost of some bias; the severity of the bias depends in part on the bandwidth. Nonparametric results are shown to be sensitive to the choice of ***h***. Taking a wider "window", or bandwidth introduces larger bias; therefore, the choice of ***h*** should consider both efficiency and bias; this is sometimes called the *variance-bias trade-off*. The variance smoothing procedure employed in nonparametric spectrum estimation will tend to lower peaks and troughs. The recommended guide is to plot a spectrum estimate using different bandwidths and then select the most plausible by subjective judgment.

15.6 *Semi-parametric periodogram estimators*

Chapter 9 outlined the parametric ML estimator for a ARFIMA model that requires the estimation of the full model, that is of both the short-run and the long-run lag parameters using $\hat{d}$ estimate of the fractional differencing long-run parameter of a time-series. We also noted the ARFIMA process is a square-summable with a d <1 fractional order of integration and autocovariances that decays hyperbolically compared to the ARIMA process with d=0 or d=1 order of interation and absolutely-summable autocovrainces that decays expentntially or geormeterically. The absolute summability implies square-summability, but the reverse is not true, and the difference in the slow dying out of the lag effects is what accounts for long memory of the ARFIMA process. If the short memory lags are relatively marginal and the main interest in a time-series analysis is in estimating the long memory fractional parameter d, we can estimate the long-run dynamic effects of the series *without* specifying the full data generating process by the estimation of semi-parametric periodogram. Here we examine three semi-parametric estimators commonly employed in regression estimation of the periodogram.

The GPH test, see Geweke and Porter-Hudak (1983), employs a semiparametric log periodogram regression to evaluate d, estimated to fall in the [- 0.5, 0.5] interval, without any specification of the short-run structure. The GPH estimate of the fractional memory d of series using X_t from

$$(1 - L)^d X_t = \varepsilon_t$$

where $\varepsilon_t$is mean zero stationary and continuous spectral density $f_\varepsilon(\lambda)>0$. The $\hat{d}$ *is computed over the fundamental frequencies* to obtain the long run factional parameter by application of the least squares

$$\log(l_x(\lambda_s)= \hat{c} + \hat{d} \log|1 - e^{i\lambda_s}|^2 + \nu_x \quad (15.6.1)$$

with the fundamental frequencies { $\lambda_s=\frac{2\pi s}{n}$, S=1, . . . ,m, m< n}. We define the discrete Fourier transform (DFT) and the periodogram as

$$\omega_x(\lambda_s)=\frac{1}{\sqrt{2\pi n}}\sum_{t=1}^{n} X_t e^{it\lambda_s} \quad (15.6.2)$$

$$l_x(\lambda_s)= \omega_x(\lambda_s)\ \omega_x(\lambda_s)^* \quad (15.6.3)$$

where * indicate differencing and x_t= $\log\left|1 - e^{i\lambda_s}\right|$, and m is the number of regression Fourier frequencies. The least squares estimate results in

$$\hat{d} = 0.5\frac{\sum_{s=1}^{m} x_s l_x(\lambda_s)}{\sum_{s=1}^{m} x_s^2} \quad (15.6.4)$$

The semiparametric regression slopes are the estimates of the series power spectrum in the neighborhood close to zero frequency. If only the band contains a few ordinates, the slope is based on a small sample, affecting accuracy; if the band contains too many observations, the medium and high-frequency cycles of the spectrum will contaminate the estimate. Though the GPH is usually based on power=0.5, its robustness is checked at 0.40-0.75 range of power values.

Phillips (1999) notes that the GPH test does not include the case of d=1 in the range of its power values, and it is inconsistent when d >1, with a tendency to be asymptotically biased toward unity. The paper proposed to modify the GPH estimate of $\hat{d}$ to take account of the distribution of d with d=1 as the null hypothesis. The modification is based on an exact representation of the DFT in the unit-root case. Phillips shows the distribution of modified $\tilde{d}$ follows

$$\sqrt{m}(\tilde{d} - d) \rightarrow_d N(0,\frac{\pi^2}{24})$$

In the limit, $\hat{d}$=d=1 is consistent for values of d around unity, as with the GPH in the stationary case. With this modification, a semiparametric unit-root test against a fractional alternative is obtained from

$$Z_d = \frac{\sqrt{m}\,(\tilde{d}-1)}{\pi/\sqrt{24}}$$

using the standard normal distribution critical values, consistent against both $d< 1$ and $d> 1$ alternatives.

Another fractionally integrated estimator that generally tends to perform better that the above is the Robinson (1995) multivariate semi-parametric estimate of the long memory parameter $d(g)$ for a set of G time-series, $y(g)$, g=1, and G with $G \geq 1$. The Robinson estimator obtains the $d(g)$ parameter estimate for each time-series from a single log-periodogram regression that allows different slope and intercept for each series. This formulation permits a multivariate model to test if different time series have a common differencing parameter, though the model is also applicable to a single time-series; the estimator also allows for a sizable fractioning of the original sample size.

Let X_{gt} stand for a G-dimentional vector for g=1, . . . , G, then the Robinson periodogram of X_{gt} is presented as

$$l_x(\lambda_s)= (2\pi n)^{-1}| \textstyle\sum_{t=1}^{n} e^{i\lambda_s}|^2, \text{ g=1, . . . , G.} \qquad (15.6.5)$$

The restriction that some of the dg are equal can be derived from the least squares estimated covariance matrix of the coefficients. Exercise 15.7 for an application of the three estimators.

15.7 *Some Applications of Spectral Analysis*

Example 1*-Alaska air temperature*. Figure 15.3 shows examples of three time series with different cyclical behavior; time-series on the left, their spectra on the right. These are typical of common empirical cyclical time-series, they are intended, together with their corresponding spectra, as examples to aid interpretation of an estimated spectrum, and decide when the application of spectral analysis is likely to be most helpful. In the top panel, the series shows cyclical behavior over different periods, namely, peaks for a period around 60, and also fluctuations over the entire period, indicating the spectrum is concentrated on different frequencies. In the middle panel, the series has more oscillations than the top series, and the largest variation, around the period (1, 50), never re~~oc~~curs in the subsequent periods. The latter dominates the series, an indication the spectrum has more concentration on the lower frequency, a pattern also examined further in two examples below. In the bottom panel, the series demonstrates a strong seasonal pattern. Its spectrum on the right shows two modes; this series is an aggregation over two or more frequencies.

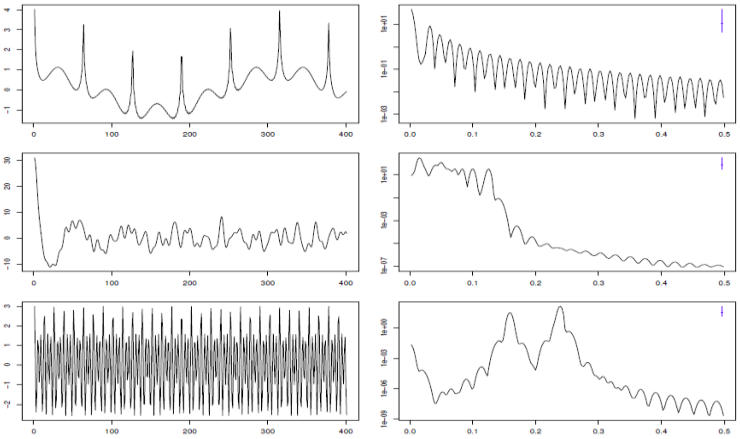


Figure 15.3 *three time-series (right) and their spectra (right)*

Figure 15.4 shows an example of air temperature series at Alaska. As expected, from the time plot of this series, it is obvious that the seasonal effects are dominant and such a deterministic component account for a very large amount of the total variation. The top panel shows a large peak at frequency of one cycle per year, and that spectrum analysis is really unnecessary with such clear pattern of seasonality. If a series contain strong trend or seasonality, then it is good practice to remove the variation from the data before the application of spectral analysis. The middle panel the Alaska air spectrum with the seasonal variation removed; now the variance is concentrated at low frequencies, indicating either a trend or short-term correlation as in a first-order positive AR process. A trend effect such a global warming is relatively small compared to other effects; hence the AR process seems more likely for this series. However, the corresponding periodogram in the bottom shows a pattern of very quick oscillations up and down that is not helpful in interpreting the properties of the data. This is evidence that the periodogram should be smoothed to obtain consistent spectrum estimation. Removing trend and seasonality is a simple example of *prewhitening*, that is , of construction a series closer to one with white noise error, an attempt to have a linear transformation of the series in order to have a relatively flat spectrum that is easier to estimate than one sharp peaks and troughs.

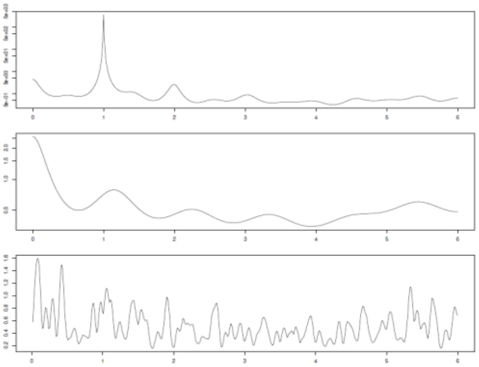


Figure 15.4 *Spectra of Average monthly air temperature in Alaska: top raw data, middle seasonally adjusted data; bottom the periodogram of the seasonally adjusted data.*

Example 2- *US Manufacturing output time-series*. Figure 15.5 (F.5-F.8) shows a plot of the Fed's from Jan. 1947 to Nov. 1989 seasonally unadjusted monthly index. The Us economy experienced recession in 1949, 1954, 1958, 1960, 1970, 1974, 1980, and 1982 roughly as a year-long periods of contraction in production; the data also shows strong seasonality; for example, production always falls in July and rises in August. The sample periodogram is shown in **Fig. 6,** this is a display of $\hat{s}_y$ (ω_j) as a function of j where ω_j=2 $\pi j/T$. The contribution to the sample variance of the lowest-frequency components (j near zero) is several times bigger that the contributions of economic recession or seasonal effects, as it is clear from the upward trend of the raw data in **Fig. 5** Moreover, the definition of the population spectrum by (15.1.4) assumes a covariance-stationary process, this is clearly not the case in **Fig. 5**. One possibility is to use differencing instead and analyze the monthly growth rate defined by

$$X_t=100.[\log(y_t) – \log(y_{t-1})] \qquad (15.7.1)$$

Figure 15.5:

F.5 US output 47:1-89:1 **F.6** Periodogram of F.5 **F.7** spectrum of F.5 **F.8** year-by-year F.7

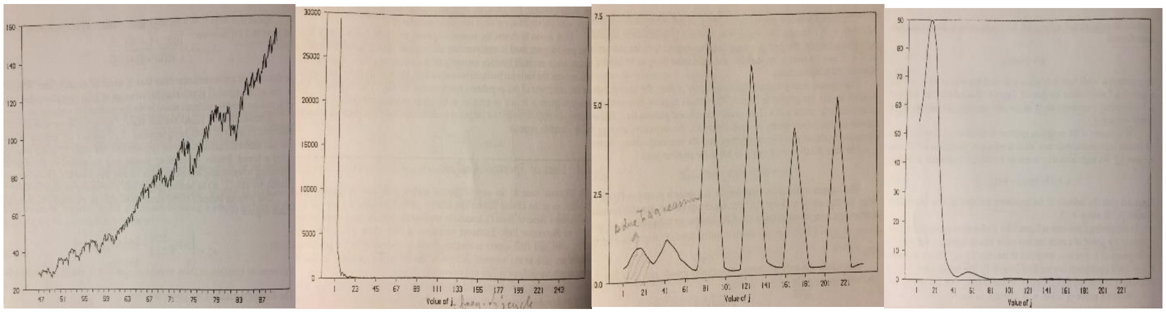


Fig. 6 suggests low-frequency components are the main determinants of the sample variance of ***y***. **Fig. 7** displays the population spectrum of ***X*** by eq. (15.5.3) using ***h***=12. The interpretation of **Fig. 7** plot is easier in terms of the period of a cyclic function rather than its frequency, i.e. the period for the frequency of a cyclic ω is $2\pi/\omega$. Thus, $\omega_j=2\pi j/T$which corresponds to a period of $2\pi/\omega_j=T/j$. Given the sample size of T=513 observations, and the appearance of the first peak in **Fig. 7** around ***j***=18, the corresponding cycle has a period of 513/18=28.5≈ 2.5 years and roughly corresponding to the 1949 recession. Such cycle effects are sometime called a *business cycle frequency* and the area under the hill describe how much of the variability in monthly growth is due to economic recession. The second peak in **Fig. 7** occurs at j=44 and corresponds to a period of 513/44=11.7 months, a 12-month cycle associated with seasonal effects. The subsequent four peaks with periods 6, 4, 3, and 2.4 months, respectively also appear to be picking up seasonal and other deterministic effects.

Since the US manufacturing usually falls temporary in July with negative growth rate and move in the opposite direction to rise gain in August with positive growth rates, the seasonality in tis case induces negative first-order serial correlation and other calendar effects in the series $\boldsymbol{x_t}$ estimated by (28) that may account for the high frequency peaks in **Fig. 7**. An alternative is to employ year-to-year growth rates in order to remove the calendar effects

$$X_t=100.[\log(y_t) - \log(y_{t-12})] \qquad (15.7.2)$$

The estimated sample spectrum for this series is plotted in **Fig. 8**. With this detrended series, all the variance left is attributed to components related to the business cycle frequencies.

Readings

For textbook discussion, see Hamilton (1994, chapter 6), Chatfield and Xing (2019, chapters 6 and 7). Ganger and Joyeux (1980) proposed fractional differencing for slow converging time series; Geweke et. al. (1983) developed the standard spectral regression model, Sowell (1992) and Robinson (1995) provide, respectively, parametric and semi-parametric spectral estimators.

Chapter 15 Spectral Analysis Exercises

Non-Empirical questions

Q15.1 According to De Moivre's theorem, given a particular $\boldsymbol{\omega} j$ frequency cycle,

e^{-iwj} = *cos(*$\boldsymbol{\omega}$*j) -i.sin(*$\boldsymbol{\omega}$*j)***.**

a.Provide a proof of this equality,

hint: Use Euler's formula $e^{i\theta}$=*con*$\boldsymbol{\theta}$+*i.sin*$\boldsymbol{\theta}$

Q15.2 Use (15.1.10)-(15.1.11) to drive the spectrum for

a. MA (1) process
b. AR (1) process
c. ARMA (p, q) process

Q15.3 Download *manemp2.dta*, the data are for US manufacture employment.

a. Plot the differenced series (mean set as -0.206), estimate AR(1) by arima, then obtain its spectral density and comment on its shape.

Q15.4_ Download *icsal.dta*, US unemployment insurance claims.

a. Plot differenced series, estimate AR(1)by *ARIMA* , then its spectral density and its plot, and comment on its shape.

Q15.5 Download *mloa.dta*, monthly carbon dioxide level data.

a. Estimate *ARIMA* nested in *ARFIMA* first, then compare that outcome with spectral density estimates for *SR* & *LR*, comments on the fractional parameter estimate.

Q15.6 Download *mumps2.dta*, series on number of mumps infections in NYC.

a. Fit *ARFIMA* model to test if 1st differencing is over-differenced.

Q15.7 Download *FTA.dta*, S&P 500 series.

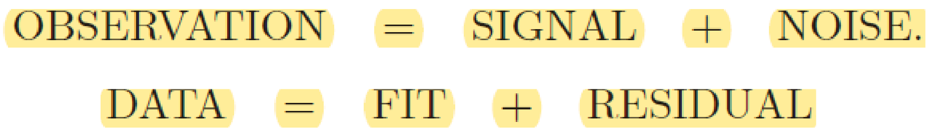
a. Estimate semiparametric log periodogram by GPH estimator, and by Phillips estimator to test for *d*=1.
b. Repeat the estimation by Robinson multivariant estimator
c. Compare the performance of the three estimators in a. & b. with regard to slow converging series.

Chapter 16 State-Space Models and Kalman Filter

Introduction

The State-Space models provide a flexible approach to many time-series models in economics and dynamic analysis of time-series in a single framework. They cover the basic least square regression and many dynamic models without restricting estimation by requirements such as stationarity, or limit application to process with long memory. Often the variable of interest may be some unobservable, or forecasting can only be made with stationary series. Many applications require more: employment of a long-memory model. The S-S approach offers algorithms that can make inference about the unobservable using estimations made by observable data without the need to restrict the estimator to be stationary, dependent on long-memory models. It is a flexible approach that presents many different time-series models based on the least squares as a special case.

Data for a time-series variable come with error; the true variable called *signal* is contaminated with *noise* due to measurement error or unobservable effects. **State-Space (S-S)** approach models attempt to obtain estimates and prediction of the true states of the variable from the observable changes of the time-series, that is by separating the series measurement of the *signal* from the noise. It assumes the signal is a *linear* combination of a set of unobservables, called **state variables,** that describes them in a $\boldsymbol{h_t}$ vector of states at time *t*. The noise can be due to measurement error in data, or from unobservable effects.



OBSERVATION = SIGNAL + NOISE.

DATA = FIT + RESIDUAL

An obvious S-S application would be modelling the consequences of measurement error. For example, in a study of the behavior of the *ex ant~~i~~e* real interest rate (nominal rate – expected inflation= $\boldsymbol{i_t} – \pi^e_t$), we have to deal with an unobservable state variable since anticipated inflation rate data are unavailable.

16.1 *Basics of Space-State*

Let the scalar state variable be $\xi_t = i_t – \pi^e_t - \mu$; where μ stands for the average *ex ant*e rate. Now assume an *AR*(1) rate equation as

$$\xi_{t+1}=\emptyset\,\xi_t+v_{t+1} \qquad (16.1.1)$$

Given the observations on the *ex-post* rate (nominal i_t – actual inflation π_t), we can write it as

$$i_t - \pi_t = (i_t - \pi^e_t) + (\pi^e_t - \pi_t) = \mu + \xi_t + w_t \qquad (16.1.2)$$

where inflation forecast error by economic agents is $w_t=(\pi^e_t - \pi_t)$. (16.1.2) is econometrically observable if agents forecast optimally, then w_t should be uncorrelated with its own lagged values and with the real *ex ante* interest rate. In this example (16.1.1) is the state equation and (16.1.2) the observation equation, and interest lies in obtaining prediction error for a forward estimate of the unobservable expected rate of inflation at *t* conditional on information up to *t*-1. At other times, we may be interested in using *all* measured data up to the current period to obtain an estimate of an unobservable effect backward, an event many decades ago. As an example, suppose an unobserved scalar C_t represents the state of the business cycle that affects *i* different observed macroeconomic variables with idiosyncratic components χ_{it} uncorrelated with the macro variables y_{it} that change with C_t. Then C_t and each χ_{it} described by univariate *AR*(1) processes form a

[(n+1) by 1] state vector as $\xi_t = [C_t \chi_{1t} \chi_{2t} \ldots \chi_{nt}]$'; the y_{it} observation equation parameter estimates in this model describe the sensitivity of the *i*th series to the business cycle at different points in the past affected by the unobservable state equations for the ξ_t vector, see section 16.7 and empirical exercise 16.5.

A *S-S* system always has one set of equations based on the observations and another set of equations for unobservable factors that affect the observations. A simple univariate S-S system consists of two equations, one describing the *state equation* (***S.E***), also known as the *transitional* equation, another the measurement equation (***M.E***), also known as the *observation* equation:

$$X_t = h'_t \Theta_t + \eta_t \qquad \eta \sim N(0, \sigma^2_\eta) \qquad M.E \qquad (16.1.3)$$

$$\Theta_t = G_t \Theta_{t-1} + \omega_t \qquad \omega \sim N(0, \sigma^2_\omega) \qquad S.E \qquad (16.1.4)$$

$\boldsymbol{X_t}$ in the M.E is defined as a function of the (*m* by 1) transposed $\boldsymbol{h'_t}$ vector of *m* unknown $\boldsymbol{\Theta_t}$ states, or a matrix of such states if more than 1 state in a series, namely, the trend and seasonal states. A state is usually assumed directly unobservable, but we assume we know how it changes over time, namely, seasonally, or as *AR* (1) process with the G_t matrix of state parameter estimates, although S-S still remains useful in obtaining time-series predictions even if the state vector is known. We further assume that the error term in (16.1.3), known as ***irregular*** variation of the M.E equation, has zero mean and homoscedastic variance; it is also normally distributed with no serial correlation and with a known coefficient G_t (m by m) matrix. The error term in (16.1.4) is multivariate normal with zero mean and a known variance-covariance matrix W_t, and no serial correlation, and no correlation between η_t & ω_t, $Co(\eta_t, \omega_t)=0$. We use the observations on X_t to make inferences about Θ_t by regression. We note that if ω_t is independent of Θ_t , Θ_{t-1} , Θ_{t-2} …, then the AR(1) process of (16.1.3) ensures that Θ_t depends on Θ_{t-1} but not on earlier values; that is, the $\{\Theta_t\}$ sequence has the Markov property[20] that employs only Θ_{t-1} to update the sequence. The model can be generalized to a vector of X_t by making h'_t a matrix of corresponding size. The states estimates are not constant over time, but change in a predictable manner, namely, with an increasing time-trend, or seasonally. S-S estimation allows both equations and error terms to change as long as we can assume that they have known distribution functions; when these are constant, then the S-S (16.1.3)-(16.1.4) model reduces to the linear OLS model. The S-S models are frequently employed because of the flexibility they offer in modeling a large class of time-series behavior by decomposing a time-series into its trend, seasonal and irregular components; application requires that the S-S decomposition be additive or, if multiplicative, must be expressed in logarithmic terms. More generally, the S-S observation equation must be a ***linear*** function of the state variables, without restricting the model to be constant over time, in order to allow local changes such as trend and seasonality to be estimated using the S.E equation. Next, we examine several common time-series models in S-S form to display its flexibility and advantages.

16.2 *State-Space Models of Time-series*

i. The random walk plus noise (RWN) model

Suppose the M.E., X_t is a function of a single unobservable random walk local level μ_t at *t*:

$$X_t = \mu_t + \eta_t \qquad \eta \sim N(0, \sigma^2_\eta) \qquad M.E \qquad (16.2.1)$$

$$\mu_t = \mu_{t-1} + \omega_t \qquad \omega \sim N(0, \sigma^2_\omega) \qquad S.E \qquad (16.2.2)$$

In this case, the state vector Θ_t, consists of a single unobserved level variable μ_t, Θ_t is a scalar, in addition h'_t is the vector of unkown states and G_t , the unobebservables states parameter matrix,

[20] The Markov chain property assumes that the probability distribution at time *t*+1depends only on the state of the system at *t*.

are also scalars both equal to unity; the important $\sigma^2_\omega / \sigma^2_\eta$, known as the **signal-to-noise ratio**, determines the properties of the S-S model. The model reduces to the constant-mean OLS because with $\boldsymbol{\sigma^2_\omega}$ =0; $\boldsymbol{\mu_t}$ becomes a constant. The RWN is called the **local level** or the **steady** model, and $\boldsymbol{\mu_t}$ is an equivalent "intercept" term that changes with *t*, analogous to the least squares intercept with the important difference of change over time. Taking the first differences of $\boldsymbol{X_t}$ to turn it into a stationary series, the simple (16.2.1) would have the same autocorrelation function as *MA* (1) and is thus an alternative to ARIMA (0, 1, 1) with a non-stationary component *q*=1. For an example of the RWM model (16.2.1), consider the level disturbances are all fixed on $\boldsymbol{\omega_t}$ =0 for *t*=1, . . . , *n*, then we have :

For *t*=1 $x_1=\mu_1+\eta_1$

$\mu_2=\mu_1+\omega_1=\mu_1+0=\mu_1$

t=2 $x_2=\mu_2+\eta_2=\mu_1+\eta_2$

$\mu_3=\mu_2+\omega_2=\mu_2+0=\mu_1$

t=3 $x_3=\mu_3+\eta_3=\mu_1+\eta_3$

$\mu_4=\mu_3+\omega_3=\mu_3+0=\mu_1$

and so on. Therefore, the local level model (16.2.1) simplifies to $x_1=\mu_1+\eta_{t,}$ with $\eta_t \sim NID$ (0, σ^2_η). Typically, the unobserved state variable at time *t*=1 is unknown. However, we can use an estimated value for it obtained, conditional on some initial values from ***t***=*i* earlier periods to start the iterations. Such starting values of the unknown state parameters and their variances are called *diffuse initialization*; in practice the initial diffuse values are set equal to the unconditional mean and variance.

i) Local level trend or Linear growth model

Suppose we expand (16.2.1) by a second unobservable variable, time changing *trend*, also referred to as a *drift* term; now we have a three-equation S-S model with two transitional equations as each unobservable has its own equatin:

$$\begin{array}{lll} X_t = \mu_t + \eta_t & \eta \sim N(0, \sigma^2_\eta) & \\ \mu_t = \mu_{t-1} + \beta_{t-1} + \omega_{t1} & \omega_1 \sim N(0, \sigma^2_{\omega 1}) & (16.2.3) \\ \beta_t = \beta_{t-1} + \omega_{t2} & \omega_2 \sim N(0, \sigma^2_{\omega 2}) & \end{array}$$

where the state vector is Θ'_t =(μ_t , β_t) as the local level and the local trend, with the latter excluded from the measurement equation; comparing to (16.1.1) and (16.1.2) shows in this case h'_t = (1, 0) and $G_t = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$, both constant over time. For example, if we fix all disturbances in ω_{t1}=0 and ω_{t2}=0, then we have

For t=1 $\quad$ $x_1=\mu_1+\eta_1$

$\mu_2=\mu_1+\boldsymbol{\beta}_1+\omega_{11}=\mu_1+\boldsymbol{\beta}_1+0=\mu_1+\boldsymbol{\beta}_1$

$\boldsymbol{\beta}_2=\beta_1+\omega_{12}=\beta_1+0=\beta_1$

t=2 $\quad$ $x_2=\mu_2+\eta_2=\mu_1+\boldsymbol{\beta}_1+\eta_2$

$\mu_3=\mu_2+\boldsymbol{\beta}_2+\omega_{21}=\mu_1+2\boldsymbol{\beta}_1+0=\mu_1+2\boldsymbol{\beta}_1$

$\beta_3=\beta_2+\omega_{22}=\beta_2+0=\beta_1$

t=3 $\quad$ $x_3=\mu_3+\eta_3=\mu_1+2\boldsymbol{\beta}_1+\eta_3$

$\mu_4=\mu_3+\boldsymbol{\beta}_3+\omega_{31}=\mu_1+3\boldsymbol{\beta}_1+0=\mu_1+3\boldsymbol{\beta}_1$

$\boldsymbol{\beta}_4=\beta_3+\omega_{32}=\beta_2+0=\beta_1$

and so on. Then, the linear trend model simplifies to $x_1=\mu_1+\boldsymbol{\beta}_1 g_t+\eta_t$, with $\eta_t \sim NID\,(0, \sigma^2_\eta)$ where the time predictor variable $g_t=t$ -1 and t=1, . . . , n, and μ_1 and $\boldsymbol{\beta}_1$ are the initial values of the level and the slope. If ω_{t1} and ω_{t2} have zero variances, then we have a deterministic *global* linear trend model based on the full system of equations (16.2.3), (not very likely); or a *local* linear model if the trend is allowed to change; or the trend is a constant but the local level changes.

ii) The *Local level trend with seasonality or the basic structural model.*

We can also incorporate seasonal effects in the (16.2.1)-(16.2.2) model; $\boldsymbol{S_t}$ denotes the number of seasons in a year; the full system of equations (16.2.4) has the state vector with $\boldsymbol{S}$+2 components and four additive error terms:

$$\begin{aligned} X_t &= \mu_t + S_{it} + \eta_t & \eta &\sim N(0, \sigma^2_\eta) \\ \mu_t &= \mu_{t-1} + \beta_{t-1} + \omega_{1,t} & \omega_1 &\sim N(0, \sigma^2_{\omega 1}) \\ \beta_t &= \beta_{t-1} + \omega_{2,t} & \omega_2 &\sim N(0, \sigma^2_{\omega 2}) \end{aligned} \qquad (16.2.4)$$

$$S_t = -\sum_{j=1}^{s-1} S_{t-j} + \omega_{t,3} \qquad \omega_2 \sim N(0, \sigma^2_{\omega 3})$$

where $\sigma^2_i = Var(\omega_{ti})$ for i=1, 2, 3 since the seasonal effects are over ***S***-1 periods. This model is called the *basic structural model*; it can be extended to incorporate explanatory variables.

The models examined so far are known as **time-invariable S-S models** because the coefficients change over time in a predictable manner. Let us now look at the models that allow coefficients to change randomly.

iii) S-S models with time-varying Coefficients

The measurement equation for this model is

$$X_t = \mu_t + \sum_{j=1}^{k} \beta_{tj} x_{tj} + \eta_t \tag{16.2.5}$$

For one predictor variable$\beta_t = \beta_{t1}$, the S-S model is in the form

$$\begin{aligned} X_t &= \mu_t + \beta_t x_t + \eta_t && \eta \sim N(0, \sigma^2_\eta) \\ \mu_t &= \mu_{t-1} + \omega_{t1} && \omega_1 \sim N(0, \sigma^2_{\omega 1}) \\ \beta_t &= \beta_{t-1} + \omega_{t2} && \omega_2 \sim N(0, \sigma^2_{\omega 2}) \end{aligned} \tag{16.2.6}$$

Setting $\boldsymbol{\omega_{t1}}$ & $\boldsymbol{\omega_{t2}}$ equal to zero, we have

For t=1 $\quad x_1 = \mu_1 + \boldsymbol{\beta}_1 x_1 + \eta_1$

$\mu_2 = \mu_1 + \omega_{11} = \mu_1 + 0 = \mu_1$

$\boldsymbol{\beta}_2 = \beta_1 + \omega_{12} = \beta_1 + 0 = \beta_1$

t=2 $\quad x_2 = \mu_2 + \boldsymbol{\beta}_2 x_2 + \eta_2 = \mu_1 + \boldsymbol{\beta}_1 x_2 + \eta_2$

$\mu_3 = \mu_2 + \boldsymbol{\beta}_2 + \omega_{21} = \mu_2 + 0 = \mu_1$

$\beta_3 = \beta_2 + \omega_{22} = \beta_2 + 0 = \beta_1$

t=3 $\quad x_3 = \mu_3 + \boldsymbol{\beta}_3 x_3 + \eta_3 = \mu_1 + \boldsymbol{\beta}_1 x_3 + \eta_3$

$\mu_4 = \mu_3 + \omega_{31} = \mu_3 + 0 = \mu_1$

$\boldsymbol{\beta}_4 = \beta_3 + \omega_{32} = \beta_3 + 0 = \beta_1$

and so on. Then, the linear time-varying coefficient linear simplifies to

$$x_1 = \mu_1 + \boldsymbol{\beta}_1 x_1 + \eta_t, \text{ with } \eta_t \sim NID\,(0, \sigma^2_\eta).$$

Therefore, the S-S models for all cases (16.2.3)-(16.2.6) redice to the least squares model. Of course, if the elements of the states' residuals are constant and the model reduces to the least squares, there will be no gain in S-S applications. The benefits from the S-S presentation come from covering a much larger class of models wth non-constant residuals of which the least squares is a very special case. The S-S models that allow the parameters to change are called the **time-variable State Space** models.

16.3 *Kalman Filter*

The Kalman (1960) filter comes from the field of control engineering employed in spacecraft and more recently in the technology used in Covid-19 to trace infection transmission. It is a ***recursive*** algorithm that can predict the motion of a body based only on its last observation. In the time-series models discussed so far, the state components are estimated using **all** past observations, [y_1, y_2, ..., y_{n-1}] to make a current period forcast, but obtaining the next period prediction of the time-series by its dynamics are based only on the past period ***t***-1 predicted MSE; this *a priori* estimation is called the **Kalman prediction**. However, the Kalman ***filtered state*** also estimates the state vector using all observations including the current period *t*. Thus, the estimation of the state vector is carried out by performing two passes through the data. First, a forward pass from *t*=1, . . , *n* using the ***Kalman filter*** is processed. This recursive method has a key Markov chain property that obtains optimal predicted values of the unobservable states of a time-series at time *t based solely on the t*-1 *past observed values* [y_1, y_2, ..., y_{t-1}], see chapter 17 for a more detailed discussion of Markov process. The backward pass from *t=n*, . . ., 1 uses a recursive algorithm known as ***state and disturbance smoothers*** applied to the output of the Kalman filter employing all measured observations, including the current period. The main purpose of state and disturbance smoothing is to obtain estimates for the values of state and disturbance vectors at time *t*, combining all available information. In the text below, we focus mainly on the forward Kalman filter algorithm; smoothing is briefly discussed at the end, see also empirical exercise Q 16.3.

In applications of the unobservable Kalman filter, it is the predicted values that are of special interest. Let $\boldsymbol{a_t}$ be the Kalman filter state of a series at time ***t***, then the central formula of the recursive Kalman filter updating process is:

$$a_{t+1} = a_t + K_t\,(y_t - z'_t a_t)+v_{t+1} \qquad (16.3.1)$$

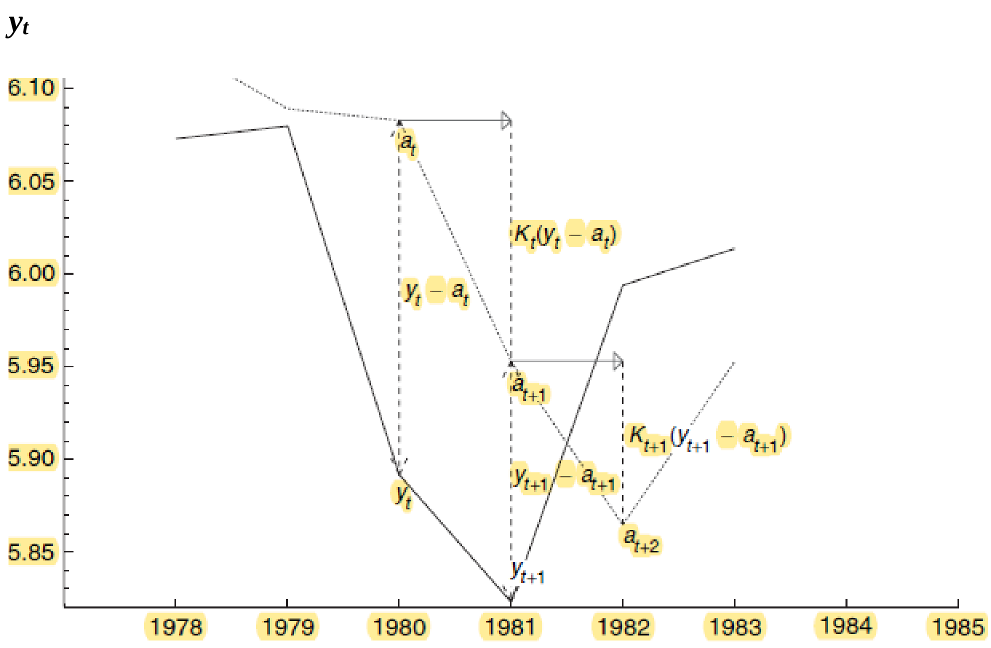
where z_t is the state vector that when applied to the single state local level model simplifies to

$$a_{t+1} = a_t + K_t\,(y_t - a_t)+v_{t+1} \qquad (16.3.2)$$

Because the Kalman forecasting procedure relies on the previous step estimates, it can also be viewed as a Bayesian forecasting method that updates the (~~*a*~~ *prior*) probability distribution of θ_t as a new observation becomes available to provide a revised (*posterior*) distribution.

Figure 16.1 illustrates how (16.3.2) works, by singling out the last three predicted values for a hypothetical series. Here the observed and filtered series only contain the level (residual) observations over 1978-83.

Figure 16.1



At time t=1980, the current value of the filtered level $\boldsymbol{a_{1980}}$ is based on all past observations

[y_{1970}, y_{1971}, ..., y_{1979}]. If the current value $\boldsymbol{y_t}$ is unknown, for instance, missing, then its best available prediction is given by simply moving the filtered state forward unchanged (horizontally by a unit (1981-1980), that is the best prediction of the filtered state at time (t+1) is $a_{t+1} = a_t$ since $a_{1981} = a_{1980}$. However, given a known value for y_{1980}, the Kalman process feeds this information into the Kalman filter (16.3.2) and the discrepancy (y_t - a_t) (the vertical arrow in Fig. 3.1) is used to update the estimate for a_{1980} to produce $a_{t+1} = a_{1981}$. In other words, the 1981 prediction is *qualified* by the discrepancy (y_t - a_t); in this case, the negative discrepancy (y_t - a_t) < 0 results in a *decrease* in the filtered level. The next step of (16.3.2) similarly moves the predicted value for $a_{t+1} = a_{1982}$ forward unchanged if the current value is unknown, horizontally by one unit of time, but if the discrepancy between y_{1981} and y_{1982} is known (the vertical arrow in Fig. 3.1), then the filter further modifies the step-one ahead prediction to obtain a new predicted update for the series; in this step (y_{1982} - a_{1982})>0, so the curve turns upward. Because each predicted state is treated as the *true* value to be updated by accounting for new information as it becomes available, the last prediction error, already containing the last rounds of correction, will thus be smaller than the previous two steps, resulting in fast stabilization of predicted values. As the update at t+1 is based on the discrepancy at t, Figure 16.1 demonstrates that the update at $\boldsymbol{a_{t+1}}$ *always lags* $\boldsymbol{a_t}$ by one observation. We call $\boldsymbol{v_t}$= ($\boldsymbol{y_t}$ - $\boldsymbol{a_t}$) *one-step-ahead* ***prediction errors***, also denoted as ***innovations*** since they add new information to the process of updating the prediction.

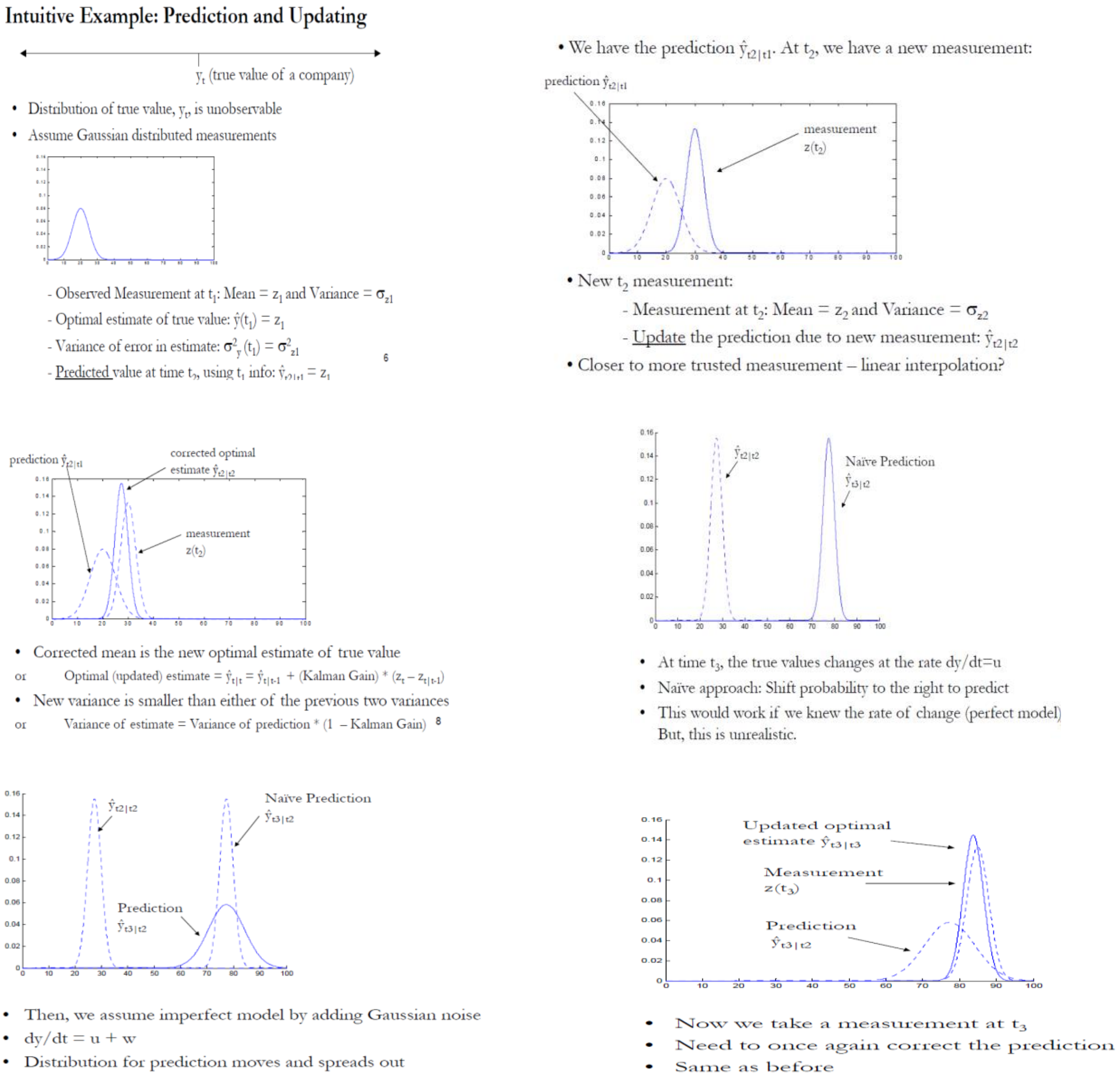
The important factor in the Kalman filter process is the value of $\boldsymbol{K_t}$ in (16.3.2), the local level scalar in this case. The value of $\boldsymbol{K_t}$ determines how much the value of $\boldsymbol{v_t}$ is allowed to affect the state estimate at time(t+1), therefore, the larger $\boldsymbol{K_t}$ at t, the greater $\boldsymbol{v_t}$ impact at (t+1). $\boldsymbol{K_t}$ is called the Kalman ***gain***; in effect it weighs the uncertainty of the state based on the past observations [y_1, y_2, …, y_{t-1}] relative to that in the new observations y_t. If the measurement uncertainty is large and the estimate uncertainty is low, that is, variance of η_t in (16.1.3) relative to variance of ω_t in (16.1.2), then the value of $\mathbf{K_t}$ will trend to zero, namely, a bigger weight to the estimate and a smaller one to the measurement, while, if the reverse is the case, the value of $\mathbf{K_t}$ will trend to one, preventing y_t big impact on the next period's state. The Kalman gain varies between 0 and 1; the Kalman gain of 0.5 indicates that the two types of uncertainties have equal weights. In the simple example of Figure 3.1, value of $\mathbf{K_t=P_t/F_t}$ where $\mathbf{P_t}$ is the filter state estimated error variance. and $\mathbf{F_t}$, the variance of one-step-ahead prediction error $\mathbf{v_t}$, though in general, the denominator consists of both variances, see below. The prediction error variances, **PEV**, are monotonically decreasing with time and converge quickly, especially. with time-invariant models, namely, those without explanatory variables; thereby simplifying the Kalman filter computations in a RWN steady S-S model. Recall that forecasting is impossible with either AR or MA processes unless they are stationary while the steady S-S model forecast are based on the random walk unit-root time-series; the forecast values dependen only on the last period information, and thereffor are valid even with a MA short-memory process. Otherwise, the Kalman PEV can still provide useful forecast analysis of the dynamics of a time-variable S-S model~~s~~ if the distribution function of a changing variable is known.

The important feature of the Kalman filter is its rapid stabilization. The forecasts quickly converge to the series' true values because of decreasing Kalman filter predicted error variance in successive rounds; this is demonstrated in the following graphs in Figure 16.2. From the visual explanation, we see a simple process starting from the initial conditions used to model (say, a constant rate of change) to make a prediction, then take a measurement to learn of the forecast error, then update the prediction by 'blending prediction and residual to finally obtain an optimal estimate with smaller variance.

At the outset (first left box), the initial predicted value and variance are usually set equal to the unconditional mean and variance; the next step treats these as the true value; the forecast error obtained from the difference between the true values and that of the linear least squares one-step ahead forecast based on the minimization of MSE, that is ($\boldsymbol{z_1}$-$\boldsymbol{z_2}$), is then used to correct the forecast in step two (second right box). The third and sebsequent steps treat the last corrected forecast as the true value and corrects the estimated forecasts the difference ($\boldsymbol{z_t}$-$\boldsymbol{z_{t-1}}$)as shown in (third left box). Therefore, the mean and variance at the third step ~~with~~ will be smaller tha~~t~~n those from both of the previous steps; explaining why the Kalman filter quickly stablizes by converging on its long run path.

In order to emphasize the predictive error ability of the Kalman filter Markov chain property, Figure 16.3 illustrates an empirical application of the local level (random walk unit-root) S-S model of the Kalman filter process to a time-series of annual road fatalities in Norway over 1970-2005.; the top panel are the predicted errors $\boldsymbol{v_t}$, while the bottom panel shows their variances $\boldsymbol{F_t}$. The important point in both Figures 3.1 and 3.2 is that while one-step-ahead forecast are based on all data t=1, 2, . . . , T, the forecast updating employs only the last t-1 information based on the Kalman Markov chain property.

Figure 16.2



Intuitive Example: Prediction and Updating

y_t (true value of a company)

- Distribution of true value, y_t, is unobservable
- Assume Gaussian distributed measurements

- Observed Measurement at t_1: Mean = z_1 and Variance = σ_{z1}
- Optimal estimate of true value: $\hat{y}(t_1) = z_1$
- Variance of error in estimate: $\sigma^2_y(t_1) = \sigma^2_{z1}$
- Predicted value at time t_2, using t_1 info: $\hat{y}_{t2|t1} = z_1$

6

- We have the prediction $\hat{y}_{t2|t1}$. At t_2, we have a new measurement:

- New t_2 measurement:
  - Measurement at t_2: Mean = z_2 and Variance = σ_{z2}
  - Update the prediction due to new measurement: $\hat{y}_{t2|t2}$
- Closer to more trusted measurement – linear interpolation?

- Corrected mean is the new optimal estimate of true value

or Optimal (updated) estimate = $\hat{y}_{t|t} = \hat{y}_{t|t-1}$ + (Kalman Gain) * $(z_t - z_{t|t-1})$

- New variance is smaller than either of the previous two variances

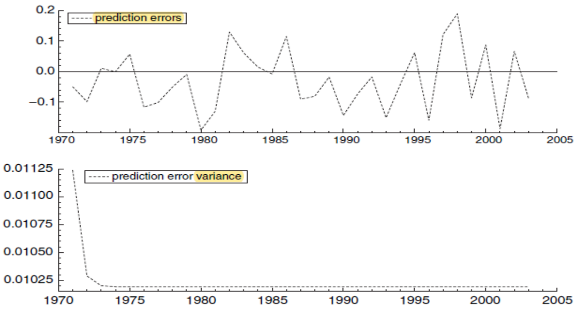
or Variance of estimate = Variance of prediction * (1 – Kalman Gain) 8

- At time t_3, the true values changes at the rate dy/dt=u
- Naïve approach: Shift probability to the right to predict
- This would work if we knew the rate of change (perfect model)
  But, this is unrealistic.

- Then, we assume imperfect model by adding Gaussian noise
- dy/dt = u + w
- Distribution for prediction moves and spreads out

- Now we take a measurement at t_3
- Need to once again correct the prediction
- Same as before

Figure 16.3



16.4 *The Kalman Filter implementation*

The implementation of the Kalman process involves three stages: the *initialization* stage, the *prediction* stage and the *updating* stage. Since the state vector values are unknown, some initial values must be provided to start the iterated estimation process, either from previous studies, or by using some of the starting values of T=1, 2, …, t. In practice, the initial state values are assumed normally distributed with the mean and variance usually set equal to the unconditional moments of the variables. We assume their disturbances are uncorrelated with the initial state variables and also uncorrelated with each other.

Next comes the prediction stage. Suppose we have $\boldsymbol{\hat{\theta}_{t-1}}$ as the 'best' linear estimate of the state variable $\boldsymbol{\theta_{t-1}}$ of a univariate time series up to time ($\boldsymbol{t}$-1); the 'best' estimate in this context means the minimum least squares *mean square error estimate of* (***MSE***) at time t [21]; we have also the estimator's (m by m) variance-covariance matrix, denoted by $\boldsymbol{P_{t-1}}$. The Kalman filter first prediction stage relates to forecasting $\boldsymbol{\theta_t}$ from data up to ($\boldsymbol{t}$-1). Using equation (16.1.4), and given $\boldsymbol{\omega_t}$ is unknown at time (t-1), the obvious estimator θ_t and its variance-covariance P_t are:

$$\hat{\theta}_t \mid {}_{t-1} = G_t \ \hat{\theta}_{t-1} \qquad (16.4.1)$$

$$P_t|_{t-1} = G_t P_{t-1} G^T_t + W_t \qquad (16.4.2)$$

(16.4.1)-(16.4.2) are the Kalman *prediction equations*. The estimator modifies the estimates as the new information on X_t at time t become available. Since the best estimator of X_t at time (t-1) for a vector of unkown states, is given by $h'_t \, \hat{\theta}_t|_{t-1}$, the prediction error at t is given by:

$$v_t = X_t - h^T_t \, \hat{\theta}_t \mid {}_{t-1} \qquad (16.4.3)$$

We can then use v_t to update, or qualify the prediction equations estimates of θ_t & P_t.

The following equations represent the next updating stage:

$$\hat{\theta}_{t-1} = \hat{\theta}_t \mid {}_{t-1} + K_t \, v_t \qquad (16.4.4)$$

(16.4.4) expresses the optimal estimated forecast as (forecast – K times forecast error), see first line of Figure 2, third left panel. The forecast variance is then

[21] Recall from chapter 6 that the least squares estimate of *MSE* is an estimate of an in-sample, one-step ahead forecast error.

$$P_t = P_{t|t-1} - K_t\, h'_t\, P_{t|t-1} \qquad (16.4.5)$$

(16.4.5) measures the variance of the forecast estimate equal to (variance of forecast prediction – K*variance of forecast prediction), see second line of Figure. 2, third left panel. These variances are measures of the estimated and measurement uncertainties discussed earlier. K_t is a weighting scheme called the ***Kalman gain matrix***. Based on variance as a measure of uncertainty, Kalman gain compares predicted and measured uncertainties based on their calculated averages by

K=*variance of predicted state error/(variance of predicted state error+ variance of measurement estimate*), or

$$K_t = P_{t|t-1}\, h_t / [\, h^T_t\, P_{t|t-1}\, h_t + \sigma^2_\eta\,] \qquad (16.4.6)$$

Since optimal forecasts are usually obtained from the minimization of *MSE*, but since the latter is based on variance, (16.4.6) is also expressed in terms of the *MSE* of state and measurement predictions. (16.4.4)-(16.4.6) are the Kalman *updating equations*; estimation of the gain matrix is the most important part of the Kalman filter procedure.

The Kalman stages can be generalized to consist of multiple state and measurement equations and include observable explanatory variables. We first state the *S-S* representation by the state vector, and state and measurement equations with the above assumptions required to obtain consistent estimates, and then develop optimal forecasts by linear least squares regression.

16.5 *Representation of State-Space Equations*

A *S-S* system are represented by two sets of equations

$$\xi_{t+1} = \mathrm{F}\, \xi_t + v_{t+1} \qquad (16.5.1)$$

$$y_t = \mathrm{A'x_t} + \mathrm{H'}\xi_t + \mathrm{w_t} \qquad (16.5.2)$$

Assumptions to ensure consistent linear estimates with optimal MSE: the disturbances are white noise and uncorrelated at all time lags; the initial state vector is uncorrelated with the disturbances:

$$E(v_t\, v'_\tau) = \mathrm{Q} \text{ for } t=\tau\text{; 0 otherwise,}$$

$$E(\omega_t\, \omega'_\tau) = \mathrm{R} \text{ for } t=\tau\text{; 0 otherwise;}$$

$$E(\, v_t\, \omega\, '_\tau) = 0 \text{ for all } t\ \&\ \tau$$

These assumptions further imply that:

$$E(\,v_t\ \xi\,'_\tau) = 0 \text{ for all } t=1, 2, \ldots, T$$

$$E(\,\omega_t\ \xi\,'_\tau) = 0 \text{ for all } t=1, 2. \ldots, T$$

where F, A$^{/}$, H$^{/}$ are parameter matrices of dimensions (*r* by *r*), (*n* by *k*), and (*n* by *r*) with *n*=vector of observed variables, **x$_t$** is a ($\boldsymbol{k}$ by 1) vector of exogenous and predetermined variables (~~hence~~ therefore, can include lagged $\boldsymbol{y_t}$), $\boldsymbol{r}$=number of state variables. We can also write the state equation (5.1) in terms of $\boldsymbol{v_{t-i}}$ by backward substitution as:

$$\xi_t = \mathrm{v}_t + \mathrm{F}\ \mathrm{v}_{t-1} + \mathrm{F}^2\ \mathrm{v}_{t-2} + \ldots + \mathrm{F}^{t-2}\ \mathrm{v}_2 + \mathrm{F}^{t-1}\ \xi_t \quad \text{for } t=2, 3, \ldots, T.$$

For example, (16.5.1)-(16.5.2) applied to the (16.1.1) S.E and (16.1.2) M.E modeling the rate of interest behavior, we have $r=n=1$, F=φ, $y_t=i_t - \pi_t$, A$^{/}$x$_t$ = μ , H=1 and $w_t = (\pi^e_t - \pi_t)$.

We summarize the Kalman filter forecasting steps:

a. For the initial iteration we obtain unconditional mean and variance of $\boldsymbol{\xi_1}$ from $\xi_1|_0$=E (ξ_1). The corresponding MSE is

$$\mathrm{P}_{1}|_0 = \mathrm{E}\{\ [\ \xi_1 - \mathrm{E}\ (\xi_1)]\ [\ \xi_1 - \mathrm{E}\ (\xi_1)]'\ \}$$

b. We then iterate on and update the MSE P$_t|_t$ by P$_t|_t$ = P$_t|_{t-1}$ – P$_t|_{t-1}$H(H$^{/}$P$_t|_{t-1}$H+R)$^{-1}$ H$^{/}$P$_t|_{t-1}$ and obtain the forecast using all information up to period *t* from

$$\hat{\xi}_{t+1|t} = \hat{\xi}_{t|t-1} + \{\ [\mathrm{FP}_{t|t-1}\mathrm{H}(\mathrm{H}^{/}\mathrm{P}_{t|t-1}\mathrm{H+R})^{-1}\ ]\ (\mathrm{y}_t - \mathrm{A}^{/}\mathrm{x}_t - \mathrm{H}^{/}\ \hat{\xi}_{t|t-1})\ \} \qquad (16.5.3)$$

where $\boldsymbol{\hat{\xi}_{t+1|t}}$ stands for the best linear forecast of the unobservable state, **P$_{t|t-1}$** provides the MSE of this forecast, $\boldsymbol{R}=E(\omega_t\,\omega'_\tau)$ stands for the (*r* by *r*) matrix of the ME error terms, and the expression inside the squared brackets expresses the Kalman gain of the forecast filter $\boldsymbol{K}$ above; and since the last term in squared brackets is the prediction error of (16.5.2), $\boldsymbol{K}$ acts to weight minimize the forecast error, that is, the difference between the two sides of (16.5.2). The forecasts of $\boldsymbol{y_t}$ and its *MSE* of (16.5.2) are given by:

$$\hat{y}_{t|t-1} = \hat{E}(\,y_t|x_t, \mathbf{Y}_{t-1})$$

As the vector of states from the S.E, $\hat{\xi}_{t|t-1}=(\boldsymbol{\xi_t}\ |\mathbf{Y}_{t-1})$, is a part of the M.E:

$$\hat{E}(\, y_t | x_t,\, \boldsymbol{\xi}_t) = \mathrm{A}^{/}\mathrm{x}_\mathrm{t} + \mathrm{H}^{/}\, \hat{\xi}_t \qquad (16.5.4)$$

and given that the cross product terms, $E[w_t(\boldsymbol{\xi}_t - \boldsymbol{\xi}_{t\,\backslash\, t\text{-}1})^{/}]=0$, drop out, covariance of y_t simplifies to

$$E[(y_t \, - \hat{y}_{t|t\text{-}1})\, (y_t - \hat{y}_{t|t\text{-}1}))^{/}\,] = (\mathrm{H}^{/}\mathrm{P}_{\mathrm{t|t\text{-}1}}\mathrm{H}+\mathrm{R}) \qquad (16.5.5)$$

where $\mathbf{Y}_t = (y'_{t,}\; y'_{t-1}, \ldots, y'_1;\, x'_{t,}\; x'_{t-1}, \ldots, x'_1)$; See the appendix for more details.

Despite its apparent complication, the general *S-S* form reduces to very simple equations in some special cases. For example, take the Kalman filter for the steady-state *RWN* model that has just one state variable $\boldsymbol{\theta_t}$ and the current level $\boldsymbol{\mu_t}$. It can be shown that as $\boldsymbol{t} \to \infty$ $\boldsymbol{P_t} \to$ *constant*, the Kalman filter form above reduces to the simple relationship

$$\hat{\mu}_t = \hat{\mu}_{t-1} + \alpha\, v_t \qquad (16.5.6)$$

where, in this particular case of the *RWN*, the smoothing constant $\boldsymbol{\alpha}$ becomes some function of the signal-to-noise ratio $\boldsymbol{\sigma^2_\omega / \sigma^2_\eta}$ (see exercise 16.1); not to be confused with the Kalman gain ratio whose denominator consists of state *and* observable variances. (16.5.6) is in fact *simple exponential smoothing*. When $\boldsymbol{\sigma^2_\omega}$ approaches zero and $\boldsymbol{\mu_t}$ becomes a constant, $\boldsymbol{\alpha}$ tends toward zero, as expected; while if $\boldsymbol{\sigma^2_\omega / \sigma^2_\eta}$ becomes large, then $\boldsymbol{\alpha}$ tends toward one. However, in general, the predicted errors are weighted by the Kalman gain matrix $\boldsymbol{K_t}$; this is shown by a second example, the linear regression model for the *S-S* time-varying coefficients in (16.2.4)-(16.2.5) model. In this case, the vector $\boldsymbol{W_t}$ in (16.5.2), or $\boldsymbol{\omega_t}$ in (16.2.5), is zero, and the regression coefficients are constant, $\boldsymbol{G_t}$ becomes an identity matrix and $\boldsymbol{P_{t|t\text{-}1}} = \boldsymbol{P_{t\text{-}1}}$. Then, based on their *unconditional* values, the predicted errors and the Kalman filter prediction error equations above reduce to

$$v_t = X_t - h'_t\, \hat{\theta}_{t-1} \qquad (16.5.7)$$

There are major practical advantages to the Kalman filter. First, its calculations are recursive, and based on the whole history of a time-series; while its prediction employ all past information up to and including period t=1, 2, . . .,n-1, the updating of that prediction is based on the previous period error forecast t-1 and therefore requires only the latest t-1 observation because of its Markov chain transition property. The process does not require long memory, nor is costrained by a short-memory processes; in that regard it compares favorably with *AR*, *MA* or *ARIMA* processes. Secondly, the models in S-S form provide~~s~~ forecasts without imposing stationarity restrictions on the series; for example, it can provide forecasts with the unit-root, local level *RWN*, by contrast, we cannot obtain linear forecasts for *AR* or *MA*, *or ARIMA* models without stationarity. Third, the filter quickly changes over time because each corrected forecast is treated in turn as a new, true value, whose next period prediction produces a forecast error, therefore variance correction is applied to already variance corrected forecasts. Finally, the procedure can handle missing data since the best forecast for a missing value is its own last lagged value.

16.6 *Maximum Likelihood Parameter Estimation*

Based on Gaussian errors for the S-S equations errors, the conditional distribution of $\boldsymbol{y_t}$ is given by

$$y_t|x_t, y_{t-1}\ N \sim[\mathrm{A}^{/}x_t + \mathrm{H}^{/}\ \hat{\xi}_{\ t-1}), (\mathrm{H}^{/}\mathrm{P}_{t|t-1}\mathrm{H}+\mathrm{R})^{-1}] \qquad (16.6.1)$$

A sample log likelihood can be constructed from (16.6.1). The *value of log-likelihood is maximized in state space methods by simultaneously minimizing the S.E prediction errors* $\boldsymbol{v_t}$*and the M.E variance* $\boldsymbol{F_t}$, unlike classical regression that minimizes the observation errors or disturbances $\boldsymbol{s_t}$ and their variances**,** $\boldsymbol{\sigma^2_\eta}$.

Once the prediction errors $\boldsymbol{v_t}$ and their variance $\boldsymbol{F_t}$ are obtained, they play a key role in the parameter estimation of the State- Space system. The Kalman filter was formulated in terms of **linear** projections, consequently, their forecasts are optimal in linear models; however, if the initial state and the innovations are multivariate Gaussian, then the filter forecasts are optimal among *any* functions of $\boldsymbol{y_t}$ & $\mathbf{x_t}$. The univariate state space models for the(initial) **diffuse log-likelihood** are defined ~~as~~ by:

$$\log L_d = -\frac{n}{2}\log(2\pi) - \frac{1}{2}\sum_{t=d+1}^{n}(logF_t + \frac{v_t}{F_t}) \qquad (16.6.2)$$

where *d* is the number of diffuse initial elements of the state. The multivariate space state log-likelihood function is written as, see Commandeur and Koopman(2007, p.89):

$$\ell = -\frac{Tn}{2}\log(2\pi) - \frac{T}{2}\sum_{t=d+1}^{T}(log|F_t^{-1}| - \frac{1}{2}e_t^{\prime}F_t^{-1}e_t) \qquad (16.6.3)$$

Under normality, (16.6.3) is *optimal in an MSE sense*, after initialization and ignoring the first set of observations. This representation of the likelihood is particularly convenient for estimating regressions involving moving average terms.

The Kalman filter applies to state space models that are linear in parameters. In many time series such as multivariate seasonal models, parameters are non-linear; even Gaussain residual when appied to non-linear transformation is no longer a Gaussian distribution. It is possible to apply a filter, known as the **extended Kalman filter**, by making a locally linear approximation to the Kalman filter model. In order to extend the filter, we employ first-order Taylor series expansion $f(\mathrm{y})=f(\bar{y})+f^{/}(y)(y-\bar{y})$. This extended filter performs effectively if the function is locally linear, that is, providing a good approximation using only first-order Taylor expansion, but it would not work well if the extended filter functions are not locally linear. However, even if the disturbances are non-Gaussian, we can still apply the Kalman filter to obtain linear projections employing the quasi-maximum likelihood *QMLE* that does not require normality, but still yields consistent estimates of the elements of ***F, Q, A, H*** and ***R*** that are asymptotically normal.

16.7 *Smoothing Kalman Filter*

The Kalman filter examined so far is an algorithm for calculating a forecast of the state vector $\boldsymbol{\xi_t}$ as a linear function of previous observations. In some applications, the value of unobservable state $\boldsymbol{\xi_t}$ may be of interest in its own right, rather than for its use in forecasting. For example, we may wish to know the state of an unobservable factor at a *historical* time *t* by making an inference about the value of $\boldsymbol{\xi_t}$ based on the full set of measured and forecasted information; such an inference is called the **Kalman smoothing estimation** and denoted by:

$$\hat{\xi}_t|_T=\hat{E}(\xi_t|y_t) \tag{16.7.1}$$

The MSE associated with (16.7.1) smoothed estimation of the matrix of $\boldsymbol{\xi_t}$ is

$$P_t|_T=E\{[\xi_t-\hat{\xi}_{t|T})]\,[\xi_t-\hat{\xi}_{t|T})]^{/}\} \tag{16.7.2}$$

A well-known application of the smoothing Kalman filter is the study of the business cycle by Stock and Watson (1991) mentioned earlier. For example, based on GDP observation from 1954 through 1990, we can estimate the value $\boldsymbol{\xi_t}$ took in 1960 using all measured data on $\boldsymbol{\xi_t}$ and $\boldsymbol{x_t}$ through date *T*. Therefore, unlike forecasting, smoothing estimation depends on the full set of measured data up to and including *t*.

16.8 *Filtering and Decomposition of Simple Time Series*

There are different approaches to filter/decomposing a time series into seasonal and cyclical trends when the models consist of a relatively simple structure. We conclude with a brief consideration of the (1977) Hodrick-Prescott (*HP*) and Hamilton (2018) time-series decomposition.

Suppose the observed time series y_t is composed of a trend component, y^*_t and a cyclical component c_t. The *HP* method isolates c_t from y_t by minimizing:

$$\text{Min}_{y*1,\, y*2,\, .\, .\, .\, ,\, y*T} =[\sum_1^T(y_t - y_t^*) +\lambda\sum_{t=2}^{T-1}(\Delta^2 y_{t+1}^*)^2] \qquad (16.8.1)$$

where λ is a smoothing parameter usually chosen by trial and error; as λ approaches 0, the trend component becomes equivalent to the original series, while as λ goes to ∞, y^*_t becomes a linear trend, since the second differenced term $(\Delta^2 y_{t+1}^*)$=0. The *HP* filter identifies the cyclical component by trading off the trend against the desired degree of smoothness, and for quarterly observations it is set to 1,600, based on prior beliefs about the magnitudes of changes in the cyclical component, relative to the trend component.

Hamilton (2018, RES) highlights three specific drawbacks to the *HP* filter. First, (16.8.1) induces spurious cycles, or spurious dynamic relationships; specifically, the filter induces spurious cycles when applied to differenced stationary time series, which is a leading example of a typical economic time series, and best described by a random walk. Second, (16.8.1) has an end-of-sample bias, as filtered values in the middle of the sample and at the end are very different. This can lead to substantial biases in small samples. Third, the common choice λ = 1, 600 is ad hoc; the formulation requires (16.8.1) error terms be white noise, which are clearly unrealistic assumptions. Estimating such an optimal λ, Hamilton (2018) finds that, for a series of macroeconomic and financial variables, it should be close to 1 rather than 1,600.

Hamilton (2018) proposed an OLS regression of the observed non-stationary time series, y_t , at date $t + h$ on a constant and its four most recent values at date t, namely:

$$y_{t+h} = \beta_0 + \beta_{1yt} + \beta_{2yt-1} + \beta_{3yt-2} + \beta_{4yt-3} + v_{t+h} \qquad (16.8.2)$$

The stationary, or cyclical, component is then obtained from the residuals,

$$\hat{v}_{t+h} = y_{t+h} - \hat{\beta}_0 - \hat{\beta}_{1yt} - \hat{\beta}_{2yt-1} - \hat{\beta}_{3yt-2} - \hat{\beta}_{4yt-3}. \qquad (16.8.4)$$

(16.8.4) has the advantage that we do not have to know the true data-generating process before applying it, and it results in stationary residuals provided the fourth differences of the original time series are stationary. In the case of quarterly data, Hamilton suggests employing $h = 8$ for analyses concerned with business cycles and $h = 20$ for studies interested in credit or financial cycles. However, empirical evidence from (8.4) applications show it shares some of the drawbacks of (16.8.1). It amplifies cycles that exceed the duration of regular business cycles, namely, longer than eight years, and completely mutes certain shorter-term fluctuations. Due to this, (16.8.4) falls short of reproducing the chronology of US business cycles. Nonetheless, this amplification of cycles may be helpful for some applications. For instance, a credit-to-GDP gap derived with (16.8.4) indicates that imbalances prior to the global financial crisis started earlier than reported by the official credit-to-GDP gap, which is derived using the *HP* filter. In general, (16.8.4) produces more robust cycle estimates than the *HP* filter, which can be important if policy measures draw upon these estimates.

Still, research interest on filtering of macroeconomic time series that go beyond the simple linear and time invariant structural *ARIMA* models may not produce identical forecasts between the structural and reduced form parameters. Then, such complex models can be more easily analyzed in the state-space form with the state of the system representing the various unobserved components such as trend and seasonal.

Appendix-Driving the Kalman Filter: key steps

The following lists the key steps in the derivation of the Kalman filter for the S-S representation given by (16.5.1)-(16.5.2); for the step-by-step derivation, see Hamilton (1994), section 13.2.

Stage 1-diffuse initialization: this is just ξ_0=E(ξ_1) , the unconditional mean of ξ and its MSE is $P_t|_0$=E{[ξ_1 – E(ξ_1)] [ξ_1 – E(ξ_1)]$^/$}

Stage 2-Forcasting y_t

$$\hat{E}(\xi_t|x_t, \mathbf{Y}_{t-1})= \hat{E}(\xi_t|\mathbf{Y}_{t-1})= \hat{\xi}_{t|\,t-1} \qquad (16.51a)$$

since x_{t-1} contains no information beyond *t-1*. Use that to predict y_t:

$$y_{t|t-1} = \mathrm{A}^/x_t + \mathrm{H}^/\hat{E}(\xi_t|x_t,\mathbf{Y}_{t-1})= \mathrm{A}^/\mathrm{x}_\mathrm{t} + \mathrm{H}^/\xi_{t\,|\,t\text{-}1} \qquad (16.5.2a)$$

Hence,

$$y_t - \hat{y}_{t-1}=[H^{/}(\xi_t - \hat{\xi}_{t|t-1})+w_t] \qquad (16.5.3a)$$

The expectation of squared (16.5.3a), after dropping the cross-products terms, results in:

$$MSE=E[(y_t - y_{t|t-1})\,(y_t - y_{t|t-1})^{/}]= E[H^{/}(\xi_t - \hat{\xi}_{t|t-1})^{/}H]+[w_t\, w_t{}^{/}]=[H^{/}P_{t|t-1}H+R] \qquad (16.5.4a)$$

Stage 3-Updating forecast ξ_{t+1}:

$$\hat{\xi}_{t|t}= \hat{\xi}_{t|t-1}+ P_{t|t-1}H\,(H^{/}\,P_{t|t-1}H+R)^{-1}(y_t - A^{/}x_t - H^{/}\,\hat{\xi}_{t|t-1}) \qquad (16.5.5.a)^{22}$$

The MSE of (16.5.5.a) for the updated projection, $P_{t|t}$, is:

$$P_{t|t}=E[(\xi_t - \hat{\xi}_{tt})(\xi_t\ \hat{\xi}_{t|t})^{/}]= P_{t|t-1} - P_{t|t-1}H(H^{/}\,P_{t|t-1}H+R)^{-1}H^{/}\,P_{t|t-1} \qquad (16.5.6.a)$$

skeeping a number of steps, see Hamilton, p.389.

Stage 4-Producing forecast ξ_{t+1} by (16.5.1.a):

$$\hat{\xi}_{t+1|t}= F\hat{\xi}_{t|t-1}+ [FP_{t|t-1}H(H^{/}P_{t|t-1}H+R)^{-1}](y_t - A^{/}x_t - H^{/}\hat{\xi}_{t|t-1}) \qquad (16.5.7.a)$$

The coefficient matrix in (16.5.7.a) is the Kalman gain matrix K_t:

$$K_t=FP_{t|t-1}H(H^{/}\,P_{t|t-1}H+R)^{-1} \qquad (16.5.8.a)$$

$\boldsymbol{K_t}$ weights or corrects the forecast error (last term in round brackets) to obtain optimal forecasts.

Readings

For textbook discussion, see Hamilton (chapter 13) and Chatfield and Xing (2019, chapter 10); Commandeur and Koopman (2007, chapters 10 and 11). The Kalman (1960) proposed filter comes from control engineering; Burmeister and Kent (1982) illustrates an application to economics.

[22] (16. 5.5.a) and (16.5.6.a) are derived from the matrix presentation of the optimal forecast for P(Y3|Y2, Y1), and its associated MSE given in Hamilton (p.99) given by equations [4.5.30] and [4.5.31]. Here corresponding equations are obtained with Y3= $\boldsymbol{\xi_t}$, Y2=$\boldsymbol{y_t}$, and Y1=(x_t', y_t')' in (16.5.4.a).

Chapter 16 State-Space & Kalman Filter Exercises

Q16.1 Show that

a. the local level S-S model is equivalent to an ARIMA (0, 1, 1) model when $\boldsymbol{x_t}$ is first differenced;

b. Show that the following local linear trend S-S model is equivalent to an ARIMA (0, 2, 2) model when $\boldsymbol{x_t}$ is second differenced.

Q16.2 a. Write out the MA (1) process in a state-space representation.

b. Explain if the following is a valid state-space representation of MA (1) process~~?~~

$$\begin{bmatrix}\varepsilon_{t+1}\\ \varepsilon_t\end{bmatrix}=\begin{bmatrix}0 & 0\\ 0 & 0\end{bmatrix}\begin{bmatrix}\varepsilon_t\\ \varepsilon_{t-1}\end{bmatrix}+\begin{bmatrix}\varepsilon_{t+1}\\ \varepsilon_t\end{bmatrix}$$ S.E with *r*=2 states

$$y_t-\mu=[1\ \ \theta]\begin{bmatrix}\varepsilon_t\\ \varepsilon_{t-1}\end{bmatrix}$$ M.E with *n*=1

Q16.3 Consider the following special case of the linear growth model

$X_t=\mu_t+n_t$

$\mu_t=\mu_{t-1}+\beta_{t-1}$

$\beta_t=\beta_{t-1}+\omega_t$

with n_t & ω_t *iid* with zero means and respective variances σ^2_n & σ^2_ω. Show that the initial least squares estimator of the state vector at time *t*=2, in terms of the observations $\boldsymbol{X_1}$ & $\boldsymbol{X_2}$, is

$[\hat{\boldsymbol{\mu}}_2\ \hat{\boldsymbol{\beta}}_2]=[\boldsymbol{X_2}, \boldsymbol{X_2}-\boldsymbol{X_1}]$ and variance-covariance matrix $P_2=\begin{bmatrix}\sigma^2_n & \sigma^2_n\\ \sigma^2_n & 2\sigma^2_n+\sigma^2_\omega\end{bmatrix}$.

Q16.4 Download *nile.dat,* data on the flow of water ~~i~~in the Nile river at Aswan**.**

a. Estimate a *S-S, AR*(1) model of *RWMN* for the water flow data

b. Post estimation: predict a smoothed local level trend using a diffuse KF filter by *rmse*, predict standardized residuals, plot each graph and then both in a single plot.

c. Compute RMSE for prediction and forecasts ~~predictions~~ plus 50% CI, and graph the result.

Q16.5 Download *manufac.dat* on US capacity utilization.

a. Estimate a model in S-S form a VARMA (1, 1) process in differenced “capital” and “hours”. Obtain the *sspace* error-form syntax for estimation, use a Model of a S-S ARMA (1, 1).

b. Post estimation: predict the differenced capital utilization by one-step ahead and the standardized residuals by KF

Q16.6 Download *dfex.dt*, a US macro data set.

a. Estimate an S-S form of a dynamic-factor model (AR model augmented by unobserved factors) that follows an AR(2) process with no exogenous variables; with AR(1) disturbances in the observable equation.

b. Estimate one-step ahead forecasts for *D.unemp* and graph the result.

Chapter 17 Bayesian Econometrics Analysis

Introduction

Bayesian Econometrics has gained much intertest as a result of greater availability of computer power. The Bayesian approach to econometrics requires the specification of a probabilistic model of prior beliefs about the unknown parameters. The presentations in this and the next chapter are intended as an introduction of this growing area of econometrics.

i. Bayesian Probability

The basic innovation of the Bayesian theorem is revising the probability of an ***A*** event when new information about it becomes available by modifying its probability conditional on event ***B***. The conditional probability of event *A*, given event *B* is $P(A|B) = \frac{P(A \cap B)}{P(B)}$, that is the conditional probability of *A* is equal to the ratio of its *joint* probability to its *marginal* probability. We also note that we obtain marginal probabilities of two events (that is, the probability of each *A* and *B* event separately) from the sum of their joint probabilities: when events *A & B* are conditional on the event *M*, then *P*(*M*)=*P*(*A*|*M*)+*P*(*B*|*M*). Bayes' theorem expands this definition by accounting for the effect of the initial probability called the **prior probability**, on the final probability, called the **posterior probability**. In general, after substituting for joint probability of an event ***i*** from the definition of the conditional, for ***n*** mutually exclusive events, we have

$$P(A_i|B) = \frac{P(A_i)P(B|A_i)}{P(A_1)P(B|A_1) + P(A_2)P(B|A_2)+\ldots+P(A_n)P(B|A_n)}$$

where the denominator expresses the marginal probabilities as the sum of the given ~~of~~ joint probabilities. Simplifying by writing the sum of the marginals in the denominator as ***P***(***B***) and dropping the index for A, we can write the Bayes' theorem as

$$P(A|B) = \frac{P(A)P(B|A)}{P(B)}$$

For a *discrete* random variable ***y*** and parameter of interest ***Θ***, the Bayes theorem is written as

$$\pi(\theta|y) = \frac{P(y|\theta)\pi(\theta)}{P(y)} \qquad (17.\ 1.1)$$

where $P(y) = \int P(y|\theta)\pi(\theta)d\theta$ for the probability mass function (*p.m.f*) of *P*(.); division by $P(y)$ makes $\pi(\theta|y)$ a normalized probability distribution; that is, integration with respect to Θ results in $\int \pi(\theta|y)\pi(\theta)d\theta$ =1.In general, for a continuous random variable y, we write

$$\pi(\theta|y) = \frac{L(y|\theta)\pi(\theta)}{L(y)} \qquad (17.1.2)$$

where $f(y) = \int L(y|\theta)\pi(\theta)d\theta$ for the probability density function (***p.d.f***) of ***y***(**.**).

Here the first term of the numerator, $L(y|\,\Theta)$ called the *likelihood function*, is a function of Θ once the data are known. As a simple example, take a coin experiment y tossed three times with results as (H, T, H), so *y* (1, 0, 1). If Θ is the probability of a head, we have the likelihood function as the product of conditional probabilities given by:

$$P(1, 0, 1)=P(1|\,\Theta)\ P(0|\,\Theta)\ P(1|\,\Theta)= \Theta\ (1- \Theta)\ \Theta = \Theta^2\ (1- \Theta)$$

The second term in the numerator of (17.1.2), $\pi(\theta)$ is the *prior density*, a measure of our belief about the distribution of Θ before seeing the data y; the prior distribution usually depends on parameters, known as *hyperparameters,* provided by the researcher. Finally, the denominator *f*(*y*) normalizes the posterior, and when it is independent of Θ, it is convenient by convention, to write it as *proportional* to the likelihood function times the prior distribution, and the likelihood is commonly written as $f\,(y|\theta)$:

$$\pi(\theta|y)\ \propto\ f(y|\theta)\pi(\theta) \qquad (17.1.3)$$

When the posterior is written without the inessential constant terms as it is here, then (17.1.3) is known as a **density kernel**.

Examples of Bayes Theorem

a. Discrete variable: Bernoulli/Binomial (with *n* fixed). The likelihood function for a single toss of a~~f~~fair coin is $P(y_i|\,\Theta) = \theta^{\,y_i}\,(1\text{-}\theta)^{1-\,y_i}$, implying $P(y_i =1|\,\Theta) = \Theta$ and $P(y_i =0|\,\Theta)$ =(1 - Θ). Now generalizing this to the case of *n* independent tosses of a coin, ~~then~~ we have

$$\begin{aligned} P(y_{1,\,,\,\dots,}\ y_n\,|\,\Theta) &= \theta^{\,y_1}\,(1\text{-}\theta)^{1-\,y_1}\dots\theta^{\,y_n}\,(1\text{-}\theta)^{1-\,y_n} \\ &= \prod \theta^{\,y_i}\,(1-\theta)^{1-\,y_i} \\ &= \theta^{\,\Sigma y_n}\,(1\text{-}\theta)^{n-\,\Sigma y_n} \qquad (17.1.4) \end{aligned}$$

The completion of (17.1.4) with this likelihood function requires a specification of a prior distribution function that meets the constraint that $0 \le \Theta \le 1$; a usual option is the beta distribution, *beta* (α, β), with its prior function defined as

$$\pi(\theta) = \frac{\Gamma(\alpha+ \beta)}{\Gamma(\alpha)\Gamma(\beta)} \ \theta^{\alpha-1} (1\text{-}\theta)^{\beta-1} \ ; 0 \le \Theta \le 1 \text{ and } \alpha, \beta > 0$$

where $\Gamma(.)$ stands for the beta function and α, β are hyperparameters; note also that the first term in the ratio is free from Θ and hence becomes a part of the constant of proportionality in (17.1.3). The beta prior function of varies in the $0 \le \Theta \le 1$ range and its distribution is defined by its first two moments, see Greenberg (2013), p. 226, as

$$E(\theta) = \frac{\alpha}{\alpha + \beta} \ \& \ Var(\theta) = \frac{\alpha\beta}{(\alpha + \beta)^2(\alpha + \beta + 1)}$$

The beta distribution moments make clear that the shape of the function depends on the values assumed hyperparameters; indeed, one reason for the choice of the beta distribution is that with different values provided by the researcher for β relative to α, the function can generate many different shapes. Thus, as shown in Figure 17.1, this prior can capture beliefs that Θ is centered at one-half, or it trends toward zero or one; highly concentrated or highly dispersed; and can have bimodal when β_0= α_0 = 0.5.

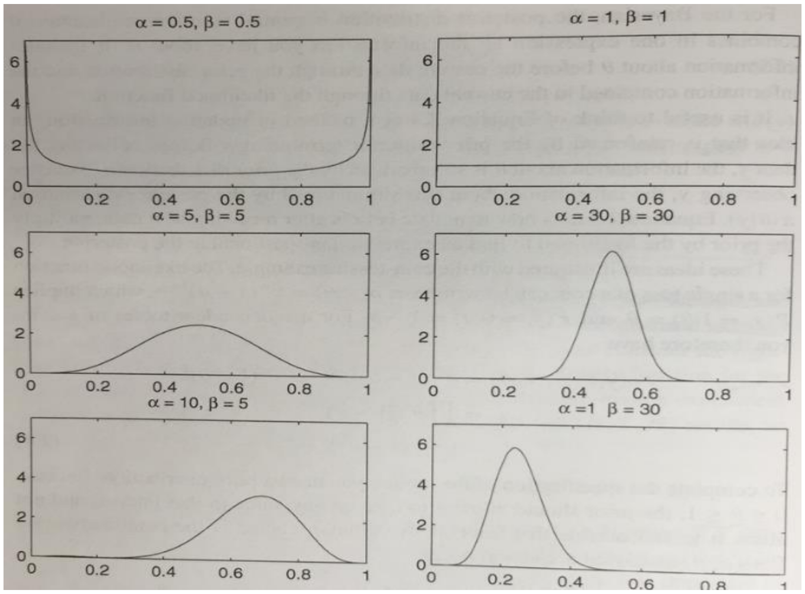


Figure 17.1 *Beta distribution for various vales of* **α & β**

Another very important reason for choosing the beta prior is that in combination with (17.1.4), the likelihood function with Bernoulli distribution with a *beta* (α, β) prior, the posterior ia also beta distribution. We can write (17.1.1) in this case as

$$\pi(\theta|y) \; \propto \; f(y|\theta)\pi(\theta)$$

$$\propto \; [\theta^{\sum y_n} \text{ (1-}\theta)^{n-\sum y_n}] \, [\theta^{\alpha_0 - 1} \text{ (1-}\theta)^{\beta_0 - 1}]$$

$$\propto \;\; \theta^{(\alpha_0 + \sum y_n) - 1}(1\text{- }\theta)^{(\beta_0 + n - \sum y_n) - 1}$$

Bearing in mind that the normalizing constant of the beta distribution (the first term in the ratio above) has been into the constant of proportionality because of its independence from $\boldsymbol{\Theta}$, the remaining terms produce $\pi(\theta|y)$ that is in the form of a beta distribution with parameters $\alpha_1 = (\alpha_0 + \sum y_i)$& $\beta_1 = (\beta_0 + n - \sum y_i)$. This is an example of a ***conjugate*** prior with the posterior in the same family of distribution as the prior distribution; examined further below. We can now easily compute the mean of the beta posterior distribution:

$$E(\theta|y) = \frac{\alpha_1}{\alpha_1 + \beta_1} = \frac{\alpha_0 + \sum y_i}{\alpha_0 + \beta_0 + n}$$

If we substitute for $\bar{y} = (\frac{1}{n})\sum y_i$ and can re-write the last line in the equivalent form ~~as~~ (since the numerator of the first ratio and the denominator of the second ratio, both from the first term, cancel out)

$$E(\theta|y) = (\frac{\alpha_0 + \beta_0}{\alpha_0 + \beta_0 + n})\frac{\alpha_0}{\alpha_0 + \beta_0} + (\frac{n}{\alpha_0 + \beta_0 + n})\bar{y} \tag{17.1.5}$$

(17.1.5) expresses as a weighted average of the prior mean, $\frac{\alpha_0}{\alpha_0 + \beta_0}$, and the maximum likelihood estimator (*MLE*) $\boldsymbol{\bar{y}}$**,** the value of $\boldsymbol{\theta}$ that maximizes *P*(y| *Θ*). (17.1.5) reveals an important feature the Bayesian method of inference: as the sample size *n* increases the weight given to the prior mean tends toward zero while that on the *MLE* one, implying $E(\theta|y) \rightarrow \bar{y}$. The prior loses importance and the features of the posterior distribution come to resemble increasingly those of the likelihood function as the latter dominate the final distribution with increases in the sample size. This outcome is demonstrated in Figure 17.2 with $\boldsymbol{\alpha_0} = \boldsymbol{\beta_0} = 2$ and ***n***=10 or 50 and $\sum y_i$= 3 or 15.

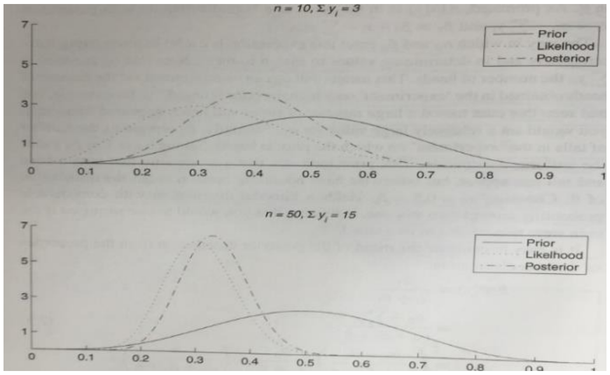


Figure 17.2 *Prior, likelihood and posterior for the coin-tossing*

b. *Continuous random variable*

Normal distribution is the workhorse of econometrics and provides a frequent model for applied Bayesian analysis, it is therefore selected as a Bayesian example of a continuous random variable y_i.

i. *Normal Distribution*

Once again, the aim is to obtain a posterior by (17.1.2), given a normal likelihood and prior distribution function.

Suppose we specify a likelihood function for (17.1.2) with a random variable $y \sim N\,(\theta, \sigma^2)$ where θ is known but the scalar σ^2 is unknown; the joint density of y, given a random sample of

(***y₁, . . . , y_N***) is:

$$L(y|\theta) = \prod_{i=1}^{N} (2\pi\sigma^2)^{-1/2}\, exp\{-(y_i - \theta)^2/2\sigma^2\}$$

$$= (2\pi\sigma^2)^{-N/2} exp\{-\textstyle\sum_{i=1}^{N}(y_i - \theta)^2/2\sigma^2\}$$

$$\propto\; exp\{\frac{N}{\sigma^2}\,(\bar{y} - \theta)^2\}$$

where $\bar{y} = \left(\frac{1}{n}\right)\sum y_i$ and we use $\sum_{i=1}^{N}(y_i - \theta)^2 = (y_i + \bar{y} - \bar{y} - \theta)^2 = \sum_{i=1}^{N}(\bar{y} - \theta)^2 + \sum_{i=1}^{N}(y_i - \bar{y})^2$. Note that multiplicative terms not involving θ are absorbed in the constant of proportionality and dropped from the last exponential line (as is the constant proportional denominator of (17.1.2)). Next, we need to specify a prior function. Given a normal likelihood function, an analytically convenient choice would be a normal prior, $\theta \sim \mathcal{N}(\boldsymbol{\mu}, \tau^2)$, since then the product of two normally distributed functions in (17.1.2) also results in a normally distributed posterior. More specifically, we say such a normal prior is a *conjugate* of the normal posterior, a large value of $\boldsymbol{\tau}$ reflects a more uncertain prior than a small value. Here we demonstrate that the resulting posterior is in the same normal density function form as the prior distribution. The prior density function is

$$\pi(\theta)= (2\pi\tau^2)^{-1/2} exp\{-(\theta-\mu)^2/2\tau^2\}$$

$$\propto \ exp\{-(\theta - \ \mu)^2/2\tau^2\}$$

Multiply the two functions to obtain the posterior, after dropping the constant term $\boldsymbol{\pi}$ and taking all terms into the exponentials; and then expand the terms inside the curly brackets:

$$L(y|\theta)\pi(\theta) = exp\{-\tfrac{1}{(2\sigma^2)} \textstyle\sum_{i=1}^{N}(y_i-\theta)^2\}exp\{-\tfrac{1}{(2\tau^2)} (\theta-\mu)^2\}$$

$$= exp\{-\tfrac{1}{(2\sigma^2)} \textstyle\sum_{i=1}^{N}(y_i^2 - 2y_i\theta + \theta^2)\}exp\{-\tfrac{1}{(2\tau^2)} (\theta^2 - 2\theta\mu + \mu^2)\}$$

$$= exp\{\tfrac{1}{(2\sigma^2)} (\textstyle\sum_{\mathrm{i}=1}^{\mathrm{N}} \mathrm{y}_\mathrm{i}^2 - 2\,\theta \textstyle\sum_{i=1}^{N} y_i + N\theta^2)\}\, exp\,\{\tfrac{1}{(2\tau^2)} (\theta^2 - 2\theta\mu + \mu^2)\}$$

Dropping the (un-highlighted) terms that do not involve *θ*, and combining the rest within only one exponent, then the expression that the exponent contains can be rearranged to collect the θ^2&$\boldsymbol{\theta}$ terms; using $N\bar{y} = \sum y_i$ to ease notation:

$$L(y|\theta)\pi(\theta) = -\frac{1}{2}(\frac{N}{\sigma^2}\theta^2 + \frac{N}{\tau^2}\theta^2 - 2\,\theta\frac{\sum_{i=1}^{N} y_i}{\sigma^2} - \ 2\,\theta\frac{1}{\tau^2}\mu)$$

$$= -\frac{1}{2}((\frac{N}{\sigma^2} + \frac{1}{\tau^2})\theta^2 - 2\,(\frac{\sum_{i=1}^{N} y_i}{\sigma^2} - \frac{1}{\tau^2}\mu)\theta)$$

$$L(y|\theta)\pi(\theta) = -\tfrac{1}{2}((\tfrac{N}{\sigma^2} + \tfrac{1}{\tau^2})\theta^2 - 2\,(\tfrac{N}{\sigma^2}\bar{y} - \tfrac{1}{\tau^2}\mu)\theta) \qquad (17.1.6)$$

This is now in the form of (*ax*^{2} -2*bx*) with θ for *x* where *a*=$(\frac{N}{\sigma^2} + \frac{1}{\tau^2})$ & *b*=$(\frac{N}{\sigma^2}\bar{y} - \frac{1}{\tau^2}\theta)$; we can transform this by dividing through by *a* (so the coefficient of *x*^{2} becomes unity and that of *x*, a ratio of *b/a=c*, becomes a constant) into an expression of the form $(x - c)^2$. The linear solution for x inside the brackets is easily compared to that of a quadratic one. The mathematical procedure to carry out for this task is known as ***completing the square***, employed when solving a quadratic equation directly is too complex, especially if the solutions for that equation depend on other quadratic equations as is the case here. Writing the last line with the simple *a & b* notations, we have:

$$L(y|\theta)\pi(\theta) = -\frac{1}{2}(a\theta^2 - 2\,b\theta)$$

$$= -\frac{a}{2}\left(\theta^2 - 2\,\frac{b}{a}\theta\right)$$

Adding + and – terms of $(\frac{b}{a})^2$ to this equation leaves it unchanged but since neither of them depends on μ, we can simply drop one of them (un-highlighted) as a part of the constant of proportionality

$$L(y|\theta)\pi(\theta) = -\frac{a}{2}\left(\theta^2 - 2\,\frac{b}{a}\theta + \frac{b^2}{a^2} - \frac{\mathrm{b}^2}{\mathrm{a}^2}\right)$$

$$\propto \{-\frac{a}{2}\left(\theta^2 - 2\,\frac{b}{a}\theta + \frac{b^2}{a^2}\right)\}$$

Now, the expression inside the curved parentheses is in the form of (x^2 - $2xc$ + c^2)=(x – c)^2; replacing the terms of the exponent, we have $exp\{-\frac{a}{2}\left(\theta + \frac{b}{a}\right)^2\}$. This reveals the posterior distribution of μ when the prior is also normal, also has a normal distribution since its mean of *b/a* and a variance of *1/a* moments involve normally distributed terms; solving (17.1.6) for its first two moments:

$$\theta|y \sim N(\mu_n, \sigma_n^2)$$

$$\sigma_n^2 = \frac{1}{a} = 1/(\frac{N}{\sigma^2} + \frac{1}{\tau^2}) \qquad (17.1.7)$$

$$\mu_n = \frac{b}{a} = \sigma_n^2(\frac{N}{\sigma^2}\bar{y} - \frac{1}{\tau^2}\mu) \qquad (17.1.8)$$

The posterior mean $\boldsymbol{\mu_n}$ is a weighted sum of the prior mean μ and the sample mean $\bar{\boldsymbol{y}}$ with weights that depend on the precision of the likelihood via $\boldsymbol{\frac{N}{\sigma^2}}$ and on the prior via $\boldsymbol{\tau^2}$. In Bayesian analysis, variability is measured by the **precision parameter**, defined as the reciprocal of the variance; the **posterior precision** $\boldsymbol{\tau^{-2}}$ is the sum of the sample precision of $\bar{\boldsymbol{y}}$, namely $\boldsymbol{\frac{N}{\sigma^2}}$, and the **prior**

precision $1/\boldsymbol{\tau}^2$, therefore, precision increases by pooling the sample and the prior information. If the prior information is imprecise, so that $1/\boldsymbol{\tau}^2$ is small, the prior has little effect in generating the posterior, and the sample information dominates the posterior as the increase in the sample size makes $\frac{\boldsymbol{N}}{\boldsymbol{\sigma}^2}$ relatively larger. This outcome is similar to asymptotic normality except that the Bayesian parameter estimation depends on the values of the sample at hand rather all possible values of the parameter.

Figure 17.3 shows an example with $\boldsymbol{\sigma}^2$=100, the prior sets $\boldsymbol{\mu}$=5 and $\boldsymbol{\tau}^2$=3, and $\boldsymbol{N}$=50 with sample mean of $\bar{\boldsymbol{y}}$=10. Then the likelihood is *N*[10, 2], the prior *N*[5, 3], and from (17.1.7) & (17.1.8), we have $\boldsymbol{\sigma}^{2*}$ =[1/(50/100)+1/3)]=1.2004 & μ_*=1.2004*[(50*10/100)+5/3)]=8.06, therefore *N*[8, 1.2]. These are the densities plotted in Fig. 1.3 that shows the posterior mean lies between the prior mean and the sample mean, whereas the posterior has a smaller variance than the variance of both the prior and the likelihood.

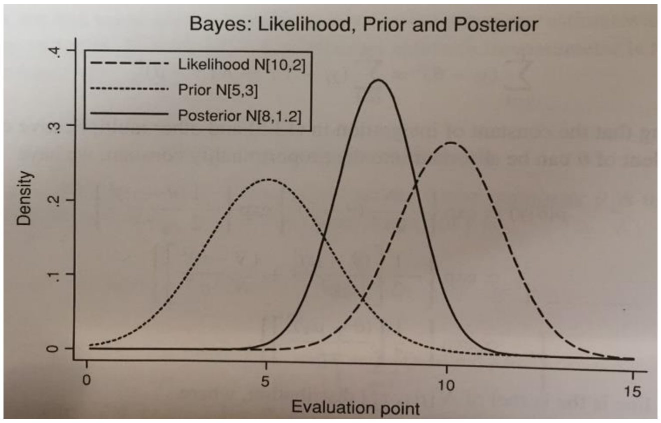


Figure 17.3*- Bayesian analysis for mean parameter of normal density: plot of normal likelihood (right), normal prior density (left), and resulting posterior density (center)*

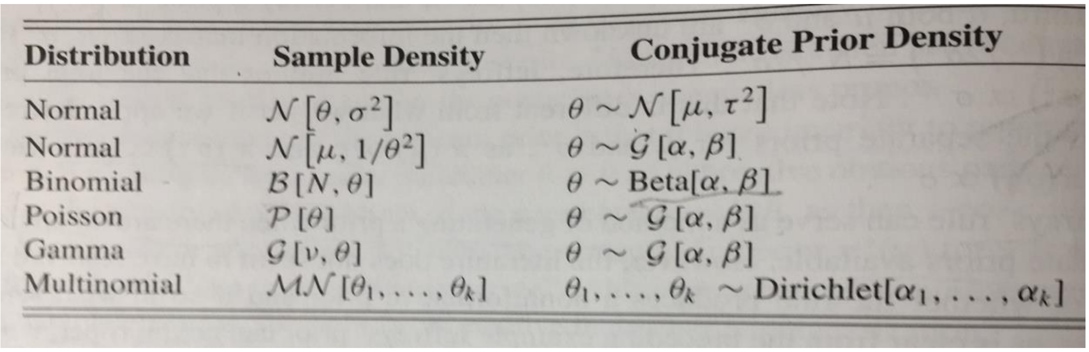
We also apply completing the square technique to obtain the key moments (17.1.7) and (17.1.8) for the posterior distribution of $\boldsymbol{\mu}$ based on a normally distributed prior conjugate, but the same results with slight variations are obtainable if the prior function such as normal-inverse gamma conjugate prior, belongs to a family of normal distribution, see Greenberg section 4.3. With a multivariate normal distribution~~s~~, the exponent has its quadratic form as $(\boldsymbol{y} - \boldsymbol{\mu})' \sum^{-1}(\boldsymbol{y} - \boldsymbol{\mu})$ and (17.1.7) and (17.1.8) are represented as functions of vectors and matrices, see Greenberg, A.1.14.

ii.*Specification of prior*

Bayesian analysis requires specification of the data generating process (*dgp*) $\boldsymbol{f}(\boldsymbol{y}|\boldsymbol{\theta})$& the prior $\boldsymbol{\pi}(\boldsymbol{\theta})$; the former is frequently assumed the same as the specification of a parametric likelihood-based model because that assumption leads to analytically tractable distribution for the posterior. Such tractable results often arise if the sample and prior densities and posterior distributions all belong to the same family of densities. An example, examined above, is, for the normally distributed data, and, a normal prior for the mean, results in a normally distributed posterior.

Table 17.1 presents some standard conjugate families; example in its first row. We note that the important class of gamma density includes exponential and chi-square as special cases. An advantage of having a posterior and prior in the same distributional class is that the posterior can act as a new, data-based prior for the next round of analysis. Despite such advantages it is important to highlight that a conjugate prior is equivalent to *imposing a restriction* and as such must be justified. The main difficulty of Bayesian analysis is the specification of a prior distribution, the princip~~a~~l of contention with the classical analysis. One option is to employ a prior that has little impact on the posterior another, is to use an informative prior if strong prior information is available, and yet an intermediate alternative is to rel~~a~~y on hierarchical priors that involve other uncertain priors. Table 17.1 illustrates some of the commonly employed conjugate functional pairs.

Table 17.1-*Leading Examples of Conjugate Families*



Distribution	Sample Density	Conjugate Prior Density
Normal	$\mathcal{N}\left[\theta, \sigma^2\right]$	$\theta \sim \mathcal{N}\left[\mu, \tau^2\right]$
Normal	$\mathcal{N}\left[\mu, 1/\theta^2\right]$	$\theta \sim \mathcal{G}[\alpha, \beta]$
Binomial	$\mathcal{B}[N, \theta]$	$\theta \sim \text{Beta}[\alpha, \beta]$
Poisson	$\mathcal{P}[\theta]$	$\theta \sim \mathcal{G}[\alpha, \beta]$
Gamma	$\mathcal{G}[\nu, \theta]$	$\theta \sim \mathcal{G}[\alpha, \beta]$
Multinomial	$\mathcal{MN}[\theta_1, \ldots, \theta_k]$	$\theta_1, \ldots, \theta_k \sim \text{Dirichlet}[\alpha_1, \ldots, \alpha_k]$

iii. *Noninformative Priors*

One possibility exists for a **uniform prior** that attaches the equal weight to all possible values of the parameter of interest θ: $\pi(\theta)$=*c* and ***c*** > 0. The problem is that if the $\boldsymbol{\theta}$ values are unrestricted, or unbounded, then the necessary integration does not sum up to one, and the prior becomes an **improper density**: $\pi(\theta)d\theta = \infty$, though you can also have an improper integral if the integrant has discontinuity, namely, 1/a-1 with *a*=1. Usually, the corresponding posterior distribution can also be improper, though not in all cases. Moreover, the uniform prior is not invariant to re-parameterization. For example, given $\theta > 0$, another alternative density for y would be $\gamma = \ln \theta$, and $-\infty < \gamma < \infty$. If θ has a uniform prior, that is if $\pi(\theta)$=*c*, then the new prior $\pi^{*}(\gamma) = \pi(\theta) \left|\frac{d\theta}{dy}\right| = ce^{\gamma}$. This prior is proper and informative, since it is not constant, changing with γ, even though its different parameterization produces an uninformative constant prior. A uniform prior for a random variable θ is sometimes specified as a proper prior with very large (uninformative) variances $\boldsymbol{\tau^2}$, assumed distributed as $N\,(\mu, \tau^2)$. Then, for values of θ likely to be supported by data for a uniform distribution, values around 0.5, $exp\{-\frac{(\mu - \mu_0)^2}{2\tau^2}\} \simeq$1, the prior $\pi(\theta) \simeq$1/2πτ^2 becomes a constant. Thus, this approach, known as **vague, flat**, or **diffuse prior**, displays the same problem as the uniform distribution of not being invariant to re-parameterization.

A widely used non-informative prior which is invariant to re-parameterization is **Jeffreys' prior**. Jeffrey's prior is based on the information matrix of θ, that is the amount of information that a random variable y offers about an unknown parameter θ; estimated by the MLE that maximizes the (conditional) log-likelihood function. Formally, the elements of this matrix are defined by the second derivatives of the MLE of y with respect to θ; it is the expected value of the MLE of θ defined as $\mathfrak{I} = -E[\frac{\partial \mathcal{L}_N(\theta)}{\partial \theta} \ \frac{\partial \mathcal{L}_N(\theta)}{\partial \theta'}$]= a constant; and it is equal to the *variance of the score vector* of the LME (first derivates of MLE with respect to θ , $\frac{1}{N}\frac{\partial \mathcal{L}_N(\theta)}{\partial \theta}$, is called its score vector). The ML estimator solves the first-order condition that implies the score vector has expected zero value, $namely, E = \left[\frac{\partial \mathcal{L}_N(\theta)}{\partial \theta}\right] = 0$. Large values of $\mathfrak{I}$ mean that small changes in θ result in large changes in the log-likelihood, suggesting $\mathfrak{I}$ offers a great deal of information about θ. Jeffrey's prior is based on the determinant of the information matrix $\mathfrak{I}|\,\theta|$ for a vector of θ as:

$$\pi(\theta) \ \propto \mathfrak{I}|\,\theta|^{1/2} \qquad (17.1.9)$$

That is, *Jeffrey's prior is proportional to the variance of the scores*. This prior provides the same information regardless of the particular parameterization or transformation of the model employed. To verify the rule in the scalar parameter case, consider transformation *γ=h(θ)*,

we have

$$\frac{\partial \mathcal{L}}{\partial \gamma} = \frac{\partial \mathcal{L}}{\partial \theta} \cdot \frac{\partial \theta}{\partial \gamma} \text{ and } \frac{\partial^2 \mathcal{L}}{\partial \gamma^2} = \frac{\partial^2 \mathcal{L}}{\partial \theta^2} \cdot \left(\frac{\partial \theta}{\partial \gamma}\right)^2 + \frac{\partial^2 \theta}{\partial \theta \partial \gamma^2}$$

Using (17.1.9) and taking the expectation of this equation (& noting that $\boldsymbol{\frac{\partial \mathcal{L}}{\partial \theta}}$ =0 by the property of likelihood scores)

$$\mathfrak{I}(\gamma) = \mathfrak{I}(\theta)\left(\frac{\partial \theta}{\partial \gamma}\right)^2$$

Given that $|\boldsymbol{\frac{\partial \theta}{\partial \gamma}}|$ is a constant, we obtain a new prior that is again proportional to the variance of the new information matrix as:

$$|\mathfrak{I}(\gamma)|^{1/2} = |\mathfrak{I}(\theta)|^{1/2} |\frac{\partial \theta}{\partial \gamma}|$$

However, although Jeffrey's prior is invariant to re-parameterization, it is not always a proper prior. As an example, suppose $y \sim N\,(\mu, \sigma^2)$, and consider three cases. First, μ is unknown but σ^2 is a known constant. The information index for μ is $\mathfrak{I}(\mu) = N/\theta^2$, therefore, though proportional to the information variance, Jeffery's prior becomes $|\mathfrak{I}(\gamma)|^{\frac{1}{2}} \propto c$, a constant. Second, μ is known but σ^2 is unknown. The information index for σ^2 by (17.1.9) is $\mathfrak{I}(\sigma^2) = N/(\theta^4)$, and Jeffrey's prior is $|\mathfrak{I}(\sigma^2)|^{\frac{1}{2}} \propto \sigma^{-2}$. Third, if both μ and σ^2 are unknown; then $\mathfrak{I}(\mu, \sigma^2) = \frac{N}{\theta^2} . \frac{N}{\theta^4} = N^2/2\theta^6$, therefore, the joint prior is $\pi(\mu, \sigma^2) \propto \sigma^{-3}$. However, if Jeffrey's prior is applied separately to μ and σ^2,-we then have a different outcome with the variance of proportionality as ~~then~~ $\pi(\mu) \propto c, \pi\,(\sigma^2) \propto \sigma^{-2}$ & $\pi(\mu)\pi\,(\sigma^2) \propto \sigma^{-2}$. The Jeffreys' prior is improper if it has an unrestricted constant.

Finally, if the regression model has normal distribution, then a type of prior specific to that model is Zellner's g-prior. This prior requires the specification of the dimension of the prior (the number of regression coefficients), a degree of freedom, and the variance parameter of the error term (see exercise Q17.4_c.)

Summarizing, noninformative methods of prior specification are either vulnerable to the improper prior problem that leads to improper posterior, or to the problem of not being invariant to re-parameterization; they may often lead to similar values obtainable by the simpler classical methods.

iv. *Bayesian linear regression with noninformative prior*

The Bayesian analysis of linear regression analysis provides a basis for general Bayesian models; the OLS estimator has a Bayesian interpretation as the mean of the posterior distribution in the noninformative prior case. Consider $y \sim N$ (μ, σ^2), with known prior constant for μ, and unknown prior for σ^2. Extending Jeffery's prior to the linear regression case means the prior views all values of regression coefficient β_j, *J=1, . . . , K*, as equally likely whereas small values of σ^2 are viewed as more likely. Assuming independence of β and σ^2, the *joint prior* is :

$$\pi(\ \mu, \sigma^2\ )\ \propto\ 1/\sigma^{\,2}$$

First, we re-expressed the likelihood function as

$$L(\beta,\ \sigma^2|y,X) = (2\pi\sigma^2)^{-\frac{N}{2}}exp\{\frac{1}{(2\sigma^2)} - (y-X\beta)'(y-X\beta)\} \qquad (17.1.10)$$

$$\propto\ \ (\sigma^2)^{-N/2}\ exp(\frac{1}{2\sigma^2} - \{\hat{u}'\hat{u} + \left(\beta-\hat{\beta}\ \right)'\mathrm{X}'\mathrm{X}\left(\beta-\hat{\beta}\ \right)\})$$

$$\propto\ \ (\sigma^2)^{-N/2}\ exp\ (\frac{1}{2\sigma^2}\ (N-K)s^2 + \left(\beta-\hat{\beta}\ \right)'\mathrm{X}'\mathrm{X}\left(\beta-\hat{\beta}\ \right))$$

where $\hat{\beta}$=(X′X) $^{-1}$ X′y, $\hat{u}$= y - X$\hat{\beta}$ and we use squared (y - X β)=$\hat{u} -$ X$\left(\beta-\hat{\beta}\ \right)$ and X$\hat{u}'$=0; and the final line employs s^2=$\hat{u}'\hat{u}$ / (*N*- *K*).

The combination of (17.1.10) with Jeffrey's proportional prior leads to the posterior density as

$$P(\beta,\sigma^2|y,X) \propto\ \ (\tfrac{1}{\sigma^2})^{\ N/2}exp\ (\frac{1}{2\sigma^2}\ \{(N-K)s^2 + \left(\beta-\hat{\beta}\ \right)'\mathrm{X}'\mathrm{X}\left(\beta-\hat{\beta}\ \right)\}\big)(1/\sigma^2)\ (17.1.11)$$

$$\propto\ \ (\tfrac{1}{\sigma^2})^{\ N/2+1}\ \ exp\ (\frac{1}{2\sigma^2}\{\ (N-K)s^2 + \left(\beta-\hat{\beta}\ \right)'\mathrm{X}'\mathrm{X}\left(\beta-\hat{\beta}\ \right)\}\big)$$

$$\propto\ \ \{\left(\frac{1}{\sigma^2}\right)^{\frac{K}{2}}\ \ exp\ (\tfrac{1}{2}\left(\beta-\hat{\beta}\ \right)'(\sigma^2(\mathrm{X}'\mathrm{X})^{-\,1})^{-\,1}\left(\beta-\hat{\beta}\ \right)\big)\} \times \left(\frac{1}{\sigma^2}\right)^{(N-K)/2+1}\ exp(-\frac{(N-K)s^2}{2\sigma^2})$$

where in the third line, we use $(\frac{1}{\sigma^2})^{\ N/2+1} = [\left(\frac{1}{\sigma^2}\right)^{\frac{K}{2}} + \left(\frac{1}{\sigma^2}\right)^{(N-K)/2} + (\frac{1}{\sigma^2})]$. The first two lines contain Jeffery's $(1/\sigma^2)$ proportionality, the third is the product of likelihood for $\hat{\beta}$=*1, . . . , K*, and the variance σ^2, *K+1, . . . N*. The conditional posterior distribution of β is $(\beta|\sigma^2, y, X)$, given σ^2, and that data on *y*, *X* is a *K*-dimensional multivariate with mean $\hat{\beta}$ and variance $(\sigma^2(\mathrm{X}'\mathrm{X})^{-\,1})$ since only the first line of the final term contains β.

The marginal (or marginalizing) posterior of β is obtained by integration out σ^2 from the second line of (17.1.11) by change of variables[23] to $z=1/\sigma^2$ & letting $\alpha=\{(N-K)s^2+(\beta-\hat{\beta})'\mathrm{X}'\mathrm{X}(\beta-\hat{\beta})]$, $c=N/2+1$, and expressing the integral as $\int_0^\infty z^c \exp(-\alpha z)\,dz$. For a given constant $\alpha>0$ and $c> -1$, using the property of exponential function integration, $\int e^x = e^x$, and noting that σ^2 in the second line of (17.1.11) is now a known constant, this yields the kernel of the marginal posterior distribution $\hat{\beta}$ conditional on σ^2as:

$$P(\beta|y,X) \propto \{(N-K)s^2+(\beta-\hat{\beta})'\mathrm{X}'\mathrm{X}(\beta-\hat{\beta})\}^{-N/2} \qquad (17.1.12)$$

$$\propto \{(1+(\beta-\hat{\beta})'(s^2(N-K)\mathrm{X}'\mathrm{X}^{-1})^{-1}(\beta-\hat{\beta})\}^{-(N\text{-}K+K)/2sert\text{ `}a\text{'}} \quad \mathbf{??}$$

The second line is a result of a change of variables $z=1/\sigma^2$ (obtained after division by $(N-K)s^2$) and integrating z, see Cameron & Trivedi (2005, p. 436). (17.1.12) is in the form of a multivariant student-*t* distribution, see Greenberg (2013), p.230, as the covariance matrix $s^2(\mathrm{X}'\mathrm{X})^{-1}$ appears between the parameter estimation terms; it is centered on $\hat{\beta}$ with (*N-K*) degrees of freedom, and covariance matrix $s^2(\mathrm{X}'\mathrm{X})^{-1}$ multiplied (*N-K*) / (*N-K-2*). Therefore, each β_j has a univariate student-*t* distribution.

$$\beta \sim t_K(\hat{\beta}, s^2(\mathrm{X}'\mathrm{X})^{-1}) \qquad (17.1.13)$$

The conditional posterior of σ^2 given β , is harder to obtain since σ^2 appears in the first and the second lines of (17.1.10).

Bayesian analysis with a non-informative prior is similar to that obtained by the least squares' method: conditional on σ^2, the posterior of $\beta \sim N\ [\hat{\beta}, \sigma^2(\mathrm{X}'\mathrm{X})^{-1}]$, and unconditional posterior of β is the multivariate *t*-distribution.

v. *Linear Regression with Informative Priors*

The Bayesian normal linear regression model can also be modeled with informative priors. For example, the normal-gamma priors with the normal conjugate prior for β, and the gamma conjugate prior for 1/$\boldsymbol{\sigma^2}$ leads to the posterior of a normal-gamma type (see Cameron (2013), p.p. 437-8). In general, using a conjugate is equivalent to augmenting the data with a second sample

[23] We employ the change of variable method to simplify a more involved integration for an independent variable *x* by using another simpler variable *u* where, as above, the relationship between *x* and *u* is known.

from the same distribution as long as the priors are correctly specified. Therefore, the normal-gamma prior is equivalent to an additional sample of the same process as the regression parameter estimate. In effect, the sample and prior information are treated symmetrically, thus disregarding that the information from the two sources may be in conflict; it is the price involved in using conjugate priors. If the prior and the sample information are in conflict, then the posterior distribution can be bimodal, with one mean for the sample and another for the prior. A prior known as **Dickey's prior** that accounts for this bimodality is the multivariate Student-t density for β that is independent of a gamma prior $1/\sigma^2$.

vi. *Hierarchical Priors*

Hierarchical priors arise when parameters in a prior are themselves modelled as having a distribution; such a distribution is an intermediate prior between the non-informative and informative priors. With this type of prior, the prior parameters depend on another set of earlier parameters called **hyperparameters**, or "prior on prior" parameters. Now, the data have joint density $L(y|\theta)$ but the prior on θ depends on parameter τ that are random rather than fixed. Using Bayes' rule and the joint priors leads to the joint posterior:

$$P(\theta,\ \tau) \propto L(y|\theta)\ \pi\ (\theta|\tau)\pi(\tau)$$

We are usually interested in the marginal posterior for θ obtained by integrating the joint posterior with respect to τ.

Hierarchical priors arise naturally in the context of **hierarchical models**, also known as **multilevel models.** The data for analysis naturally fall into strata where one expects groupwise parameter variation in the model under study. Examples are modeling test scores by individual characteristics varying across students, grade class characteristics varying across grades, and school characteristics that vary across schools; such data involve *clustering* of observations.

Consider a two-stage linear regression model hierarchical in regression parameters but not in parameter variance. Denote the first stage linear regression as $Y=X_1\ \beta_1+u$, and β_1 depending on both parameters and data, so $\beta_1= X_2\ \beta_2+v$, and the errors assumed normally distributed; for example, the first level can be an individual firm features and the second level the industry characteristics. Then the second-level β_2 are unknown and a prior is specified for them resulting in the following model, see also Cameron & Trivedi (2005), p.441:

$$\text{y}|X_1, \beta_1, \sigma_1^2 \sim [\, X_1\beta_1, \sigma_1^2 I_N]$$
$$\beta_1|X_2\,, \beta_2, \textstyle\sum_2 \sim [\, X_2\beta_2, \textstyle\sum_2]$$
$$\beta_2 \sim N\,[\beta^*, \textstyle\sum^*]$$
$$\sigma_1^{-2}|v^*, \sigma^{*2} \sim \mathcal{G}\,[\tfrac{v^*}{2}, (v^*\sigma^{*2})/2] \qquad (17.1.14)$$

The second line provides the prior for the regression parameter in the first line, while the third line provides the subsequent second-stage prior, or a prior on a prior for β_2, assuming $\sum_2$is known. $(\beta^*, \sum^*)$ are the hyperparameters; the fourth line provides a prior for variance parameters parameter σ_1^2 by specifying $v^*, \sigma^{*2}$by gamma distribution $\mathcal{G}$. Assuming no functional form mis-specification, we can collapse the stages into a two-level model. As an example, suppose the data falls naturally into *J* group with differing population mean across groups; for individual *i* in group *j*, suppose $y_{ij} \sim N(\theta_j\ , \sigma^2)$ with known , σ^2. Then the sample mean for the *j*th group with N_j number of individuals s $\bar{y}_j \sim N(\theta_j\ , \sigma^2/\ N_j)$, assuming independence; a hierarchical model specifies the mean with a prior $\theta_j \sim N(\mu, \tau^2)$.

The methods discussed so far may not be able to capture the full first stage posterior distribution parameters analytically, but recent advances in computational methods can handle a Bayesian hierarchical model; in particular, the Gibbs sampler, examined below, is well suited to hierarchical priors because of their recursive structure.

***vii.** Bayesian Updating*

An interesting feature of Bayesian inference is that it updates the posterior as new information becomes available, namely, it treats the current posterior as a new prior as new information unfolds. As an example, let y_1 be the number of heads in tossing a coin n_1 times; the probability of heads is θ. Hence,

$$\pi(\theta|y_1) \propto f(y_1|\theta)\, \pi(\theta)$$

If now a new set of data y_2 becomes available, we compute a new posterior given the complete data set, $\pi(\theta|y_1, y_2)$, that is:

$$\begin{aligned}\pi(\theta|y_1, y_2) &\propto f(y_1, y_2|\theta)\, \pi(\theta)\\ &= f(y_2|y_1, \theta)\, f\,(y_1|\theta)\pi(\theta)\\ &\propto\ f(y_2|y_1, \theta)\, f\,(y_1|\theta) \qquad (17.1.15)\end{aligned}$$

If the data sets are independent, then $f\,(y_1|\ y_2,\ \theta)= f(y_2|\theta)$. By this procedure, the Bayesian posterior is updated to become the new prior for the next computation to reflect new information.

For a simple example of updating, consider data generated from Bernoulli trials with beta prior parameters α_0, and β_0 ; suppose the first n_1 trials and set $s_1=\sum y_{1i}$, and the second n_2 trials and set $s_2=\sum y_{2i}$. Then the posterior based on the first experiment is, (see also Greenberg (2013), p.26)

$$f(\theta|s_1)\ \propto\ \theta^{\ \alpha_0-1}\ (1\text{-}\theta)^{\beta_{0-1}}\ \theta^{\ s_1}\ (1\text{-}\theta)^{\ n_1\ -s_1}$$

The first two moments are obtained by the sum of the exponents to the bases θ & (1 - θ):

$$\theta|s_1\ \sim Beta\ (\alpha_0 + s_1, \beta_0 + (n_1 - s_1))$$

However, if instead we regard the latter as the prior for the second experiment, then we have:

$$f(\theta|s_{1,}s_{2,})\ \propto\ \theta^{\ \alpha_0+s_1\ -1}\ (1\text{-}\theta)^{\beta 0+(\ n_1\ -s_1)-1}\ \theta^{\ s_2}\ (1\text{-}\theta)^{\ n_2\ -s_2}$$

Once again, the sum of the exponents to the bases θ & (1 - θ) provide the first two moments of the new beta prior:

$$\theta|s_1,\ s_1\ \sim Beta\ (\alpha_0 + (s_1 + s_2\,),\ \beta_0 + (n_1 + n_2) - (s_1 + s_2))$$

The latter distribution follows from a $B(\alpha_0, \beta_0)$ prior specification obtaining (s_1 + s_2) from (n_1+n_2) trials. Therefore, if the data are sequentially generated, the Bayesian posterior becomes a new prior on new evidence. The approach allows new information to influence beliefs about a parameter.

viii. *Bayesian Inference*

The posterior distribution provides the basis of Bayesian inference; that is marginal posterior estimates, point estimation, interval estimation, and hypothesis testing. There are important conceptual differences between Bayesian and classical inference; let us consider each in turn.

ix. *Marginal Posterior*

In general, given a multinominal $\theta^{/}= (\,\theta_1, \ldots, \theta_q)$, interest lies in the marginal posterior density P(θ_k |y) of the individual *k*th parameter, θ_k, obtained by integrating out (marginalizing) the joint posterior of all the remaining (q-1) elements of θ.

$$P(\theta_k\,|y) = \int p(\,\theta_1, \ldots, \theta_p|y)d\theta_1 \ldots d\theta_{k-1}\, d\theta_{k+1} \ldots d\theta_q$$

$$= \int p(\theta_p|y)\, d\theta_{-k}$$

θ_{-k} in the second line denotes all elements of θ other than θ_k; unlike the symmetric and unimodal classical asymptotic normal distribution, Bayesian marginal density is usually asymmetric and not always unimodal.

***x.** Point estimation*

Bayesian approach to the estimation of a scalar parameter θ employs a loss function, ~~that is,~~ the loss involved if $\theta \neq \hat{\theta}$; usual examples are the absolute loss function $L_1(\hat{\theta}, \theta)=|\hat{\theta}, \theta|$ and the quadratic loss function $L_2(\hat{\theta}, \theta)=(\hat{\theta}, \theta)^2$. These loss functions minimize the $(\hat{\theta} - \theta)$ difference

And the loss increases with an increase in $|\hat{\theta} - \theta|$. The Bayesian estimator minimizes the expected value of the loss taken over the posterior distribution of θ:

$$E[L(\hat{\theta}, \theta)] = \int L(\hat{\theta}, \theta)\pi(\theta|y)d$$

Under quadratic loss, the function minimizes:

$$E[L(\hat{\theta}, \theta)] = \int (\hat{\theta} - \theta)^2 \pi(\theta|y)d$$

To obtain $\hat{\theta}$ estimate, differentiate this function with respect to $\theta$and set the resulting equation equal to zero.

$$2\int (\hat{\theta} - \theta)\pi(\theta|y)d\theta = 0$$

$$\hat{\theta} = \theta\pi(\theta|y)d\theta$$

That is, for a quadratic loss, the optimal point estimator of θ is the mean of the posterior distribution of θ. It must be noted that the interpretation of $\hat{\theta}$ is quite different from the classical point estimation that takes $\hat{\theta}$ as an unbiased estimate of the true θ, or consistent estimate of the true θ asymptotically in repeated sampling. Bayesian point estimation seeks to obtain an estimate of the entire posterior distribution conditional on the observed data y, not on all possible values of θ as an asymptotic requirement. Therefore, the Bayesian point estimation is not restricted to the mean and also provides estimates of other quintiles such as the medium. To see the difference, consider the coin-tossing example with $\hat{\theta} = (1/n)\sum y_i = \bar{y}$. To determine if the estimator is unbiased, we find the distribution of $\bar{y}$ for the Bernoulli model and compute its expected value over the entire distribution of $\bar{y}$,

$$E(\bar{y})= \int \bar{y}\, f\,(\bar{y}|\,\theta)d\bar{y}$$

Corresponding to every possible value of data, the classical estimator calculates every possible value of $\bar{y}$, not just the observations available from the sample at hand. Bayesian estimates, on the other hand, are conditional only on the data observed; there is no attempt to estimate a value for the true parameter. Against this advantage, however, you should bear in mind Bayesian possible issues related to the potential distortions by inappropriate function form of the prior, either due to improper noninformative functions, or from the implied restriction imposed by informative conjugates.

***xi.** Interval estimation*

The Bayesian confidence interval has a simpler interpretation compared to the classical approach. In the latter, 95% interval means in 95% of repeated sampling, the different point estimates of $\boldsymbol{\theta}$ all fall inside the upper and lower boundaries of the interval; this involves data that are not observed. By contrast, a Bayesian 95% posterior confidence interval of $\boldsymbol{\theta}$ means the estimate lies within the interval boundaries with posterior probability of 95% based only on the observed sample of the data.

***xii.** Hypothesis testing*

Since there is little interest in determining the true value of a parameter, hypothesis testing is not a focal issue in Bayesian econometrics; instead, the focus is on the range of values that $\boldsymbol{\theta}$ may take, given the data and a prior. For this kind of problem, the Bayesian approach gives more attention to model comparison.

***xiii.** Large Sample Bayesian Consistency*

We have discussed how the likelihood function dominates the posterior estimates as the sample size increase in the discrete case with a Bernoulli coin-tossing example. The same result holds with a continuous variable as the influence of even informative priors on the posterior estimation goes to zero as sample size gets larger. Because the posterior distribution is hard to work with, an asymptotic approximation as a substitute for the finite posterior is of interest in Bayesian analysis, and obtaining the asymptotic posterior with a large sample size then becomes easy since it is equal to the likelihood. Assume observations are iid, then the log-posterior is:

$$\sum_{=1}^{N} lnP(\theta|y_i) = ln\,\pi(\theta) + \sum_{i=1}^{N} ln\,f(y_i|\theta) \qquad (17.1.16)$$

(17.1.16) demonstrates clearly that with the contribution of the prior fixed as the sample grows with N, the posterior is dominated by the likelihood contribution; the asymptotic properties of the posterior model, $\hat{\theta}$, is then the maximum of the posterior, assuming the posterior to be unimodal and approximately symmetric. Moreover, note that the posterior mode converges to the *MLE* since the second term in (17.1.16) dominates as $N \rightarrow \infty$, therefore, the posterior mode is consistent if the *MLE* is consistent.

To obtain the asymptotic distribution of $\hat{\theta}$, consider a second order Taylor series expansion of the log posterior density around the posterior mode $\hat{\theta}$:

$$\ln \mathrm{P}(\theta|y) \simeq \ln P(\hat{\theta}|y) + \frac{1}{2}\left(\theta - \hat{\theta}\right)'\left[\frac{\partial^2 \ln P(\theta|y)}{\partial\theta\partial\theta'}\,|_{\theta=\theta'}\right]\left(\theta - \hat{\theta}\right) \qquad (17.1.17)$$

Note where (17.1.16) is simplified because $\partial P(\theta|y)/\partial\theta = 0$ when evaluated at the posterior mode, and we assume higher order terms are negligible. Let $\mathfrak{I}(\hat{\theta}) = - \frac{\partial^2 \ln P(\theta|y)}{\partial\theta\partial\theta'}\,|_{\theta=\theta'}$ be the observed information based on the posterior density ln P$(\theta|y)$ evaluated at the posterior mode, then exponentiating (17.1.17)yields

$$P(\theta|y) \propto \exp(-\frac{1}{2}\left(\theta - \hat{\theta}\right)'\mathfrak{I}(\hat{\theta})\left(\theta - \hat{\theta}\right)$$

This is the kernel of multivariate normal distribution with mean $\hat{\theta}$ and variance matrix $\mathfrak{I}\left(\hat{\theta}\right)^{-1}$; ~~thus~~ θ is distributed as

$$\theta|y \sim^{a} N\,[\hat{\theta}, \mathfrak{I}\left(\hat{\theta}\right)^{-1}] \qquad (17.1.18)$$

As $N \rightarrow \infty$, *the likelihood component dominates the posterior while the impact of the prior tends to zero, so the LME mode* $\hat{\theta}$ replaces the mode of the likelihood. This important result, known as the Bayesian central limit theorem, demonstrates that asymptotically the classical and Bayesian inferences will be based on the same limiting multivariate distribution. Therefore, there should be no difference between them. The full force of the implication of this result will become clear when examining numerical methods for approximating the posterior distribution.

xiv. *Model Comparison*

Model comparison to determine which among several models is best supported by the prior and the data is the main aspect of Bayesian inference. In ~~the~~ classical regression, two models may differ by which covariates are included corresponding to different specification of the parameter vector. In the Bayesian approach two models can differ by their priors, their likelihood, or their parameter. The problem is dealt with by computing the probability that Mi is the correct model, given the data; with only two models i=1, 2, we first compute $P(M_1|y)$ and then use that to obtain $P(M_2|y)$= 1 - $P(M_1|y)$. Employing Bayes theorem and first introducing the parameters, and then integrating them out:

$$P(M_1|y) = \frac{P(M_1)f_1(M_1|y)}{f(y)}$$

$$= \frac{P_1\int f_1(\mathrm{y},\theta_1|)M_1 d\theta_1}{f(y)}$$

$$= \frac{P_1\int f_1(\mathrm{y}|\theta_1,M_1)\pi_1(\theta_1|M_1)d\theta_1}{f(y)}$$

where

$$f(y)= P_1\int f_1(\mathrm{y}|\theta_1, M_1)\pi_1(\theta_1|M_1)d\theta_1 + P_2\int f_2(\mathrm{y}|\theta_2, M_2)\pi_2(\theta_2|M_2)d\theta_2 \qquad (17.1.19)$$

Therefore, each term of $f(y)$ contains the integral of a likelihood function with respect to a prior distribution.

$$m_i\,(y)=\int f_i(\mathrm{y}|\theta_i, M_i)\pi_i(\theta_i|M_i)d\theta_i \qquad (17.1.20)$$

(17.1.20) is called the *marginal likelihood* for model *i* and interpreted as the expected value of the likelihood function with respect to the prior. Using the definition of the posterior function, we have

$$\pi(\theta|y) = \frac{f(y|\theta)\pi(\theta)}{f(y)}$$

$$= \frac{f(y|\theta)\pi(\theta)}{\int f(y|\theta)\pi(\theta)d\theta}$$

Note that the marginal likelihood is equal to the inverse of the normalizing constant of the posterior distribution (the expressions in front of P_1 above), therefore, the correct marginal value requires including the normalizing constants of $f(y|\theta)$& $\pi(\theta)$.

The Bayesian comparison of two models is usually undertaken by computing the odds ratio in favor of Model 1 over Model 2, given the data:

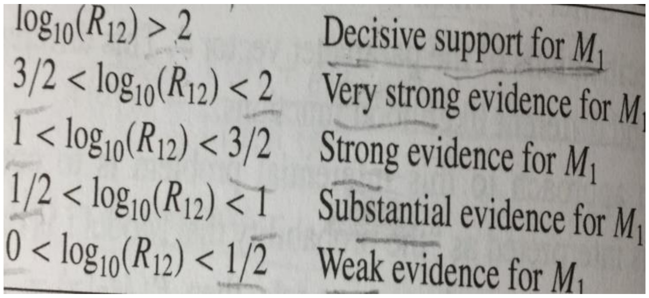
$$R_{12}=\frac{P(M_1|y)}{P(M_2|y)}$$

$$=\left(\frac{P_1}{P_2}\right)\left(\frac{\int f_1(\mathrm{y}|\theta_1,M_1)\pi_1(\theta_1|M_1)d\theta_1}{\int f_2(\mathrm{y}|\theta_2,M_2)\pi_2(\theta_2|M_2)d\theta_2}\right)$$

$$=\left(\frac{P_1}{P_2}\right)\left(\frac{m_1(y)}{m_2(y)}\right)$$

Note *f(y)* is dropped from the ratio above because it is common to both models. The first term on the RH is the prior odds ratio, the ratio of the prior probability M_1 to the prior probability of M_2. The second term, the ratio of the marginal likelihood of the two models, is called the **Bayes factor** and denoted by B_{12}. A large value of R_{12} is evidence of better support for M_1 over M_2 from the data and the prior information; a small value of R_{12} is evidence of better support for M_2; while a value around 1 suggests both models are equally supported by the data and the prior. A pairwise comparison can also be undertaken when there are more than two models by conveniently presenting the results in terms log_{10} (R_{12}) rather than R_{12} (integral part of a logarithm to base ten is interpreted as a power of ten). Table 17.2 provides guidelines for interpreting log_{10} (B_{12}). When there is little information with which to specify the prior odds ratio, the burden is on the Bayes factor to choose the models, provided there are no problematic prior specifications issues.

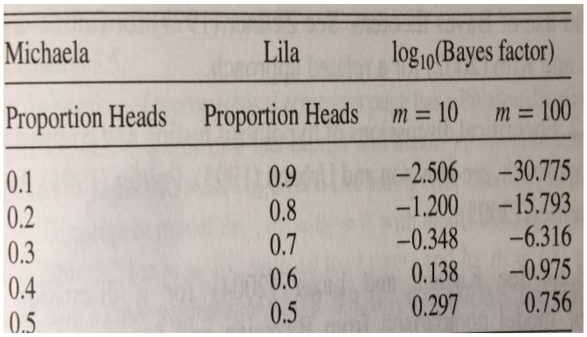
Table 17.2-*Jeffery's model comparison Guidelines*



$\log_{10}(R_{12}) > 2$	Decisive support for M_1
$3/2 < \log_{10}(R_{12}) < 2$	Very strong evidence for M_1
$1 < \log_{10}(R_{12}) < 3/2$	Strong evidence for M_1
$1/2 < \log_{10}(R_{12}) < 1$	Substantial evidence for M_1
$0 < \log_{10}(R_{12}) < 1/2$	Weak evidence for M_1

As an example, consider two competing models for *m* times tossing of a coin for a head with probability $\theta_1$when tossed by a boy (Michael) v. $\theta_2 \neq \theta_1$ when tossed by a girl (Lila), and test of $\theta_1=\theta_2=\theta$ in M_1: v. $\theta_2 \neq \theta_1$ in M_2. For simplicity, we assume priors $\pi_1(\theta_1)=B(1, 1)=\pi_1(\theta_1)$ which implies $\pi(\theta)=1$, $0 \leq \theta \leq 1$. The results of this experiment are shown in table 17.3 for selected values of outcomes with *m*=10 & 100, showing the $\log_{10}$(Bayes factor) supporting M_1 model as the proportion of heads for both players approaches 0.5(weak evidence), while M_1 is decisively rejected with large differences between the players and such differences are magnified with the larger sample size.

Table 17.3-*Bayes Factor for Possible selected outcomes*



Michaela	Lila	$\log_{10}$(Bayes factor)	
Proportion Heads	Proportion Heads	$m = 10$	$m = 100$
0.1	0.9	−2.506	−30.775
0.2	0.8	−1.200	−15.793
0.3	0.7	−0.348	−6.316
0.4	0.6	0.138	−0.975
0.5	0.5	0.297	0.756

An important advantage of the Bayesian comparison is the ability to conduct **non-nested** hypothesis testing. A common example is the choice between *y* and *log(y)* as the response (dependent) variable. Suppose that under M_1, the likelihood function is $f_1(\mathrm{y}|\theta_1)$, and under M_2, we have $f_2(\mathrm{z}|\theta_2)$, where $z=g(y)$ and $g(y)$ is monotone. Since y and $g(\mathrm{y})$ contain the same information, the posterior odds ratio should not depend on whether we have

$$\frac{P(M1|y)}{P(M2|y)}\text{or}\ \frac{P(M1|z)}{P(M2|z)}$$

Apply the usual transformation of variable rule:

$$f(\mathrm{z}_i|\theta)=f(\mathrm{y}_\mathrm{i}|\theta)|\frac{dyi}{dzi}|$$

The independence of the Bayes factor from different definitions of the response variable is clear from this transformation because the first derivative Jacobin terms cancel out from the top and bottom of the posterior odd ratio. By contrast, the classical approach has to first create a hybrid model combining both models, and then test each model against the hybrid.

Readings

For textbook discussion, see Greenberg (2014, Part I), Cameron and Trivedi (2005, chapter 13). Garthwaite et. al. (2005) survey elicitation methods for more objective priors.

Chapter 17 Bayesian Econometrics Exercises

Q17.1 (*Conjugate Bernoulli***)** Given the parameter $0 < \theta < 1$, N *iid* Bernoulli random variables Y_t=(1, 2, 3, . . . , T), each with *pmf*, and the likelihood function

$$P_B(y_t|\theta)=\begin{Bmatrix} \theta \text{ if } y_t = 1 \\ 1-\theta \text{ if } y_t = 0 \end{Bmatrix} \;\&\; L(\theta)=\theta^m(1-\theta)^{T-m} \qquad (1)$$

where $m=N$ for the number of successes, y_t=1. Suppose prior beliefs concerning θ are represented by a beta distribution with *pdf*

$PB(\theta\,|\underline{\alpha}, \underline{\delta})=[B(\underline{\alpha}, \underline{\delta})]^{-1}\,\theta^{\underline{\alpha}-1}(1-\theta)^{\underline{\delta}-1}$, $0< \theta <1$

Where $\underline{\alpha} > 0$ & $\underline{\delta} > 0$ are known, and $B(\underline{\alpha}, \underline{\delta})=\Gamma(\underline{\alpha})\Gamma(\underline{\delta})/\Gamma(\underline{\alpha}, \underline{\delta})$ is the beta function defined in terms of the gamma function $\Gamma(\alpha)=\int_0^\infty t^{\alpha-1}\exp(t)\,dt$. Find the posterior density of θ.

Q17.2 (*Conjugate gamma*) Consider a random sample Y_t (t=1, 2, . . . , T) from a distribution with *pdf* $P(\theta\,|y)=\theta y^{\theta-1}$ for 0<y<1, & $P(\theta\,|y)=0$ otherwise. Suppose the prior distribution of θ is the gamma $G(\underline{\alpha}, \underline{\beta})$ with $\underline{\alpha}>0$ and $\underline{\beta}>0$. Determine the mean and variance of the posterior distribution of θ.

Q17.3 (*Bernoulli sampling*) Consider a random sample from a Bernoulli distribution with the *pmf* given by (1) in Q17.1. Find Jeffreys' prior.

Q17.4 (*Jeffreys' prior re-parametrization*) Suppose Y_t (t=1, 2, . . . , T) are *iid* random variables from the exponential distribution with mean θ.

a. Derive Jeffreys' prior for θ.

b. Derive Jeffreys' prior for $\alpha=\theta^{-1}$.

c. Find the posterior density of θ corresponding to the prior density in (a). Be specific in noting the family to which it belongs.

d. Find the posterior density of α corresponding to the prior density in (b). Be specific in noting the family to which it belongs.

Chapter 18 Bayesian Simulation Models

Introduction

In many applications of Bayesian models, the parameter of interest is analytically interactable, and must rely on numerical methods to obtain approximations to various features of the posterior distribution. Such methods have computer-intensive time requirements and their employment has expanded rapidly with computer processing power. Such key moments of the posterior distribution can be estimated with explicitly obtaining the distribution itself. Many of the applications are attempts to approximate a quantity such $E[g(X)]$ given a distribution for $X \sim f(x)$, but an analytical computation $\int g(x)f(x)dx$ is not possible; similarly, this is the case for other moments, including those with intervals.

18.1 *Classical Simulation*

In this section we examine four most frequently employed simulations that generate *independent* samples from probability distributions: integral transformation, composition, accept-reject and importance sampling. These methods that also appear in modifications as parts of more general, Markov-based simulations that are *not* drawn from independent sampling but can handle a greater variety of distributions than those examined in this section, as discussed below.

i.*Integral Transformation Method*

The simplest method can demonstrate the ability of Bayesian simulation to generate independent samples from probability distributions. We use the convention that a variable in capital letters denotes a random variable while one in lower case denotes a particular value of that variable.

Suppose $F(.)$ stands for the *cdf* of a continuous random variable X; make draws from uniform distribution U [0, 1] and set $X= F^{-1}(U)$ which implies $U= F(X)$:

$$P[X \leq x]=P(F(X) \leq F(x)]$$

$$=P(U \leq F(x)]$$

$$= F(x)$$

The second line follows from the non-decreasing cdf X, and the last line follows from the property of the uniform distribution that $P(U \leq U)= u$; evaluated over X interval at the smallest values of such intervals. This method is called the **inverse transformation** or **probability integral transformation**; the simulation algorithm involves

1- Drawing *u* from *U[*0, 1]; values assumed to be independently drawn.
2- Return to *x*= $F^{-1}(U)$ as a draw from *f*(*x*).

The application requires the *cdf F(x)*to be fully known (normalization constant, and its kernel known) and its inverse easily computable; most computer programs have routines for the inverse functions of standard distributions.

Example: suppose we draw a sample from a random variable

$$y \sim f(y) = \begin{Bmatrix} \frac{3}{8}y^2 & if\ 0 \leq y \leq 2 \\ 0, & otherwise \end{Bmatrix}$$

First, find the d.f. for $0 \leq y \leq 2$ by computing the positive integral of this function (*y^3* as integral *$3y^2$* multiplied by the cubic root of 1/8), therefore:

$$F(y) = \frac{3}{8}\int_0^y y^2 dy = \frac{1}{8}y^3$$

Next, draw a value from *U*(0, 1), set *U*= $\frac{1}{8}y^3$ and solve for *Y*= $2U^{1/3}$as a draw from *f*(*y*). An important application of the inverse transformation method is to sample from truncated distributions such as univariate truncated normal or Student-t densities because accurate approximations to the inverse *cdf*.s are widely available from computer packages.

ii.*Composition Method*

Sometimes the density *f*(*x*) can be mixture densities of a compound distribution

$$f(x)=\int g(x|z)h(z)dz$$

If you can sample *z* by drawing from *h*(*z*), and conditional *x* from *g*(*x*| *z*), then the value of *x* is a drawing from *f*(*x*). As an example, consider the negative binomial distribution [λ, $\lambda(1+\alpha\,\lambda)$] where α & λ are given constants; ~~and~~ this function is based on the Poisson distribution and can be regarded as a Poisson-gamma compound distribution. First draw *z* from a gamma distribution with mean 1 and variance of α by a transformation of the exponential; then make draws from the Poisson distribution with mean $\boldsymbol{\lambda z}$, given $\boldsymbol{z}$ from the first step. This method is convenient to employ for estimating a joint distribution rather than a mix of conditional and marginal distributions.

iii.*Importance Sampling Method*

Suppose we want to estimate $E[g(X)]=\int g(x)f(x)dx$ but the integral is not analytically computable and we cannot use method of composition because we cannot sample from $f(x)$. An alternative is to take $h(X)$ to be a distribution from which we know how to simulate the integral

$$E[g(X)]=\int \frac{g(x)f(x)}{h(X)}\, h(x)\mathrm{dx}$$

This integral can be approximated by drawing a sample of G values, $x^{(g)}$, from $h(X)$; and then compute

$$E[g(X)] \approx \frac{1}{G}\sum g(X^{(g)})\frac{f(X^{(g)})}{h(X^{(g)})}$$

This is a weighted average of $g(X^{(g)})$ where the weights $\frac{f(X^{(g)})}{h(X^{(g)})}$ determine the importance of different points in the sample space, ~~hence~~ so the method is called importance sampling; it obtains approximation for $E[g(X)]$ by a *Monte Carlo simulation* when an analytical solution is unavailable. However, though sound in theory, the method is not practical, because $E_g[h(X)]$is unknown, but the approach offers potential gains if the weights are fairly flat. To find a suitable distribution for $h(.)$, the main question in the implementation of importance sampling, we note that a large f(x)/h(x) tends to occur when the tail of h(.) is very small compared to the tail of f(.). This suggests that the normal distribution is not a suitable choice for h(.) since it tends to zero very quickly. However, suppose the values for the moments of the parameter of interest θ are θ^sobtained from s=1, . . ., S draws of θ from the importance sampling density $g(\theta)$; given that, the importance sampling-based estimates of the posterior moments are consistent and asymptotically normally distributed, see Cameron and Trivedi (2005), p. 444-45. In view of the asymptotic normality of the log posterior, a suitable choice for $g(\theta)$ density is a multivariate t-distribution, see Greenberg (2013), A,1.17, with the mean set to the posterior mode, and degrees of freedom set to a value sufficiently small to ensure thick tails.

Example: obtain an approximation $[(1 + x^2)^{-1}]$ where $x\sim e^1$, truncated to [0, 1] interval. Therefore, we approximate the integral

$$\frac{1}{1-e^{-1}}\int_0^1 \frac{1}{1+x^2}\ e^{-x}dx$$

by choosing *Beta* (2, 3) defined on [0, 1], this provides a good match between beta function and target density in the interval. Then, apply the following algorithm

1. Generate a sample G values, $X^{(1)},\ldots,X^{(G)}$ from *Beta* (2, 3) function
2. Calculate $\frac{1}{G}\sum_1^G(\frac{1}{1+\,(X^{(g)})^2})(\frac{e^{-X^{(g)}}}{1-e^{-1}})(\frac{B(2,3)}{X^{(g)}(1-\,(X^{(g)})^2})$

iv. Accept-Reject Methods

Suppose we want to draw from the density *f*(*x*)but it is difficult to do so. However, there is another distribution $g(x)$ from which we can draw easily that covers *f*(*x*) in the sense that *f*(*x*)≤ *kf*(*g*) for some finite constant *k* for all *x* values. If the unknown is *k*≥1, it is possible to simulate values from a density $g(x)$ for all X in the support, namely the range of values, of *f*(*x*). *f*(*x*) is called the **target density**, typically the posterior, *g*(*x*) the **proposal density**, and *k* **dominating density**. The draws from *g*(*x*) rather than *f*(*x*) are accepted if

$$k \le \frac{f(x)}{g(x)}$$

where *k* is drawn from the uniform distribution. If this condition is not satisfied, then the draw is rejected and further draws are made until the condition is met. Therefore, the algorithm of this method is as follows:

1-Generate a value *x* from *g*(*x*)

2-Draw a value *u* from *U*[0, 1]

3-Return *x* as a draw from *f*(*x*) if $u \le \frac{f(x)}{kg(x)}$; if not, reject and return to step 1. Accept *x* with probability $\frac{f(x)}{kg(x)}$ and continue until the desired number of draws is obtained. This procedure is known as the **Accept-Reject** (**AR**) method. To show how the AR method works, consider$h[x|u \le \frac{f(x)}{kg(x)}]$; using Bayes' theorem together with the property of uniform distribution *P(u*≤ *t)*=*t* for 0 ≤ *t* ≤ 1, we have:

$$h\left[x\middle|u \leq \frac{f(x)}{kg(x)}\right] = \frac{p[u \leq f(x)/kg(x)|x]g(x)}{\int p[u \leq f(x)/kg(x)|x]g(x)dx}$$

$$= \frac{[f(x)/kg(x)]g(x)}{\frac{1}{k}\int f(x)dx} = f(x)$$

where $\frac{1}{k} = \int p[u \leq f(x)/kg(x)|x]g(x)dx$. While this proves the method works, it also points out to its limitation, since on average a draw will be accepted with probability 1/*k*, so that many draws are necessary if *k* is large. This suggests the choice of *k* should be as small as possible in order to maximize the probability of acceptance because rejected draws use computer time without adding to the sample. The attraction of the *AR* method depends on the ease of drawing from *g*(*x*) rather than *f*(*x*). It should be noted that the *AR* algorithm is also useful when the normalizing constant of *f*(*x*) is unknown, when for *f*(*x*)= *k r*(.), we know *r*(.), but *k* is still unknown, Then, choose *k* so that *r*(x) ≤ *kg*(x), so, accepted values of x are a sample of *f* (*x*). Thus, the AR method can be employed even if the normalizing constant of the target distribution is unknown; in this case it is not required that *k* ≥1. Figure 2.1 explains the AR method

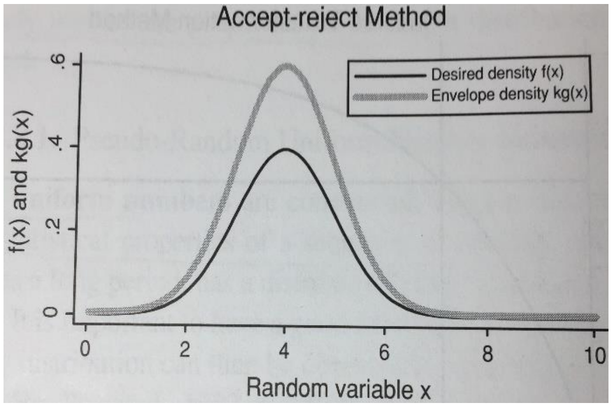


Figure 2.1-the ***AR*** method draws from density ***g***(***x***) where ***kg***(***x***) envelopes the desired density ***f***(***x***)

Example1: consider drawing from the negative binomial ~ [*λ*, *λ*(1+*αλ*)] where *α* & *λ* ae constant. Since the negative binomial distribution has a mixed Poisson-gamma distribution, see notes on the Poisson model, first draw *ε* from a gamma distribution with mean 1 and variance *α,* obtained from a transformation of the exponential. Then, draw from the Poisson distribution with mean *λε*, given *ε* from step 1.

Example2: Consider sampling from *Beta* (3, 3) with *U* (0, 1) as the proposal density. The maximum of the target density occurs at y=1.2 where the density function equals c=1.8750, and 1/c=0.5333[24]. In this case, the target is far from that generated by the proposal function because values close to zero and one are over-sampled by the proposal function.

The AR method is similar to the Metropolis-Hastings (M-H) method (section III below) since both involve a rejection step but with important differences. First, the M-H is more general; it is employed to sample from a greater variety of distributions. Second, the M-H method tends to generate positively correlated rather than independent samples, which produce a smaller variance and more information from a given sample size. True, negatively correlated, samples result in even smaller variance than independent samples, but there are no sure known methods that generate either independent or negatively correlated samples, so the M-H method remains popular for simulation in applied Bayesian analysis.

18.2 *Markov Chain Simulation*

The simulation methods examined so far aim at obtaining estimates for the summary moments of the target distribution. In any case, the classical simulation methods are often inefficient; for instance, the AR procedure can result in a high percentage of rejected draws. An alternative is to simulate by sequentially drawing simulated values that, if the sequence is run long enough, converge to a stationary invariant distribution that corresponds to the target posterior density. Once convergence is achieved, such sequential draws can be employed to estimate summary measures for the posterior. This alternative approach is known as the **Markov Chain Monte Carlo** (***MCMC***) simulation; it attempts to obtain the distribution of the target function from a large sample drawn from the posterior distribution, and then estimate the desired distributional moments from such a sample. The draws are positively correlated; therefore, the precision of the estimates will be reduced as the estimated variance will exceed the usual one; however, the approach makes up for this drawback by offering great flexibility. The development of the ***MCMC*** since the 1990's has greatly increased the scope of the Bayesian methods. We first look at Markov chain transitional

[24] Since the beta distribution function parameters are related to the gamma function, find the nearest row in the gamma table col. headed 0.5, that is 0.5371, and difference that from the corresponding value in the first column of the same row (the cumulative value) is then c=0.5371-0.0047=0.5333, and 1/c=1.875.

probability from one state to another, for example probability of oberseving a person in a day in state of “working”, “eating”, “resting”, or “sleeping”; the list of all possible states forms the “state-space” of this example. A Markov chain gives the probility of moving from “working” state, say to another, “eating” without “resting” first.

***i.**Finite State ,*

With a two-state Markov chain, there are 4 possible transitions since each state can return back to itself as shown in Figure 2.2. If the movement between states are as likely, the probability of transition from one state to another is 0.5. As the number of states increase, we use a transition matrix to account for the scores

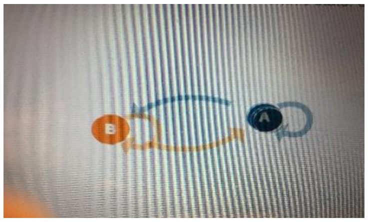


Figure 2.2-2-state Markov Chain

	A	B
A	P(A|A): 0.5	P(B|A): 0.5
B	P(A|B): 0.5	P(B|B): 0.5

In the matrix transional presenation, each state is shown by one row and one column, so the number of cells increases quadratically with each new state added to a Markoc chain. A real-world Markov chain application is by comuter simulation, and often displays “stickiness”, that is greater than 0.5 probabilty of staying in the state and smaller than 0.5 probabilty of transitioning to another state. For our example, it more likely to observe the person in the state of “working”, or “sleeping” in the next period than transitioning to the state of “resting”, or “eating”. Figure 2.3 mimics such as a two-state Markov chain that has a 0.9 probability of staying state “S” when starting in state “S”, and only 0.1 chance of leaving it to transition to state “R”. That is, once you start from a given state, there is a much higher probability of staying put in the same state.

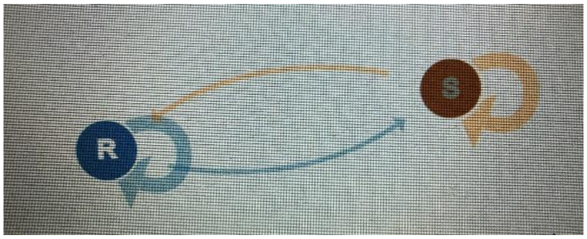


Figure 2.3-“Sticky” 2-state Markov Chain

We examine the ***MCMC*** mainly in the context of a stochastic process $\boldsymbol{X_t}$ that takes values in the finite discrete set $\boldsymbol{S}$=(1, . . ,$\boldsymbol{s}$) where the index $\boldsymbol{t}$ signifies time; we then briefly introduce the additional conditions required to ensure convergence to an invariant distribution when the state spaces are not finite or are continuous.

Given a pair of integers $i, j \in S$, let P_{ij} be the probability of $X_{t+1} = j$ given that $X_t=i$; that is:

$$P_{ij} = P(X_{t+1} = j | X_t = i),\ i, j \in S$$

where the P_{ij} are the *transitional probabilities*. The key assumption here is that the probability distribution at time $\boldsymbol{t}$+1 depends *only* on the state of the system at $\boldsymbol{t}$; a stochastic process that has this property is known as a **Markov process.** A Markov process is more general than an independent process but does not include all stochastic processes. Since P_{ij} are prob~~i~~abilites, we have $\boldsymbol{P_{ij}} \geq 0$, and since the process remains in $\boldsymbol{S}$, we also have $\sum_{j=1}^{s} P_{ij} = 1$., e discussion of M-H below. We define the s by s transitional matrix by $P = \{P_{ij}\}$. The ith row of P specifies the distribution of the process at time t +1 over the set S, given that it is in state i at t. For example, the transitional matrix

$$P = \begin{bmatrix} 0.750 & 0.250 \\ 0.125 & 0.875 \end{bmatrix} \qquad (18.2.1)$$

stays in state 1 with probability of 0.750, and moves to state 2 with probability of 0.250 if it starts in state 1; and if it starts in state 2, it moves to state 1 with probability of 0.125 and stays in state 2 with probability of 0.875. Next, we introduce some properties of the Markov chain needed to establish that the MCMC will converge on an invariant distribution of the target function.

The rows of a completely random or independent *P* are identical if $P_{ij} = P_i$, so a move from *i* to depends only on *j*, namely, an independent coin tossed with $P_1 = 2/3$ and $P_2 = 1/3$ will have a transition matrix with equal rows. If $p_{ij}^{(n)} > 0$ for some *n* states n≥1, then *j* is *accessible* from *i*,

i →*j* ; if *i* →*j* and *j*→*i*, *j* & *j communicate*, denoted as *i* ↔*j*. Using these notations, the relationship between two states *i* and *j* defines an *equivalence relationship* if they meet the following conditions: $i \leftrightarrow i$ (reflexivity), $i \leftrightarrow j \Leftrightarrow j \leftrightarrow i$ (symmetry); $i \leftrightarrow j$ and $j \leftrightarrow k \Rightarrow i \leftrightarrow k$ (transitivity). If starting from state ***i***, a Markov process can reach any other state with a positive probability, then the process has only one equivalence class, such a Markov process is called **irreducible**. More generally, if there are a sub-set of states you cannot reach from state *i*, then the process is reducible. On the other hand, irreducibility means the stochastic process goes from one state to any other state in a finite *n* numbers of steps.

For example

$$P_R = \begin{bmatrix} p_1 & 0 \\ 0 & p_2 \end{bmatrix} \qquad (18.2.2)$$

is not irreducible because if the process starts from any state {1, 2}, it will never leave that state. The implication in such cases is that the state at which the process starts has a huge impact on its subsequent path.

Another condition required in te application of the MCMC is *aperiodicity*. Figure 2.4 shows a periodic Markoc chain that when starts from state 0, retuns to it at *n*=3, 6, …. That is, the probability of returning to the same state is zero if *n* is not divisible by 3, $P_{00}^{(n)}=0$. Then such a state is called a *periodic* state with period *d* (0)=3. More geneally, if the period of *i*, *d*(*i*)>1, the state *i* is periodic and if *d*(*i*)=1, state *i* is **apriodic**; a Markov chain is aperiodic if all its states are aperiodic, that is, if $i \leftrightarrow j$, then *d*(*i*)= *d*(*j*). If we can go from state *i* to itself in *l* and again in *m* steps, then we have an aperiodic Markov chain. It follows that an irreducible Markov chain is also aperiodic because any state with a self-transition is aperiodic[25].

[25] If *l* and *m* are two co-prime numbers, that is their largest common divisor is 1, then $p_{ii}^{(l)} > 0$ and $p_{ii}^{(m)} > 0$, and we have an aperiodic chain. Since the number 1 is co-prime to every integer, any state with a self-trsnsition is aperiodic.

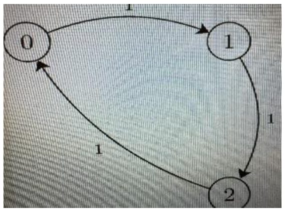


Figure 2.4 *Aperidocity*

Consider the distribution of states at *t*+2 denoted $p_{ij}^{(2)}$that can be computed as going from state *i* at *t*+1, to any other state *k* at *t*+2, and then moves to state *j* at *t*+3; so, the transition from *i* to *j* occurs in two steps with probability:

$$p_{ij}^{(2)} = \sum P_{ik}\, k_{ij}$$

If we define $p_{ij}^{(0)}$ =1 if *i* = *j*, zero otherwise, we can also define a general *n*-step transition matrix where the values of $p_{ij}^{(n)}$ are the *ij*th entries in the matrix P^n. Using a multiple-step process, we call a Markov chain *periodic* if the process at ***t***=1 must proceed to a second state at ***t***=2, before it can return to the first state at ***t***=3. Such a periodic Markov process has a chain of 2, or 2 steps; more generally in a ***n***-step transition process; such a positive probability of return to the same state exists only at even values of ***n***. If the period is 1 for all states, the chain is aperiodic. A Markov chain is aperiodic if $p_{ii}^{(n)}$> 0 for all i, and for sufficiently large *n*.

We employ irreducibility *and* aperiodicity to establish the invariant distributional property of the ***MCMC*** methods. The probability distribution π=(π_1, . . . , π_s)$^{/}$ is an invariant distribution for *P* if $^{/}= \pi^{/}P$, where π' is the characteristic vector[26]of *P*, or :

$$\pi_{j=} \sum_i P_{ij}\, \pi_j, \quad j=1, . . , s. \qquad (18.2.3)$$

The RHS of this equation is interpreted as the probability of starting at state I with probabilityπ_i , and then moving to state *j* with probability P_{ij}. It is π_j on the LHS that defines invariance; the fact that the probability is π_j , indicates that the system is in *j* at *any time*. Take the earlier example of a transition matrix:

[26] Given an *n by n* square matrix *D*, a scalar *r*, an *n . 1* vector *x*, and if *Dx=rx* is satisfied, then *r* is called a characteristic root of matrix *D*, and *x* a characteristic vector of that matrix; the equation *Dx=rx* is equivalently solved as a system of equations (*D – r* I)*x=0*.

$$(\pi_1\,,\ \pi_2\,)\begin{bmatrix}0.750 & 0.250\\ 0.125 & 0.875\end{bmatrix}= (\pi_1\,,\ \pi_2)$$

or $0.750\pi_1 + 0.125\ \pi_2 = \pi_1$ which, given $\pi_2 = 1\text{-}\ \pi_1$, leads to π=(1/3, 2/3).

The next question relates to the existence and uniqueness of the invariance distribution of the *MCMC*. It is clear that from (18.2.2) irreducibility is a necessary condition for a unique invariance distribution of *P*. Suppose π_1 and π_2 vectors satisfy $\pi'_1P_1= \pi'_1$ and $\pi'_2P_2= \pi'_2$. Then π is a weighted average as $\pi=[w\pi_1,\ (1\ –\ w)\pi_2\ ]$, $0\leq w\leq 1$shows the invariant distribution for *P* is not unique. Moreover, an irreducible and aperiodic Markov chain $p_{ij}^{(n)}$ can be shown to converge to a unique invariant distribution at a geometric rate when *n* is large enough, suggesting $P^{\ n}$ converges very quickly to the invariant distribution π', and the initial state *i* plays little role in the convergence process. This is the property of an independent $P^{\ n}$ process can be explained with the transitional matrix after 10, and then 20 simulations, showing $P^{\ n}$ has nearly reached invariance after *n*=10 simulations to two decimal points; has achieved an invariant distribution (all rows are equal) after *n*=20 to three decimal points.

$$P^{\,(10)} =\begin{bmatrix}0.339 & 0.661\\ 0.330 & 0.670\end{bmatrix} \qquad \& \qquad P^{\,(20)} =\begin{bmatrix}0.333 & 0.667\\ 0.333 & 0.667\end{bmatrix}$$

The above outlines the theorem that is the basis for MCMC methods:

Theorem 2.1: *If* ***P*** *is irreducible and an aperiodic transition matrix over a finite state spaces, then there is a unique probability distribution* $\boldsymbol{\pi}$ *such that* $\pi_{j=}\sum_{i}^{n} P_{ij}\,\pi_j$ *for all* $j\ \epsilon\ S$ *; and convergence to that distribution is at a positive geometric rate of* $0<\mathbf{r}<1$.

An informal method to verify the theorem is to consider what happens if these conditions are violated? First consider the reducible transition matrix (18.2.2)

$$P_R^n = \begin{bmatrix}P_1^n & 0\\ 0 & P_2^n\end{bmatrix}$$

Since the rows are not the same after n simulation, P_R^n does not have a unique invariant distribution. Now, take an irreducible but P_R^n is periodic

$$P_P^2 = \begin{bmatrix}P_1P_2 & 0\\ 0 & P_2P_1\end{bmatrix} \quad \& \qquad P_P^3 = \begin{bmatrix}0 & P_1P_2P_1\\ P_2P_1P_2 & 0\end{bmatrix}$$

Because this alternating pattern of the rows continues for every iteration, once again P_P^n does not converge to a matrix with identical rows. Therefore, the theorem maintains that irreducibility and aperiodicity are necessary and sufficient conditions to secure an invariant distribution outcome by the Markov chain process.

This theorem is generalizable to countable state spaces, states that are not finite but still have discrete values, and also to continuous distribution when a quantitative variable has an infinite number of possible values that are not countable. Since most application of the MCMC methods is with continuous distributions, this generalization is important. However, irreducibility and aperiodicity are not enough to secure the same Markov chain outcome with continuous distributions. We discuss the generalization first for the countable case, and then further refine the conditions for the continuous case. In this case, the transitional probabilities are

$$p_{ij} = \begin{Bmatrix} p, \text{if } j = i + 1 \\ r, \text{if } j = i \\ q, \text{if } j = i - 1 \end{Bmatrix}$$

That is, starting from state *i*, the process moves to *i*+1 with probability *p,* to *i*-1 with probability *q* and stays at *i* with probability *r*; *p*+*q*+*r*=1, *p, q, r*≥0. Figure 2.2 below demonstrates the outcome for an example of a random walk model with the first 500 values generated from a random walk with *p*=0.55>*q*=0.45, showing the process approaches +∞ in the sense that $p_{ij}^{(n)} \rightarrow 0$ for all *i, j*. This means starting from state *i*, the probability that any finite value of *j* will be reached approaches zero. To ensure that simulation by the Markov chain produces the invariant distribution in more general contexts beside the finite state spaces, the chain must have an additional property**.**

With the random walk of ***p>q***, all the states are transient, and none are recurrent because the process approaches infinity with probability of 1; the probability of returns to any state is not 1. To prevent such outcomes, all states must be revisited with probability of one; such a state is called a **recurrent state**, in contrast to a *transient state* that will not be revisited with some positive probability. Let the probability of event A starting at state *j* be p_j(A), then J is a recurrent state if :

$$p_j(X_n = j \text{ i.o})\text{=1}$$

where *i.o* stands for "infinitely often", namely the process returns to state j an infinite number of times with probability 1; otherwise, it is a transient state. In the random walk example, the process approaches + ∞ disappear; all states are recurrent if ***p=q*** as illustrated in Figure 2.3. In practice, a stronger condition, called **positive recurrent**, defined in terms of the time it takes for the process to make its first return to state j, is required to ensure recurrence.

To further define recurrence for continuous state spaces, let p_x(A) stand for the probability of event *A*, given the process starts at *x*. Then a *π*-irreducible chain with invariant distribution *π* is, for each *B* with *π*(*B*)>0:

$$p_x(X_n \in B \text{ i.o})\text{=0 for all } x$$

$$p_x(X_n \in B \text{ i.o})\text{=1 for } \pi\text{-almost all } x.$$

The chain is **Harris recurrent**, if $p_x(X_n \in B$ i. o)=1 for all *x*, the condition required for recurrence for a continuous process; that is, if a Markov chain for continuous distributions has the property of recurrentce, then the chain is Harris recurrent.

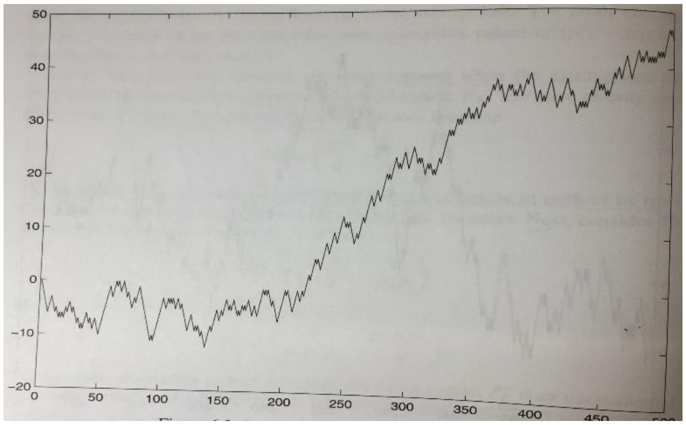


Figure 2.2- Random Walk with ***p***=0.55, ***q***=0.45

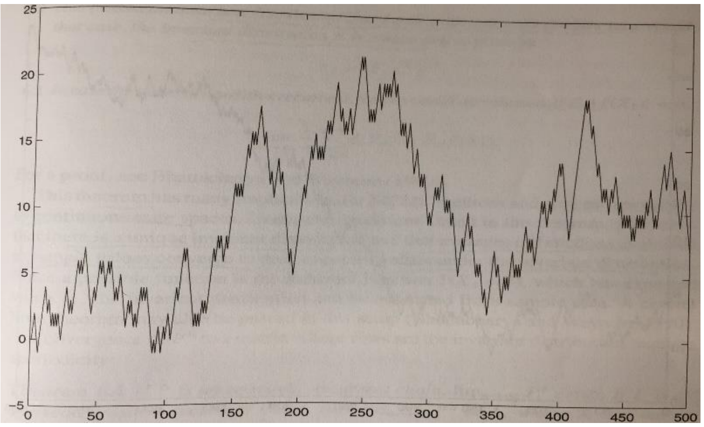


Figure 2.3-Random Walk with ***p*=*q***=0.5

If a Markov chain satisfies irreducibility and aperiodicity, then, for a large enough ***n***, the probability distribution of the drawings is the invariant distribution. This theorem has the important implication for simulation: if a Markov process is available for the target distribution, you can simulate from the Markov process to generate values for the target distribution. The aim of Bayesian applications is to obtain draws from the posterior distribution; a Markov chain draws the initial value of the parameter of interest from a sample of the transition kernel. Then, by a suitable method of drawing pseudo-random numbers, a new vector of values is drawn from the transition kernel evaluated at the initial values of the parameter. The process continues and at the ***n****th* stage draws are from the transition kernel of ***n*** -1 step. The Markov chain thus employed as ***n*** → ∞ is the limiting distribution of the posterior. Once convergence to the limiting distribution is reached, all subsequent draws are also from this invariant distribution, though they will be correlated.

18.3 *Simulation by MCMC*

Introduction

MCMC methods produce approximation to the invariant posterior when there is no analytically interactable solution ~~is~~ available for the exact posterior, but the question still remains as to how to find its distribution kernel. The two widely used Markov chain algorithms employed to find such an invariant distribution are the *Gibbs Sampler* and the *Metropolis-Hastings* (***MH***) algorithm. The MH is a general principle for finding such kernels while the Gibbs sampler is a special case of the MH. Let G stands for the number of simulations ~~that~~ (which can be very large, given the limit on computer capacity), and a larger ***G*** leads to a more accurate approximation, while ***n*** refers to the number of observations, fixed at the time the data are collected. For Bayesian inference, we denote the random variable of interest θ and the target, posterior distribution as $\boldsymbol{\pi}(\boldsymbol{\theta} \mid \boldsymbol{y})$ where ***y*** stands for the data.

i.*Gibbs Sampler*

Let $\theta = [\theta_1, \theta_2]^{/}$ have posterior density $P(\theta) = P[\theta_1, \theta_2]$ where we have suppressed dependence on y, $P(\theta \mid y)$, for convenience. If the posterior has no analytical solution, and the conditional densities of both $P(\theta_1 | \theta_2)$ & $P(\theta_2 | \theta_1)$ are known, then *alternating* the draws sequentially from $P(\theta_1 | \theta_2)$ & $P(\theta_2 | \theta_1)$, in the limit, converges to draws from the posterior $P[\theta_1, \theta_2]$. Such a MCMC procedure is known as the **Gibbs Sampler**. The employment of this method requires the ability to sample from every conditional distribution, given a non-standard joint distribution of the posterior $f(x_1, x_2)$ where the variables are in two conditional blocks $f(x_1 \mid x_2)$and $f(x_2 \mid x_1)$, with known simulation algorithms.

Algorithm for a two-Block Gibbs Sample:

1-Choose a starting value $x_2^{(0)}$

2-At the first iteration, draw

$$x_1^{(1)} \text{from } f(x_1 \mid x_2^{(0)})$$

$$x_2^{(1)} \text{from } f(x_2 | x_1^{(1)})$$

3-At the *g*th iteration, draw

$$x_1^{(g)} \text{from } f(x_1 \mid x_2^{(g-1)})$$

$$x_2^{(g)} \text{from } f(x_2 | x_1^{(g)})$$

And continue until the desired number of iterations for convergence to the invariant distribution ~~is~~ are met. Note that the starting value is not drawn from the invariant distribution, therefore, some portion of the initial sample, usually set at several hundred or several thousand and known as the **burn-in** sample, must be disregarded. There is no theory as to what the burn-in sample size should be, but for the number of iterations larger than that size, the distribution of the draws is approximately the target distribution, with G denoting the sample size after setting aside the first burn-in observations.

To demonstrate that the invariant distribution of the Gibbs kernel is the target distribution, let $x=f(x_1, x_2)$ be the values of the random variables at the start of algorithm iteration and $y=f(y_1, y_2)$ be the values at the end of the iteration. Then the Gibbs kernel is:

$$P(x, y)= f(y_1 \mid x_2)f(y_2 \mid y_1)$$

We can compute this kernel from

$$\int P(x, y)f(x)dx=\int f(y_1 \mid x_2)f(y_2 \mid y_1)\, f(x_1, x_2)\, dx_1\, dx_2$$

$$= f(y_2 \mid y_1)\int f(y_1 \mid x_2)\, dx_2$$

$$=f(y_2 \mid y_1)f(y_1)$$

$$= f(y)$$

The single integration in the first line reflects the *interdependence* of random value draws from each of the two blocks at each iteration.

In the second line, the final constant values come out of the integral, and x_1 is integrated out; in the third we integrate out x_2 to obtain finally the invariant distribution.

The extension of the Gibbs sampler to more than d>2 blocks requires the possibility of sampling from all the conditional densities $f(x_i \mid x_{-i})$ where x_{-i} are all the variables in the joint distribution other than x_i .

i.*Algorithm for a* ***d****-block Gibbs Sampler*

1-Choose a starting value $x_2^{(0)}, \ldots, x_d^{(0)}$

2-Draw

$$x_1^{(1)} \text{from } f(x_1 \mid x_2^{(0)},\ldots,x_d^{(0)})$$

$$x_2^{(1)} \text{from } f(x_2 \mid x_1^{(1)}, x_3^{(0)},\ldots,x_d^{(0)})$$

$$\vdots$$

$$x_d^{(1)} \text{from } f\,(x_d\,|x_1^{(1)}, \ldots, x_{d-1}^{(0)}\ )$$

3-At the *g*th iteration, draw

$$x_1^{(g)} \text{from } f\,(x_1 \mid x_2^{(g-1)}, \ldots, x_d^{(g-1)})$$

$$x_2^{(g)} \text{from } f\,(x_2\,|x_1^{(g)}, x_3^{(g-1)}, \ldots, x_d^{(g-1)}\ )$$

$$\vdots$$

$$x_d^{(g)} \text{from } f\,(x_d\,|x_1^{(g)}, \ldots, x_{d-1}^{(g)}\ )$$

For an example, consider bivariate normal data with uniform prior for the mean and known covariance matrix. Let $y{=}f\,(y_1, y_2\,){\sim}\ N\,[\theta, \sum\,]$, where $\theta = [\theta_1, \theta_2\,]'$ for a two-block sample and $\sum$ has diagonal entries as 1 and off- diagonal entries as ρ. Then, given a uniform prior for θ, the posterior can be shown to be a bivariate normal $\theta \mid y \sim N\,[\,\bar{y}, N^{-1}\sum\,]$.

The conditional posterior distributions are

$$\theta_1| \,\theta_2, y\ \sim N\,[\,[(\bar{y}_1 + \rho(\theta_2 - \bar{y}_2))\,, \tfrac{(1-\rho^2)}{N}\,].$$

$$\theta_2| \,\theta_1, y\ \sim N\,[\,[(\bar{y}_2 + \rho(\theta_1 - \bar{y}_1))\,, \tfrac{(1-\rho^2)}{N}\,]$$

We can use the above to simulate from each conditional normal distribution using updated values of $\boldsymbol{\theta_1}$ and $\boldsymbol{\theta_2}$, if the chain is long enough, then it will converge to the bivariate normal. Another example is the posterior distribution of the normal linear homoscedastic regression model, given normal-gamma conjugate priors, see Cameron and Trivedi (2005), p. 448. The conditional posterior of β given σ^{-2} is multivariate normal, and the conditional posterior of σ^{-2} , given β, is gamma. Though we can drive the posterior explicitly, it is easier to use the Gibbs sampler to draw a large sample from the joint posterior distribution. The chain consists of recursive draws from the normal conditional on the precision parameter σ^{-2} and from the gamma distribution conditional on the β.

The Gibbs sampler usually performs effectively but not in all contexts. If there is high correlation between one or more random variables in different blocks, the algorithm may “mix” poorly, that is the sampler draws disproportionately from some range of the sample space rather than its full support, thereby generating iterations from only a limited portion of the sample. As an example, consider a more simple version of the bivariate normal distribution or a joint function X=(X_1, X_2) distributed as N_2 (0, $\sum$) where :

$$\sum = \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}$$

If X_1, and X_2 are the two blocks of the Gibbs sampler, then similar to the example above, we have f (X_1| x_2)~ N [ρx_2, $1 - \rho^2$] & f (X_2| x_1)~ N [ρx_1, $1 - \rho^2$]. The algorithm does not work well if $\rho \approx$ 1since that implies the conditional variance of both variables [$1 - \rho^2$] is close to zero. In each iteration, the sampler generates values that are very close to the value of the previous iteration; implying the initial values $x_1^{(0)}$or $x_2^{(0)}$can have a big impact on the generated sample.

ii. *Algorithm of Metropolis-Hastings*

The ***MH*** algorithm is more general than the Gibbs sampler since it is a ***MCMC*** method that can be applied when the full set of conditionals are not available for sampling. First, we examine the ***MH*** algorithm in one block before considering a two-block ***MH*** algorithm.

We denote the current value of the random variables as ***x***, the next value ***y*** , and the invariant target distribution ***f*** **(.)**. The aim is to find a kernel $P(X, Y)$ with ***f*** **(.)** as its invariant distribution and the full conditional distribution ~~is~~ unavailable, so we cannot employ the Gibbs sampler. We call a kernel function ***q*** **(., .)** a **reversible kernel** if it enable us to write the target function in equivalent forms as:

$$f(x)q(x, y)=f(y)q(y, x)$$

If q is reversible, then

$$P(y \in A) = \int_A \int_{R^d} f(x)q(x, y)\, dx\, dy \qquad (18.3.1)$$

$$=\int_A \int_{R^d} f(y)q(y, x)\, dx\, dy$$

$$=\int_A f(y)dy$$

What this demonstrates is that f(.) is the invariant distribution for the reversible kernel distribution because the final probability of y is contained in the set A from which f (.) is computed. By contrast, an *irreversible* kernel for the values of a pair of random variables (x, y) is

$$f(x)q(x, y)> f(y)q(y, x)$$

This means that the kernel moves from x to y with greater probability than it moves from y to x. Let us define a probability mass function α (x, y) that captures this difference in probability. Multiplying both sides of the above inequality by α (x, y) would again equalize the equation as long as we account for that, the kernel moving to x from y with high probability. This can be done by setting α (x, y)=1 on the short side of the above inequality. Doing so will then allow the necessary modification to the kernel distribution as

$$f(x)q\,(x, y)\,\alpha\,(x, y) = f(y)q\,(y, x).\,1$$

from which we can obtain the definition of α (x, y)

$$\alpha\,(x, y) = \begin{Bmatrix} min\left\{\frac{f(y)q(y,x)}{f(x)q(x,y)}, 1\right\}, f(x)q(x,y) \neq 0 \\ 0 \qquad\qquad\qquad\qquad otherwise \end{Bmatrix} \qquad (18.3.2)$$

demonstrating that the *MH* algorithm based on (18.3.2) has a decision criterion similar to the Accept-Reject (*AR*) algorithm. Note that the starting value would be in the support range ~~of~~ for [f (.) q (., .)] distribution, therefore, we would not choose a y for which $q(x, y) = 0$, and [f (.) q (., .)] ≠ 0 is a problem. Also note that we do not need the unknown constant in the target distribution to compute α (., .) because it cancels out via the fraction $f(y)/f(x)$ in (18.3.2).

The expression q (x, y) α (x, y) has the following interpretation: if the process starts from x to generate y from the kernel $q(x, y)$, the move to y is with probability of α (x, y). If the move to y is rejected, the *process remains in x*. Here $q(x, y)$ acts similarly to the AR algorithm, but with an important difference. The AR algorithm continues to generate values until a draw is accepted while the MH algorithm by contrast returns the current state of the process as the next state when a draw is rejected and continues to the next iteration; this implies the *MH* values may be *repeated* in a simulation run. Note that (18.3.2) combines a continuous kernel $q(x, y)$ with a probability mass function α (x, y).

Using (3.2), we can summarize the MH algorithm:

1-Given x, generate Y from $q(x, y)$

2-Generate U from U [0, 1]; if:

$$U \leq \alpha\ (x,y) = \left\{ \begin{matrix} min\left\{ \frac{f(y)q(y,x)}{f(x)q(x,y)},1 \right\},f(x)q(x,y) \neq 0 \\ 0 \qquad\qquad\qquad\qquad\qquad\qquad otherwise \end{matrix} \right\}$$

Return to Y; otherwise return to x and go to step 1.The above is only the necessary condition for convergence of the MH kernel to the target distribution. The following states the full theorem.

Theorem 18.3.2 *Suppose P by (18.3.1) is a π-irreducible Metropolis kernel. Then P is Harris recurrent* (has positive recurrence with *p=q* for the continuous states *p & q*).

The implementation of the MH algorithm requires a choice for the proposal kernel ***q*** **(., .**) that provides well-mixed simulations. On the one hand, we wish to choose a proposal kernel that generates an agreeable probability of acceptance; on the other hand, by generating proposals that are close to the current point, the sampling will be confined to a limited section of the support, resulting in poor mixing. Two possible candidates to avoid poor mixing are the random-walk kernel and the independent kernel.

First, the random walk kernel generates the proposal y from the current value of x by the addition of *u*, a random variable, or a vector of such variables, *y=x +u*, by specifying a function for *u*. Since that distribution is symmetric, *h*(*u*)= *h*(- *u*), the kernel has the property that $q(x,y) = q(y,x)$, suggesting

$$\alpha\ (x,y) = \left\{ \begin{matrix} min\left\{ \frac{f(y)}{f(x)},1 \right\},f(x) \neq 0 \\ 0 \qquad\qquad\qquad otherwise \end{matrix} \right\}$$

Thus, with a random walk kernel, a move from x to y is certain if *f* (*y*)> *f* (*x*), but the probability of a move from a higher density to a lower density is with *f* (*y*)/*f* (*x*) less than with one.

Second, the independent kernel has the property $q(x,y) = q(y)$; meaning, the proposal density is independent of the current state of the chain.

$$\alpha\ (x,y) = \left\{ \begin{matrix} \left\{ \frac{f(y)/q(y)}{f(x)/q(x)} \right\},f(x)q(y) \neq 0 \\ 0 \qquad\qquad\qquad otherwise \end{matrix} \right\}$$

The probability of a move will be similar to that for the random walk kernel by replacing f (.) by [f (.) /q (., .)]

Example: Beta (3, 4) provides an example of an independent chain with U [0, 1]as the proposal density with the following algorithm:

1-set $x^{(0)}$ equal to a number between 0 and 1

2-At the *gth* iteration (after the burn-in sample), generate U_1 and U_2 from U [0, 1]

3-If

$$U \leq \alpha\ (x^{(g-1)}, U_2) = \frac{U_2^2(1 - U_2)^3}{(x^{(g-1)})^2(1 - x^{(g-1)})^3}$$

Set $x^{(0)} = U_2$, otherwise set $x^{(0)} = x^{(g-1)}$.

4-Go to 2 and continue until the desired number of iterations is achieved.

Fig. 3.3 shows the results for 5,000 iterations, after the first 500, with a good fit between the generated values; the acceptance probability is 0.57, meaning 57 percent of the proposals were accepted. The mean of the sample is 0.4296 compared to the theoretical mean of 3/7=0.4286.

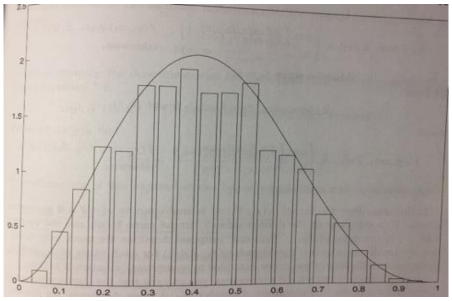


Fig. 3.3 MH simulation sampling of *Beta* (3, 4) with U (0, 1) proposal

MH algorithm with two blocks

We can find a suitable proposal more easily if the target distribution has two blocks f (X_1, X_2). Consider the state (x_1, x_2)

1-Let the state at the *gth* iteration be (x_1, x_2), and at the *gth*+1 iteration be (y_1, y_2) and draw

Z_1 from q_1 ($x1$, Z_1 |x_2) & U_1 from U (0, 1)

2-If

$$U \leq \alpha\ (x_1, Z_1 |\ x_2) = \begin{cases} min\left\{\frac{f\left(Z_1, x_2\ \right)q_1(Z_1, x_1|x_2)}{f\left(x_1, x_2\ \right)q_1(x_1, Z_1|x_2)}, 1\right\}, f(x_1, x_2)q_1(x_1, Z_1|x_2) \neq 0 \\ 0 \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad otherwise \end{cases}$$

Return *y₁=Z₁*, otherwise return *y₁=x₁*.

3-Draw

Z_2 from q_2 (x_2, Z_2 |y_1) & U_2 from U (0, 1)

$$U \leq \alpha\ (x_2, Z_2 |\ y_1) = \begin{cases} min\left\{\frac{f\left(y_1, Z_2\ \right)q_2(Z_2, x_2|y_1)}{f\left(\ y_1, x_2\ \right)q_2(x_2, Z_2|y_1)}, 1\right\}, f(y_1, x_2)q_2(x_2, Z_2|y_1) \neq 0 \\ 0 \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad otherwise \end{cases}$$

Return to y_2=Z_2, otherwise y_2=x_2..

In this two-block algorithm, the kernel $q_1(x_1, Y_1|x_2)$ acts like the kernel proposal q (*x*, *Y*) to generate a value Y_1 conditional on the current value x_1 in the same block and the current value x_2 in the other block; the new densities are specified for $q_1(x_1, Z_1|x_2)$ and $q_2(x_2, Z_2|y_1)$ for each value of x_2 and Y_1.

We can now show the Gibbs sampler is a special case of the *MH* algorithm. Consider *α*(., .) when the kernel for moving from the current value of x_1 to the proposal value of Y_1 is the conditional distribution $f(y_1 \mid x_2)$, assumed available for sampling. Then

$$\frac{f\left(Y_1, x_2\ \right)q(Y_1, x_1|x_2)}{f\left(\ x_1, x_2\ \right)q(x_1, Y_1|x_2)} = \frac{f\left(Y_1, x_2\ \right)f(x_1|x_2)}{f\left(\ x_1, x_2\ \right)f(Y_1|x_2)}$$

Since $f(Y_1|x_2) = \frac{f(Y_1, x_2)}{f(x_2)}$ and $f(x_1|x_2) = \frac{f(x_1, x_2)}{f(x_2)}$, then it must be the case that $\alpha\ (x_1, Z_1 |\ x_2)$ =1, namely, that the Gibbs algorithm is an *MH* algorithm where the proposal is always accepted. We can still apply the Gibbs Sampler to any blocks of the *MH* algorithm where conditional distributions are available to sample from, leaving the *MH* algorithm to be employed for finding suitable proposal densities and accepting them with probability $\alpha\ (x, y)$.

Readings

Greenbeg (2014, Part II), Cameron and Trivedi (2005, chapter 13). Casella and George (1992) discuss the Gibbs sampler, Geweke (1989), that of the MCMC simulation.

Chapter 18 Bayesian Simulation Exercises

Q18.1 (*Inverse Transfer*) Consider the exponential density with the density function

$P(x|\theta) = \theta^{-1} exp(-x/\theta)$, $x > 0$

Define $X = F^{-1}(U)$ where $U \sim U(0, 1)$ is a uniform random variable on the unit interval.

(a) Show that the inverse transformation method can be used to generate draws from the exponential density.

(b) The logistic density function is given by

$$P(x) = \frac{\exp(-[x-\mu]/\sigma)}{\sigma(1+\exp(-\frac{[x-\mu]}{\sigma}))^2}, \ -\infty < x < \infty, \ -\infty < \mu < \infty, \ \sigma > 0$$

Q18.2 (*Importance Sampling*) Consider estimating a posterior moment of the form $E[f(\theta)|y]$.

However, direct sampling from the posterior is often unavailable. Instead, suppose

we wish to estimate the expectation of a function of some multivariate random variable X,

$E_P[f(x)]$, where $E_P[f(x)] \equiv \mu = \int_p f(x) P(x) dx$, with $\mathcal{P} \equiv \{x: \mathcal{P}(x) f(x) \neq 0\}$ and assume direct sampling is not possible.

(a) Suppose you can generate samples from some approximating density, $I(x)$, by importance sampling method from a collection of M simulations, $x_1, \ldots, x_m$ from I. Explain how you employ these simulations to obtain the desired moment from the following estimator where IS denotes the importance sampling estimator:

$$\hat{\mu}_{IS} \equiv E\widehat{\mathrm{P}[f(x)]} = \frac{1}{M} \sum_{m=1}^{M} \frac{\mathrm{P}(x_m) f(x_m)}{I(x_m)},$$

discuss the conditions for the choice of I.

(b) Derive the variance of $\hat{\mu}_{IS}$ and describe how it can be estimated; note that the "sample size", the number of draws from $I(.)$, is under your control.

Q18.3 (*Accept-Reject sampling*) Consider drawing from a density $f(x)$ defined over the compact support $a \leq x \leq b$:

1-Generate two independent uniform variables U_1 and U_2 as follows:

$U_i \sim^{\text{iid}} U(0, 1)$, $i = 1, 2$

2-Let M≡ $max_{a\leq x\leq b}\, f\,(x)$ if $MU_2 > f\,(a+[b\text{-}a]U_1)$, go back to the first step and generate new values for U_1 and U_2, and again determine if $MU_2 > f\,(a+[b\text{-}a]U_1)$. If $MU_2 \leq f\,(a+[b\text{-}a]U_1)$, set $x= f\,(a+[b\text{-}a]U_1$ as a draw from f (x).

(a) What is the probability that any specific application of this algorithm will produce a draw that is accepted?

(b) Sketch a proof as to why x, when it is accepted, has the distribution function $F(x)=\int_a^x f(t)dt$.

Q18.4 (***AR generalization***) Suppose we are interested to generate draws from a target density

$f\,(\theta)$ with support $\mathbf{\Theta}$ but with an unknown normalizing constant and suppose there is some approximating proposal density s(Θ)with support $\boldsymbol{\Theta}^*$ such that $\boldsymbol{\Theta\subseteq\Theta}^*$. Write the kernels of both the proposal and target densities as

$f\,(\theta)=c_f\tilde{f}(\Theta)$& $s\,(\theta)=c_s\tilde{s}(\Theta)$

where $\tilde{f}$& $\tilde{s}$ respectively denote the target and proposal kernels, and c_f& c_s the associated normalizing constants; let $\widetilde{M} = sup_{\Theta\in\Theta}(\frac{\tilde{f}(\Theta)}{\tilde{s}(\Theta)})$and consider the following algorithm:

1-Draw U uniformly on [0, 1], namely $U\sim U(0, 1)$.

2-Draw a candidate from the proposal density s(Θ), namely, $\Theta^{\text{cand}}\sim$ s(Θ)

3-if $U \leq \frac{\tilde{f}(\Theta^{cand})}{\widetilde{M}\tilde{s}(\Theta^{cand})}$, then set $\Theta = \Theta^{\text{cand}}$ as a draw from $f\,(\theta)$; otherwise return to the first step and repeat until step 3 is satisfied. Show this algorithm includes the line in Q18.3 as a special case.

Bayesian Computer Exercises

i.*A Markov chain regression*

Q18.5 The Markov chain provides the basis of Bayesian simulation, but its application plays a pivotal role in many classical fields, most prominently in State-space econometrics, as exercises here demonstrate. Download the data set on *usmacro.dat*.

a. Fit a *Markov* regression with *dr* (for quick adjustment; use *mswitch* command) for *fedfunds* and comment on the outcome.

b. Fit a *mswitch MSDR* with switching coefficients and lag

Q18.6 Download *snp500.dta*.

a. Fit *mswitch* model allowing for non-constant variance across 2 states of switching variances; provide comments

Q18.7 Download *rgnp.dta*, real growth data set.

a. Fit *mswitch* with *ar* option for a *MSAR* to growth of gdp with lags=1/4 over 52q1-84q4 period; comment on the outcome

b. Fit mswitch lags=1/2 with switching coefficients with comments

c. Fit Markov *ar* regression with constraints & comment on the outcome

ii. *Basics of Bayesian regression*

Q18.8 Download *oxygen.dta*, Oxygen uptake data set.

a. Regress *change* on *group & age* by OLS and compare with *bayesmh* command, using normal likelihood, flat prior for change, and Jeffreys' prior, (1/σ^2) density, for variance; comment on the outcome.

b. Fit the same model with informative conjugate normal prior for parameters conditional on inverse gamma prior (2.5, 2.5) for variance.

c. Check the model's diagnostics for convergence to an invariant posterior to ***i***) change, ***ii***) all parameters, ***iii***) sample-size related stats, ***iv***) variance parameter.

Q18.9 Download *usmacro.dat***.**

a. Fit regression to Q18.5 model, this time by *bayes*' command and provide comment.

b. Fit the above with the Gibbs option and provide comment

c. Alter the *bayes'* default priors by specifying prior () with *change* using Zellner prior (3, 12) and variance using inverse gamma (0.5, 4).

iii. *Further Applications with bayes' command*

Q18.10 Download *heartsbwitz.dta*, a heart disease study data set.

a. Fit Bayes’ logistic regressions and provide comment

b. Fit Bayes’ logistic regressions, this time with _asis_ option to prevent dropping the variables, and _nomleinitial_ option to prevent use of initial ML estimates; provide comment

*****_Bayes’ survival regression_

Q18.11 Download hip3.dta, a hip replacement study data.

a. Fit ~~a~~ _streg_ and _bayes_’ survival Weibull PH models

b. Obtain diagnostic for male and provide comment

***** _bayes’ Panel data_

Q18.12 Download _pig.dta_, data on pigs weight gain study.

a. Fit a two-level random-intercept, random-coefficients panel regression and _Bayes’_ panel regression; provide comm**ents

b. Fit a bayes panel model in Q18.12_a with default output, without _melabel_ option, and explain its components.

c. Fit _Bayes’_ random-coefficient model to allow different (pigs) growth rates; provide comments

***** ba*yes Autoregressive models*

Q18.13 Download _yd.dta_ on log of US income.

a. Fit _bayes’ AR_(1)model for log(yd).

b. Use a uniform (-1, 1) prior to ensure AR(1) stationary assumption & provide comment

c. Select the best _bayes_’ AR model by comparison of AR(1/5)models

CHAPTER 19 Linear Machine Learning Models for Prediction and Inference

19.0 Machine Learning and Econometrics. Statistical Machine Learning (**ML**) deals with *bid data* using machine, i.e. computer. That often means when the number of regressors, also called *features*, *inputs* or *predictors* in ML, are larger than the number of observations. In that context, the usual traditional econometrics models, especially the least squares-based models, become inapplicable. The ML approach have made impressive strides in developing models capable of quite accurate prediction for a response variable from a large number of predictors by exploiting very large data sets. Since prediction also occupies an important part of econometrics, there have been new attempts to adapt econometrics predictive models employing ML. However, inference rather than prediction is the main focus of econometrics, and this raises the question of the place of the ML approach in econometrics models and tools. The focus of most ML models so far has been on predictions of y from x; coefficient estimation has so far received much less attention in the ML literature. By contrast, the main focus of econometrics has been in developing a large body of work addressed to obtaining good (consistent and efficient) parameter estimates β from the underlying relationship between y and x. While econometrics has benefitted much from ML approaches to prediction, the models of ML inference are not as developed, and when available, as examined in the last section of this chapter, they are rarely consistent and often lack the type of asymptotic theory that justify consistency of classical econometric models. This contrast suggests ML can be employed in econometrics provided the application is for relevant $\hat{y}$ tasks, that is, if the application goes through a ML prediction model or involves a predictive step, given very large big data inputs. One type uses new data to answer older questions; for example, predicting measuring change in poverty from satellite images; another is when the inference requires prior prediction as prominently employed by two-stage least squares for consistent coefficient estimates into which the first stage prediction enters as inputs, also examined later in this chapter. Therefore, most of ML models examined in this and particularly the next chapter on non-linear ML, focus on the accuracy of the predictive estimation that requires methods to reduce the scope for *overfitting* them. We discuss the ML linear models in this chapter and nonlinear models in the next chapter.

19.1-*Regression Overfitting.* We estimate a function by linear and non-linear methods that share common features. As an example, with n=30 data points i=1, 2, . . . ,n that teach the method what type of function f to employ for estimation with j=1, 2, . . . , p predictors, each represented by x_{ij}

with corresponding response observations$y_1, y_2. . . , y_n$. The sample employed trains the *in-sample* prediction using the *training* sample; the prediction accuracy on a different, *out-of-sample* data set. The goal is to find a function for *f.* either parametrically by linear methods, or non-parametrically by a non-linear method. Fig. 19.1 below represent three different methods to estimate a function such that $Y \approx f(X)$. F. 19.1 shows three different methods of estimating the relationship between X and Y; p+1 coefficients $\beta_0, \beta_1, . . . ,\beta_p$ by the least squares method.

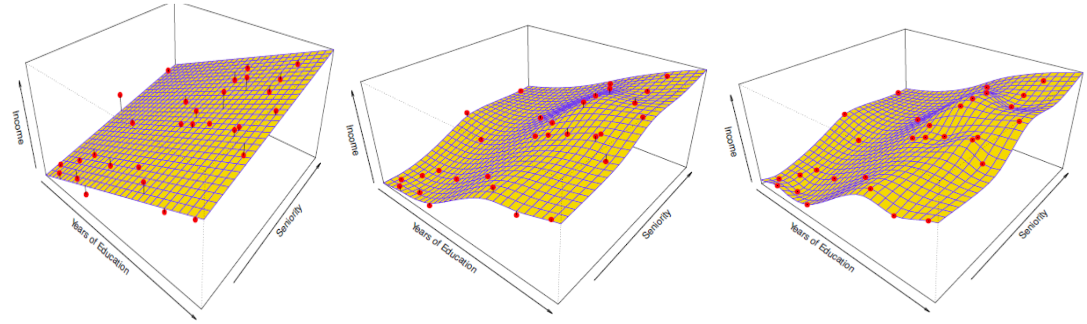


Fig. 19.1 *Linear fitting with different degrees of Smoothing*

Fig. 19.1 Plot on the right assumes a linear model fitted to the training *income* data observation set (in red) of US Atlantic region of wage for men as a function of age and seniority to estimate their coefficients that are unbiased with minimal inaccuracy. The center non-parametric *thin-plate spline* plot estimation assumes no functional form of *f* function in order to get as close as possible the data points to improve the fit. for age and seniority with a function f that attempts to be as close as possible to the data points. The left plot demonstrates the same as the center using less smoothness with a rougher plot to obtain a perfect fit! While the first lacks sufficient accuracy indicated by the distance between observations and the true model (the yellow surface), the third *overfits* the model, picking up on too many unique features of the sample at hand. resulting in poor prediction when tested on a new, as yet unused set of wage data; the second is a compromise between the degree of smoothness and prediction accuracy.

Fig. 19.1 provides a contrast in the applications of the two most commonly used methods of the parametric linear least squares and the non-linear, non-parametric *k*-nearest-neighborhood method. The linear regression fit with a *continuous quantitative* output variable predicts the outcome Y from

$$\hat{Y} = \hat{\beta}_0 + \sum_{j=1}^{p} X_j\hat{\beta}_j \qquad (19.1)$$

The estimated coefficient values $\hat{\beta}_j$ obtained from the gradient $f'(X) = \beta$, a vector of inputs; the intercept estimate, $\hat{\beta}_0$ in machine learning is called the *bias*, and chosen so as to minimize the residual sum of squares

$$\text{RSS}(\beta)=\sum_{i=1}^{N}(y_i - x_i^T\beta)^2$$

However, the output can also be a *discrete, qualitative* type, varying as a *classification;* binary or categorical variables. Scatterplot 19.2 is a classification with a simulated training data set with the class variable G coded as blue for 0 or orange for 1. It fits a linear r regression by converting the continuous Y into a class variable according to

$$\hat{G} = \begin{cases} \texttt{ORANGE} & \text{if } \hat{Y} > 0.5, \\ \texttt{BLUE} & \text{if } \hat{Y} \leq 0.5. \end{cases}$$

Fig. 19.2 shows the outcome obtained with the training data with a binary two-class Y regression we note some instances of misclassification on either side of the decision boundary; such errors indicate the linear model rigidity. Let us now examine the regression/classification outcome with the more flexible nearest-neighborhood method, discussed in 11.2.

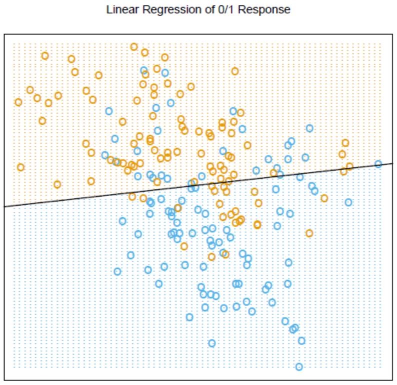


Fig. 19.2-Linear Boundary Classification

The training data set is now divided into *k*=15 nearest neighbor regions; each region defined by the 15 closest points x_i in the training data and obtained corresponding$\bar{y}_i$; in general, with $N_k(x)$ k closest points to $x_i$we fit

$$\hat{Y}(x) = \frac{1}{k}\sum_{x_i \in N_k(x)} y_i \qquad (19.2)$$

Fig. 19.3 shows the proportion of $\bar{y}_i$ in the neighborhood of 0 and 1and demonstrates $\hat{G}$ for the binary classification G. The decision boundary is now more irregular because it responds to local values by choosing where the predicted values by majority tend to dominate. Here the number of k corresponds to the p parameters of the linear model; the *effective* number of parameters *N/k* is usually larger than p and deceases as we increase k because there would then be *N/k* nonoverlapping with a one parameter (a mean) fit in each k region.

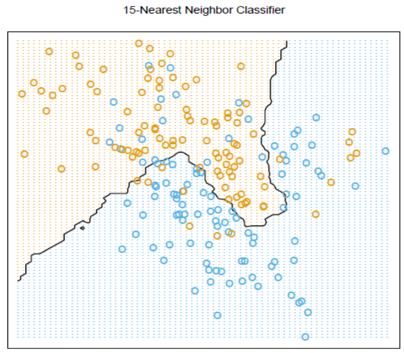


Fig. 19.3- Non-Linear Boundary Classification

We note that there are now far fewer misclassified observations, and we cannot apply this method to the training data sum-of-squared errors since we would always obtain *k*=1! The linear model is smooth and tends to be stable but this is substantially the result of a linear decision boundary criteria. We call this linear model one with low variance but with potentially high bias. By contrast, the *k*=nearest neighborhood makes a few strong assumptions but the outcome is non-smooth and unstable=high variance and low bias. Therefore, we can increase the number of k to obtain increasingly improved fit to the training data but then the price is higher variance. There are also some methods that generalize the nearest neighbor approach. *Kernel* methods employ weights that decrease smoothly to zero with distance from the target instead of 1/0 weights used in the neighborhood approach and in high dimension p, the Kernel emphasizes some p more than others; locally weighted linear fits local rather than constant weight least squares; and the projections pursuit and neutral net models discussed latter rely on the sum of non-linear transformation of the linear model.

The boundary decision theory requires a loss function for penalizing prediction errors based most commonly on squared errors loss function $L(\mathrm{Y}, f(X))=(Y-f(X))^2$ with choice to criteria

$$\mathrm{EPE}(f) = \mathrm{E}(Y - f(X))^2$$

$$= \int [y - f(x)]^2 \, \Pr\,(dx, dy)$$

That leads to minimization of the expected error prediction by solving $f\,(x)$=E(Y|X=x). The nearest-neighborhood does this directly using the training data by averaging all y_i with input x_i=x

$$\hat{f}(x) = \mathrm{Ave}(y_i | x_i \in N_k(x))$$

Two approximations are possible: expectation by averaging over the sample or conditioning on some region near the target. The linear model does not condition on X, its averaging over the sample comes from assuming that the model is additively linear in its predictors. We can expand the binary classification regression with a categorical G function or 0-1 loss function to minimize expected errors respectively by

$$\mathrm{EPE} = \mathrm{E_x} \sum_{k=1}^{K} L[\,\mathcal{C}_{\mathrm{k}}, \hat{G}(x)]\Pr\,(\mathcal{C}_{\mathrm{k}}|X)$$

$$\hat{G}(x) = argmin_{g \in \mathcal{C}}[1 - \Pr\,(g|X = x)|$$

where $\hat{G}(x)$ is obtained by minimizing the length of the regressors. The solution, known as the *Bayes classifier*, is

$$\hat{G}(x) = \mathcal{C}_{\mathrm{k}}\; if \Pr\bigl(\mathcal{C}_{\mathrm{k}}\big|X = x\bigr) = max_{g \in \mathcal{C}}\Pr\,)g = X = x)$$

The Bayes criteria allocate to the most probable class, using discrete distribution Pr(G|X); the error rate of this classifier is called the *Bayes rate*.

If we condition the mean estimation of Y on the loss function rather than the predictors, that is on E|Y - f(X), the solution in that case is $\hat{f}(x) = median(Y|X = x)$, a more robust solution than the conditional mean. We should mention that given finite N, we must impose as a restriction the RSS criterion, one based on the number of parameters, and on the unspecified function f(.)

$$\mathrm{RSS}\;(f) = \textstyle\sum_{i=1}^{N}(y_i - \; f(x_i))^2$$

Often some restriction on the regression equations in a small data location such as a linear or low-order polynomial function fitted in that neighborhood means_the larger the size of the neighborhood, the stronger the constraint and more sensitive the solution to the choice of a constraint.

19.2- Model Selection by MSE.

Model selection and assessment are closely related to model complexity. If the data is quantitative, or nearly so, e.g. with interval scale response, then the training data error loss function estimation between Y and $f\,\hat{}\,(X)$, L(Y, $f\,\hat{}\,(X)$), is typically based on either squared error or absolute error, as discussed in chapter 6. The average training error sample loss function is then given by

$$\overline{err} = \frac{1}{N}\sum_{i=1}^{N} L(y_i, \hat{f}(x_i)) \qquad (19.3)$$

The training error measurement should then be compared with the model expected test error in order to select and assess the best model complexity. This is because as the training data uses more data adapted to more complex underlying features of the data in order to decrease estimation bias, it will increase in variance; clearly a zero-training error is an overfit and will have poor expected test error. We seek some middle level of complexity that leads to minimum expected test error employing a *tuning parameter* α to minimize error by controlling complexity. This argument applies similarly with ~~a~~ qualitative or categorical response data, using either a 0-1 loss function or the probability $p_k(X) = Pr(G = k|X)$ to obtain misclassification error; typically, we employ the log-likelihood loss function for the Poisson, gamma, etc. Given large data, we divide the set into three parts: training, validation and test sets; larger for the first, e.g. 50%, the rest as 25% and 25%. For the situations with inadequate data, there are two approaches to approximate the validation step analytically by AIC and BIC discussed in chapter 7, or by sample iteration with cross-validation and the bootstrap. The point to note is that the bias-variance tradeoff differs for the 0-1 as compared with the squared error loss, hence, t best choice of tu~~r~~ning parameters result in very different error measurements for the two loss functions.

Error minimization. in regression or classification this is achieved by obtaining the best trade-off between the squared error and variance by decomposing them. Start with assuming Y=f (X) +ε where E$(\varepsilon) = 0$ & Var$(\varepsilon) = \sigma_\varepsilon^2$. Then the expected loss error at x=x_0 decomposes

$$Err(x_0) = E[\left(Y - \hat{f}(X^0)\right)^2 |X = x_0$$

$$= \sigma_\varepsilon^2 + \left[E\hat{f}(x_0) - f(x_0)\right]^2 + E\left[\hat{f}(x_0) - E\hat{f}(x_0)\right]^2$$

(19.2.2)

$$=\sigma_\varepsilon^2 + Bias^2\hat{f}(x_0) + Var\hat{f}(x_0)$$

$$=\text{Irreducible Error} + \text{Bias}^2 + \text{Variance}$$

Usually, the more complex the model, the lower the estimation bias but the higher the variance.

Optimism. A clearer approach to lowering error measurement is by fixing the training data set *T* and allowing other quantities to change, and the error will be average over the training sample

$$\text{Err} = E_{\mathcal{T}} E_{X^0 y^o}[L(Y^0, \hat{f}(X^0) | \mathcal{T}] \quad (19.4)$$

where $E_T$stands for the training sample error, and E_{rr} is estimated directly from the training sample to avoid difficult estimation of E_T, thus the average over the training sample given by

$$\overline{\text{err}} = \frac{1}{N}\sum_{i=1}^{N} L(y_i, \hat{f}(x_i) \quad (19.5)$$

(19.5) will be smaller than the training error because the model adapts to the training data to produce too optimistic an estimate of the training error. To measure optimism, first define *in-sample* error for Y_0 at the end of each by

$$Err_{in} = \frac{1}{N}\sum_{i=1}^{N} E_{y^o}[L(Y_i^o, \hat{f}(x_i) | \mathcal{T}]$$

Then define optimism *OP* as the difference between the in-sample and training error averages:

$$\text{op} \equiv \text{Err}_{\text{in}} - \overline{\text{err}}$$

Typically, a positive measure since $\overline{\text{err}}$ is generally biased downward; its average given by

average optimism is the expectation of the optimism over training sets

$$\omega \equiv \text{E}_{\text{y(op)}} = \frac{2}{N}\sum_{i=1}^{N} \text{Cov}(\hat{y}_i, y_i) \quad (19.6)$$

Therefore, the harder the fit, the greater the covariance, resulting in greater optimism. We can then improve estimation of predicted error by adding an estimate for optimism to the training error err^{-} by starting with the key equation

$$\mathrm{E_y Err_{in}} = \mathrm{E_y}(\overline{\mathrm{err}}) + \frac{2}{N}\sum_{i=1}^{N} \mathrm{Cov}(\hat{y}_i, y_i) \quad (19.7)$$

Modifying for a linear fit $Y = f(X) + \varepsilon$ with *d* basis linear functions or inputs

$$\sum_{i=1}^{N} \mathrm{Cov}(\hat{y}_i, y_i) = d\sigma_\varepsilon^2 \quad (19.8)$$

(19.8) is the basis for the definition of the effective number of parameters discussed above. AIC and BIC work with this method, by adding an OP component to the training error $\overline{err}$. By contrast, cross-validation and bootstrap methods are direct estimates of the extra-sample error.

AIC. The in-sample error estimate with an average OP estimate is

$$\widehat{Err}_{in} = \overline{err} + \hat{\omega} \quad (19.9)$$

That applied to a loss function with *d* parameters and an estimate of the noise variance $\hat{\sigma}_\varepsilon^2$ leads to

$$C_p = \overline{err} + 2.\frac{d}{N}\hat{\sigma}_\varepsilon^2$$

Hence, we adjust the training error by a factor proportional to the number of *d* (basis functions). The Akaike AIC is similar and uses a log-likelihood loss function

$$\mathrm{loglik} = \sum_{\mathrm{i=1}}^{\mathrm{N}} \mathrm{logPr}_{\hat{\theta}}(\mathrm{y_i})$$

WherePr_θ(*Y*) is a family of densities for Y, including the "true" density, and $\theta^{\wedge}$ is the ML estimate of θ and *loglik* is the maximized log-likelihood. For example, the logistic regression for binomial log-likelihood is

AIC= - 2/*N* .loglik + 2.*d/N*

For the Gaussian model with known constant variance, C_p is equivalent to AIC; the AIC criterion selects a model with the smallest AIC value over the set of all models.

BIC. The Bayesian is also implemented by maximizing a log-likelihood by

BIC= - 2.loglik+(log *N*).*d* (19.10)

The BIC statistic times 1/2 is known as the Schwarz criterion; under the Gaussian constant variance assumption, the BIC has a known Gaussain constant variance, BIC is written as

$$\text{BIC} = \frac{N}{\sigma_\varepsilon^2}[\overline{err} + (logN).\frac{d}{N}\sigma_\varepsilon^2]$$

demonstrating that BIC is proportional to AIC with a factor of AIC (C_p) with 2 replaced by log N. With N >≈ 7.4, BIC penalizes the complex model more heavily in preference to a simpler model. However, the relationship of the BIC to the Bayesian approach shows the difference between AIC and BIC. Suppose we have M_m number of potential models to choose from each with corresponding parameter θ_m for *m*=1, 2, ... , *M*; given a prior *Pr*($\theta_m|M_m$) ~~and~~ for each model results in the posterior probability with Z representing training data $\{x_i, y_i\}_1^N$as

$$\Pr(M_m|Z) \propto \Pr(M_{m)}.\Pr(Z|M_m)$$

Use the posterior odds to compare two models

$$\frac{\Pr(M_m|Z)}{\Pr(M_\ell|Z)} = \frac{\Pr(M_m)}{\Pr(M_\ell)}.\frac{\Pr(Z|M_m)}{\Pr(Z|M_\ell)}$$

The last quantity on the right represents the contribution of the data to the posterior odds and is known as the *Bayes factor*.

$$logPr(Z|M_m) = logPr(Z|\hat{\theta}_m, M_m) - \frac{d_m}{2}.logN + O(1) - 2logPr(Z|\hat{\theta}_m, M_m)$$

This is equivalent to the BIC criterion, and minimizing it leads to choosing a model with the approximately largest posterior. BIC is asymptotically consistent and will select the correct model as N →infinity. That motivates the difference with the AIC which selects *more complex* models as N →infinity, though with finite sample, the BIC selects models too simple compared to the AIC.

19.3 *Orthogonality condition.*

Machine learning models are easily interpretable, still applicable to transformations of the original predictors, generalizations that are called the *basis-function* approach, for prediction that approach can prove more effective than non-linear models; the latter are in fact generalizations of the linear models.

The linear model assumes a linear regression function E(*Y*|*X*); X predictors have several varieties: quantitative ones and their transformations, basis-function expansions, and different levels of qualitative inputs, and interactions between predictors. The least squares method assumes inputs orthogonality. Coefficient estimates do not affect each other, but real data are never orthogonal, and therefore least squares orthogonality must be enforced. Suppose a p column of input data matrix X consisting of $x_1, x_2, . . . , x_p$vector of *N* ones orthogonal to each other, let 1=x_0 and define $\bar{x} = \sum_i x_i/N$. This simple regression involves two steps: i) regress x on 1 to obtain the residual z=x – $\bar{x}$.1; ii) regress y on z to obtain the coefficient $\hat{\beta}_1$. In this approach, the simple univariate "regression of b on a with no intercept "orthogonalizes" b with respect to a for the case of two inputs x_1&x_2. First the vector of x_2 is regressed on the vector of x_1; since the two vectors are orthogonal to each other as shown in Fig. 19.4, that leaves only the residual vector *z*; then the regression of y on *z* results in the multiple regression coefficient on x_2. Adding the projections on each $x_1$and *z* results in the OLS fit $\bar{y}$.

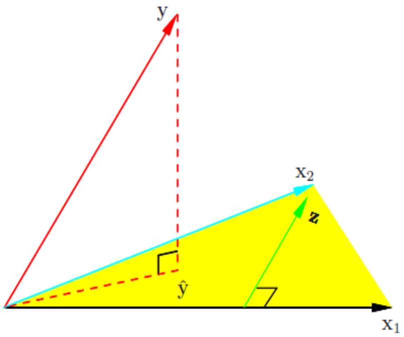
The generalization to *k* multiple outputs $Y_1, Y_2, . . . , Y_k$ is possible with corelated errors ($\varepsilon_1, \varepsilon_2, . . . , \varepsilon_k$) if we modify a RSS of a multivariate Gaussian regression weighted by Cov(ε)=**Σ**

$$RSS(\mathrm{B};\, \Sigma) = \sum_{i=1}^{N} (y_i - f(x_i))^T \sum_{\square}^{-1} (y_i - f(x_i))$$

The procedure is known as the Gram-Schmidt multiple regression. Orthogonalization is a powerful idea and we will return to its role with reference to ML inference by partialization in 19.4 below.

19.4 ***Selection by regressor exclusion***. The least squares regression results may be inadequate either because they are not sufficiently accurate and require some kind of shrinking method to reduce variance at the expense of acceptable sacrifices for bias, and/or because interpretation with a large number of inputs often requires selection of a smaller number of inputs with the strongest impact in favor of sacrificing less important details.

Fig. 19.4-Orthogonalization



The *Best subset* regression identifies a subset of inputs $k \in \{0, 1, 2, \ldots, p\}$ that produces the smallest sum of residual error; the choice of k requires an assessment of the bias-variance tradeoff discussed above and typically involves the smallest model that minimizes an estimate of the predicted expected error. Here we examine a number of linear regressions which prove that. Selecting inputs in steps by adding and excluding them sequentially provide procedures to achieve that goal. If we use *Stepwise Model Selection* as the number of *p* increases, the selection process through all possible alternatives quickly becomes impractical (typically once *p* > 40); therefore, we need a feasible search method. *Forward-stepwise selection* begins the search with the intercept and then adds sequentially to the model a new predictor to lower its predicted error. This is a *greedy* process in that it works by adding predictors until the fit stops improving; though we cannot compute the best subset, we can compute stepwise sequences for all *p* including when *p* ≥*N*. By contrast, *Backward-Stepwise selection* begins with the *p* full model and excludes the predictors that have the least impact on improving the fit; hence it only works with *N*>*p*. The outcome tends to be similar by either process and some software combine both at each step so as to minimize the AIC score. A more constrained selection is by *forward step stage* (FS) regression that begins like stepwise by at each step identifying the variable most correlated with the current *residual* and adding that to the current coefficient of that variable; the process continues until no variable is correlated with the residual; that is, we fit the least squares model when *N*>*p*. Since at each step only the current variable is adjusted, all other variables remain unadjusted, this selection method can take many p steps to produce the least squares fit. However, the "slow fitting" regression is advantageous in high-dimensional applications see below.

19.6 *Regularization & Shrinkage Estimators.* The linear ML models frequently employ shrinkage estimators that reduce the number of main variables of interest to a relatively small number. In this section, we examine the most common linear shrinkage models, namely Lasso, Ridge, elastic net, Adaptive Lasso and Smoothly clipped Absolute Deviation (SCAD) and compare their predictive performance when employed as linear regression econometric models. These models have become increasingly popular when the traditional estimators such as the OLS are no longer applicable, in particular when the number of regressors are larger than the number of observations. In that context, shrinkage estimators offer an approach to identification of a set of relevant variables from a much larger pool of covariates, known ~~as~~ variously as *predictors, features, or regressors*, based on the fundamental assumption of *sparsity* that the number of non-zero coefficients are relatively small. We should note at the outset that unlike the main focus of econometrics on parameter inference, the focus of machine learning shrinkage models has been on obtaining prediction by the best approximation for the response (dependent, or endogenous) variable, therefore side-stepping parameter significance and interpretation of the regressor coefficients. Machine learning models usually attempt to reduce the number of model predictors by separating the zero coefficients from the non-zero. The econometrics approach, however, is also addressed to the distinction between true zero observations in the data generating process and those that appear close to zero because of noise and measurement error by taking into account regressor signals, typically based on their importance in terms of the size of their standard errors and by t or F tests. Hence, while machine learning treats data as “pure information”, in econometric analysis, the signal to noise ratio has an important role in identifying the relevant *control* (zero or close to zero) regressors from the variables of interest, and provide different solutions by conducting valid statistical inference with shrinkage estimators on the coefficients of interest. Although inference has not been the focus of machine learning, more recently ML ~~ec~~onometric applications for inference have received more attention, see section 19.12 on ML inference below, and Chan & Matyas (2022) for a good introduction to linear machine learning.

Subset selection procedures of either retaining or disposing variables creates a discrete process leading to high variance and inability to reduce predicted error of the full model. Shrinkage selection methods are more continuous and hence, produce lower variance than step-based methods. We will also discuss the justification for this approach in terms of its asymptotic *oracle* properties. The context for machine learning regression is almost always when $p > N$. The application of the least squares linear regression is not valid in that context if the matrix $X^{/}X$ does not have full rank, then p_1 predictors with zero coefficients values are deleted a priori when $p_1 < N < p$. By contrast, a shrinkage estimator imposes a restriction on the length of the vector $\hat{\beta}$; as this length is fixed, the idea is to obtain the best response variable approximation by setting to zero or close to coefficients as mis-identified predictors that are not useful in predicting y_i ; shrinking the β vector by setting some or even most of the coefficients equal to zero would increase the degree of freedom for the estimation of other, important non-zero predictors. This is achieved by optimizing

$$\hat{\beta} = arg_{\beta} \min g(\beta; y, X)$$

$$\text{s.t. } p\ (\beta;\ \alpha) \leq c$$

where $p\ (\beta;\ \alpha)$ is the function that *regularizes* the length of β vector, similar to the penalty function employed in time-series ~~ec~~onometrics to minimize the number of regressors, and the total length of β is bounded by the constant $c > 0$ selected a priori. The choice of c is of critical importance since if c is too small, the small coefficient values due to measurement error or being noisy will incorrectly be regarded as zero values. The optimization can also be expressed in terms of the Lagrange multiplier λ, fixed by the researcher to reflect c the length of the *a priori* selected coefficient vector, as

$$\hat{\beta} = arg_{\beta} \min g(\beta; y, X) + \lambda\, p\ (\beta;\ \alpha)$$

There is a one-to-one correspondence between λ and c, with λ being a decreasing function of c; and $\lambda \rightarrow \infty$ as $c \rightarrow 0$, the Lagrangian approaches the OLS estimator under $p < N$ assumption. λ is called the *tuning parameter* to be selected by *cross validation* (**CV**), see below.

Different regularizers with different definitions for $p\ (\beta;\ \alpha)$ produce different shrinking estimators. In this chapter we examine ML shrinkage models the *Least Absolute Shrinkage and Selection Operator* (***Lasso***), based on $p(\beta)=\sum_{i=1}^{p} |\beta_i|$, the ***Ridge*** *estimator* based on $p(\beta)=\sum_{i=1}^{p} |\beta_i|^2$, the elastic net based on $p(\beta)=\sum_{i=1}^{p} \alpha|\beta_i| + \sum_{i=1}^{p}(1-\alpha)|\beta_i|^2$, and *Smoothly Clipped Absolute Deviation* (***SCAD***) and adaptive Lasso (***adaLasso***) among other linear ML estimators.

Ridge Regression selects coefficients by applying a penalty on their *size*, shrinking them, though without excluding any of them, by minimizing a penalized residual sum of squares as

$$\hat{\beta}^{ridge} = argmin_{\beta} \{\sum_{i=1}^{N}\left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j\right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2\} \qquad (19.11)$$

The complexity parameter here that controls for the amount of shrinkage is $\lambda \geq 0$; larger values of λ results in larger amounts of shrinkage. The equivalent way to express this relationship is by the constrained imposed on the model parameters as

$$\hat{\beta}^{ridge} = argmin_{\beta} \sum_{i=1}^{N}\left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j\right)^2 \qquad (19.12)$$

$$\text{sub. to } \sum_{j=1}^{p} \beta_j^2 \leq t$$

Where the λ and t have one-to-one correspondence, and the solutions should have standardized mean and standard deviation for equivalent outcomes. The Ridge penalty term is also called the $\ell_2$penalty. The penalty term excludes intercept β_0; in effect minimization can be separated into two steps, first demean each x_i by $x_{ij} = x_{ij} - \bar{x}_j$ and first estimate β_0 by $\bar{y} = \frac{1}{n}\sum_1^N y_i$. Then the ridge regression shrinks the rest of the coefficients without the intercept. We also note that the complexity control parameter becomes a part of the criterion for deciding the coefficient significance level by the *effective degree of freedom* as a function of λ as well as the number of *p* inputs. With the inputs orthogonal to each other, the ridge estimates are the least squares estimates scaled by $\hat{\beta}^{ridge} = \hat{\beta}/(1+\lambda)$.The ridge regression can also be expressed in terms of a posterior of a regression with appropriate Bayesian priors. Suppose …, and the parameters are independent of each other, and distributed as $N(0, \tau^2)$, the negative log-posterior of, $\hat{\beta}^{ridge}$ with assumed τ^2, σ^2known, is equal to λ=$\frac{\sigma^2}{\tau^2}$ for the expression inside the curly bracket above. That is, the ridge estimate is the mode of the posterior distribution and also its mean, given the Gaussian posterior.

Lasso Regression. A more constrained shrinkage regression that imposes zero values on the small, insignificant coefficients, and applies shrinkage only to the subset of remaining parameters is the Least Absolute Shrinkage and Selection Operator (***Lasso***) regression; defined by

$$\hat{\beta}^{lasso} = argmin_{\beta} \sum_{i=1}^{N}\left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j\right)^2 \qquad (19.13)$$

$$\text{sub. to } \sum_{j=1}^{p} |\beta_j| \leq t$$

or in the equivalent Lagrangian form as

$$\hat{\beta}^{lasso} = argmin_{\beta}\ \{\tfrac{1}{2}\sum_{i=1}^{N}\left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j\right)^2 + \lambda\sum_{j=1}^{p}\left|\beta_j\right|\} \qquad (19.14)$$

Hence, the ℓ_1Lasso penalty $\sum_{j=1}^{p}|\,\beta_j|$substitutes the ℓ_2 Ridge penalty$\sum_{j=1}^{p}\beta_j^2 \leq t$. This brings out the contrast between the two error minimalization solutions. The Ridge regression employed all parameters, with none set to zero, to minimize a continuous convex set with a closed-form, linear solution, while the Lasso zero exclusion on a subset of the parameters contains a continuous set but also zero regions, and is therefore a *non-linear discrete function* with no closed form solution and requires a more complex, iterative minimization method. This is because the Lasso constraint with sufficiently small *t* sets some of the coefficients exactly equal to zero. Therefore, if *t* is set as $t_0 = \sum_1^p|\hat{\beta}_j\,|$, then $\hat{\beta}_j = \hat{\beta}_j^{ls}$ (the least squares estimates), while with *t*= $\frac{t_0}{2}$ for example, then on average, we shrink the least squares estimates by 50%. As with the Ridge regression, the Lasso parameters are standardized as $s = \frac{t}{\sum_1^p|\hat{\beta}_j\,|}$, s=1.0 leads to the least squares solution; the estimates shrink toward zero as *s* → 0.

Bridge estimator. The control for ML different estimators to reduce the size of the coefficient vector measuring the length of the vector β is expressed by assigning a *norm* to each estimator. Let us denote $L\gamma$ norm $||\beta||_\gamma$ of a vector $\beta=(\beta_1, \beta_2, \ldots, \beta_p)^/$ and define it as

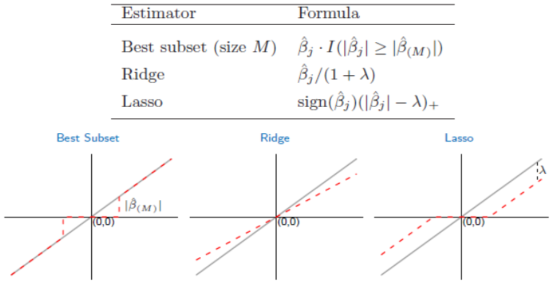
$$||\beta||_\gamma = \left(\sum_{i=1}^{p}|\beta|^\gamma\right)^{1/\gamma} \quad \gamma > 0 \qquad (19.15)$$

(19.15) encompasses a general class of linear ML models in terms of the norm of their control variable and is known as the *Bridge* estimator. When γ=1, (19.15) becomes then the L_1 norm of β leading to the Lasso estimator; when γ=2, then the L_2 norm of β leads to the Ridge estimator. A simple example for the norm of Lasso is $\beta=(\beta_1, \beta_2)$, then the L_1 norm of β is $||\beta||_2 = \sqrt{|\beta_1|^2 + |\beta_2|^2}$. An advantage of the L_1 norm (Lasso) is that it can produce parameter estimates that are exactly zero, that is, elements of $\hat{\beta}$ can be exactly zero, unlike the L_1 norm (Ridge) that does not produce $\hat{\beta}$ coefficient estimates exactly equal to zero.

The bottom part of Fig. 19.4 compares the contour plots of Ridge and LASSO for *p*=2. 19.4a shows the plot of LASSO when $|\beta_1| + |\beta_2| = 1$; if one of the coefficients is in fact zero, the contour of the least squares will intersect with one of the corners first to identify the appropriate coefficient as 0, i.e. with

$0 + |\beta_2| = 1$, the contour will be at one of the corners defined by 1 and 0 coordinates. By contrast, 19.4b plot for Ridge has no sharp corners (no $\hat{\beta}$ element exactly equal to zero). However, the Ridge has a computational advantage over the Bridge since when $\gamma \neq 2$, there are no closed form solutions for constrained optimization of $\hat{\beta}$ vector, requiring solution by a numerical method, while $\gamma = 2$, the Bridge offers a closed form solution. When $\gamma \geq 1$, the regularizer is a convex function, making available the whole set of algorithms for optimization while when $\gamma < 1$ it becomes a more complex problem and that affects the asymptotic properties of the estimators that are different when $\gamma < 1$ and $\gamma \geq 1$.

Fig. 19.4 compares the selection approaches examined above for the case of orthogonal inputs. *sign* indicates the direction of the Lasso argument (± 1), $x_+$the positive part of x, estimators show by a broken line, and the 45^0 lines those of the unrestricted estimates as reference. Each method applies a single transformation to the least squares coefficient estimate $\hat{\beta}_j$. The best subset drops all inputs smaller than the *Mth* largest, a type of "hard thresholding"; the Bridge by a proportional shrinkage and Lasso by converting each coefficient by a constant factor λ, truncated at zero called a "soft thresholding".



Estimator	Formula
Best subset (size M)	$\hat{\beta}_j \cdot I(\lvert\hat{\beta}_j\rvert \geq \lvert\hat{\beta}_{(M)}\rvert)$
Ridge	$\hat{\beta}_j/(1+\lambda)$
Lasso	$\text{sign}(\hat{\beta}_j)(\lvert\hat{\beta}_j\rvert - \lambda)_+$

Fig. 19.4 Shrinkage Methods

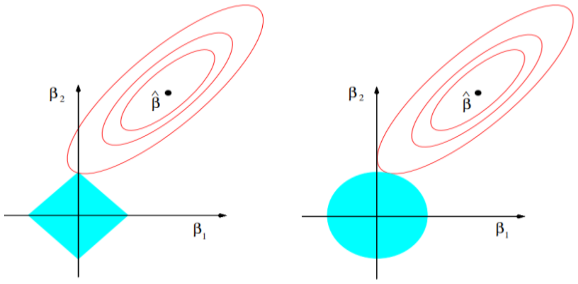


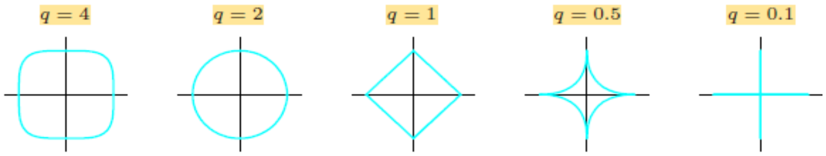
Fig. 19.5 Lasso and Ridge Shrinkage

Fig. 19.5 illustrates the difference between Lasso and Ridge shrinkage methods. The solid areas are the constraint regions where $|\beta_1| + |\beta_2| \leq t$ & $\beta_1^2 + \beta_2^2 \leq t^2$while the ellipses are the contours of the least squares error function. Both the Ridge and Lasso methods find the solution here where the elliptical contours touch the constraint region, smoothly for the Ridge circle with no corners but at a corner for the Lasso diamonds that with corners for regions where parameters are set exactly at zero. The contrast between the Ridge and Lasso methods raises the question of their generalization by adding a new parameter $q \geq 0$ that varies between no zero Ridge parameters and some exactly zero Lasso parameters viewing their estimates in terms of Bayes *priors*.

$$\tilde{\beta} = argmin_{\beta} \{\sum_{i=1}^{N}\left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j\right)^2 + \lambda \sum_{j=1}^{p}\left|\beta_j\right|^q\} \tag{19.16}$$

Fig. 19.6 shows the contours of constant values of $\sum_{j=1}^{p} |\beta_j|^q$where $|\beta_j|^q$ are the log-prior density of β_j and also equi-contours of the prior distribution of the parameters, for the case of two inputs at different values of q. At q=0 corresponds to the variable subset selection using the number of non-zero parameters, q=1 and q=2 to that of the Lasso and the Ridge regressions respectively. The values of $q \in (1, 2)$ therefore offer a compromise between the Ridge and Lasso regressions.

Fig. 19.6 Shrinkage comparison by Bayes Priors



However, with q > 1, $|\beta_j|^q$ remains differentiable at 0, so does not set some coefficients exactly equal to zero as in Lasso.

Elastic net. We can specify the regularizer in more general terms than one based only on the norm. Elastic net proposed by Zou and Hastie (2005) is a linear combination between ℓ_1 and ℓ_2 norms with Lasso and Ridge as special cases.

$$p(\beta;\ \alpha) = \alpha_1||\beta||_1 + \alpha_2||\beta||_2^2 \qquad \alpha \in [0,1]$$

When (α_1, α_2)=(1, 0), reduces to Lasso and when (α_1, α_2)=(0, 1) to Ridge. The exact values of (α_1, α_2) are specified by the researcher together with γ via cross validation; hence, unlike Bridge, elastic net has more than one tu~~r~~ning parameters. A common choice is $(\alpha_2 = 1 - \alpha_1)$ with $\alpha_1 \in$

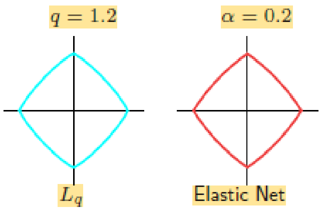
$[0,1]$, a case known as an *affine combination* of the ℓ_1 and ℓ_2 norm. The above suggests an elastic net penalty as

$$\lambda = \sum_{j=1}^{p}(\alpha\beta_j^2 + (1-\alpha)|\beta_j|) \qquad (19.17)$$

The elastic net overcomes some limitations of Lasso and Ridge by striking a balance between the two. Fig. 19.7 shows the elastic net and ℓ_q penalty with contours of $\sum_{j=1}^{p}|\beta_j|^q$set at q=1.2 and elastic net penalty, sum expression in front of λ in (19.17) at α=0.2. The elastic net selects inputs like Lasso and shrinks correlated inputs together like Ridge; it shows a great advantage over L_q penalties; here elastic net has non-differentiable, sharp corners while the q=1.2 penalty does not.

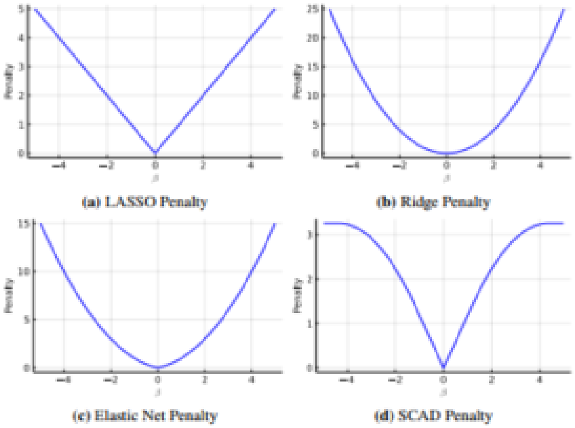
Closely related to the Lasso is the relatively recent Least Angle Regression (LAR); rather than fitting a variable toward zero, it moves its coefficients toward its least squares estimates, hence causing its correlation with the error term to decrease in absolute value. The process is stopped as soon as another variable reaches the same correlation with the error term, then that variable is added to the active set and their coefficients tied together decrease toward the least squares value. The process continues until all the variables are included with a full least squared fit. With centered data to remove the intercept, if $p > N$-1, the LAR reaches zero correlation residual after N -1 steps. The LAR makes the smallest and equal angle with each of the predictors.

Fig. 19.7- ℓ_qPenalty



However, we should note that in the ML literature, the difference between unbiasedness and consistency is not always clear, while in econometrics that distinction is the basis of how asymptotic distributional properties are formulated. We should thus interpret the unbiasedness feature of liner shrinkage models as a consistency property.

SCAD & AdaLasso. The increased weighting by the size of non-zero coefficients of some variables by imposing a restriction on the tuning parameter γ, as a means of reducing the vector of coefficients, implies that Lasso estimates are biased due to the potential misspecification that would undermine the potential statistical significance of some predictors. An alternative shrinkage estimator that addresses the issue of estimation LASSO bias~~ed~~ is *Smoothly Clipped Absolute Deviation* (**SCAD**). SCAD has three features: unbiasedness, addressed to a Lasso biased estimate, sparsity for a threshold role that sets unnecessary variables to 0, and continuity in data. The first makes the SCARD regularizer a function of the tuning λ itself. The second, while sparsity of SCAD divides the parameter space into zero and non-zero sectors, unlike Lasso, the penalty term for SCARD does not increase when the magnitude of the coefficient is large, that is, when $|\beta|$ exceeds a certain magnitude. Fig. 19.2 shows that all four estimators have increased penalty as the coefficient size increases; and the rate of change equals the tuning parameter, λ; both Ridge and elastic net treat large coefficients similarly but for the latter the penalty rate of change for coefficients close to zero is larger than for the former, plots (*c*) *v.* (*b*); hence, Ridge pushes small coefficients to zero. By contrast, the SCAD behaves like Lasso when coefficients are small but the SCAD penalty remains constant when the coefficients are large as in plot (*d*), helping to improve estimation bias of ML estimators like Lasso.



(a) LASSO Penalty

(b) Ridge Penalty

(c) Elastic Net Penalty

(d) SCAD Penalty

Fig. 19.8-*Linear Model Penalties.*

Adaptive Lasso (**adaLasso**) is a Lasso modification that has fewer variables and hence better model-selection properties. AdaLasso takes the consistent estimator information, and applies notably larger penalties to the coefficients close to zero by assigning them weights as

$$k_j = 1/|\hat{\beta}_j|^{\delta},$$

where δ > 0; “1./” indicates element-wise division, |*β*| denotes an element-by-element absolute value operator, and $\hat{\beta}$ is some consistent estimator of *β*. We note that employing a consistent estimator for the construction of the weight is a drawback of adaLasso since a consistent estimator may not be available when *p* > *N*. Instead of setting *K*-fold CV for *J* variables k_j= 1, it starts CV Lasso with k_j= 1, but then adapts the second Lasso by excluding $\hat{\beta}_j$=0 and for the remaining sets has $k_j = 1/|\hat{\beta}_j|^{\delta}$, with δ=1 so as to select variables with larger coefficients to receive a smaller penalty. AdaLasso, Sou (2006), is defined by

$$\hat{\beta}_{ada}= arg_{\beta} \min g(\beta; y, X) + \lambda \sum_{j=1}^{p} w_j |\beta_j| \qquad (19.18)$$

Where w_j > 0 are weights for *j*=1, 2, . . . , *p.* predictors predetermined by the researcher. The adaptive nature of this estimator stems from its vector of weights $w = (w_1, w_2, \dots, w_p)'$ being based on any consistent estimator of *β*. The default is to have one adaptive step, CV selection of tuning parameter and model selection based on BIC or plug-in iterative used for estimation rather than prediction, see below. Unlike Lasso, the adaLasso has important Oracle properties discussed below that the standard LASSO does not possess.

Group Lasso. This mrthod applies where the predictors belong to pre-definte groups, for example dummy variables presenting the levels of categorical predictors. In such instances we wold like to shrink and select the members of a group together, and the group Lasso~~n~~ can achieve that. Divide the *p* predictors in *L* goups with an X_{ℓ} matrix and a coefficients vector of β_{ℓ}, then the group Lasso minimizes the convex set

$$min_{\beta \epsilon \mathbb{R}^p}(||y - \beta_0 1 - \sum_{\ell=1}^{L} X_{\ell}\beta_{\ell}||_2^2 + \lambda \sum_{\ell=1}^{L} \sqrt{p_{\ell}}||\beta_{\ell}||_2 \qquad (19.19)$$

Where $\surd p_j$ acounts for the changing group sizes, and $||\ .||_2$ is the Eulidian (unsquared) norm; for some values of λ, an entire group of predictors may be dropped from analysis. *Group Lasso* is employed when interpreting the coefficients requires the sub-set of variables to be non-zero, for example a categorical variable with *M* options where each of *M* dummies represent a single category and interpretation becomes problematic if some of the coefficients are zero. In this case it makes sense to group the coefficients together to ensure sparsity at the categorical rather than individual dummy level. The construction of group Lasso imposes the ℓ_2 norm (like Ridge) on the grouped coefficients (none exactly zero) while imposing ℓ_1 as Lasso to each of the continuous variable coefficients and the *collective coefficients* of the categorical variables.

19.5 ***Oracle properties of Shrinking Estimators***.

The asymptotic distribution of linear shrinking estimators is usually examined in terms of their *Oracle properties*. Oracle estimators share the same properties as estimators with the correct set of covariates; given that, Oracle estimators have the ability to 'foresee' the correct set of covariates. To define Oracle properties, rearrange the true vector β_0 so that all non-zero values are grouped into one sub-vector, all zero values into another sub-vector, both containing corresponding coefficient indices. Let

A={j:$\beta_{0j} \neq 0$} & $\hat{A} = \{j: \hat{\beta}_j \neq 0\}$ and $\beta_0 = (\beta'_{0A}, \beta'_{0A^c})$ & $\hat{\beta} = (\beta'_A, \beta'_{A^c})$. Then estimator $\hat{\beta}$ has the oracle Properties if

Selection Consistency: $\lim\limits_{N \to \infty} \Pr(\hat{A} = A) = 1$ and

Asymptotic normality: $\sqrt{N}(\hat{\beta}_A - \beta_A) \overset{d}{\to} N(0, \Sigma)$

where the variance-covariance matrix Σ for the *oracle estimator* defined as

$$\hat{\beta}_{oracle} = arg_{\beta:\beta_{A^c=0}} \min g(\beta) \qquad (19.20)$$

(19.20) has the same asymptotic distribution as the estimator with only the non-zero variables. We note that selection consistency is a weaker condition than the traditional statistical consistency employed in econometrics since it only requires the ability to discriminate between zero and non-zero coefficients. We can view the ML estimators discussed above in terms of the Oracle Properties. It is clear that neither Lasso nor Ridge has oracle properties; Lasso estimation is inconsistent due to weighting non-zero coefficients toward zero, while Ridge does not possess selection consistency. By contrast, adaLASSO, SCAD and Group Lasso all have oracle properties; this is a reason for the popularity of adaLasso.

19.6 Principal Components Analysis (**PCA**).

An alternative approach to solving the *p* >*N* dimensionality problem is to retain all regressors but group them into a limited number of sets with largest intra-group variation and assess their impact on the response variable. Chief among them is the Principal Components Regression (PCR) that has a derived predictor column $z_m = x_{vm}$ that is orthogonal. Then, regression of y on $z_1, z_2, \ldots, z_M$ for $M \leq p$ becomes just a sum of the univariate regression

$$\hat{y}^{pcr}_{(M)} = \bar{y}1 + \sum_{m=1}^{M} \hat{\theta}_m z_m \qquad (19.21)$$

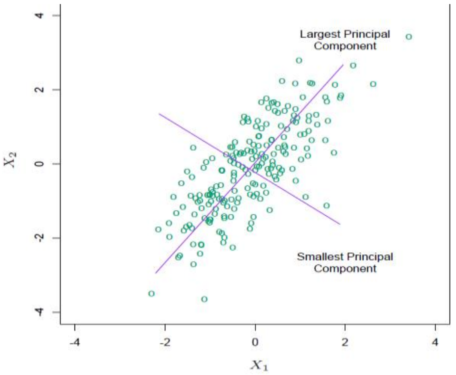
where $\hat{\theta}_m = \langle z_m, y\rangle / \langle z_m, z_m\rangle$. The solution expressed in terms of x_j coefficients is

$$\hat{\beta}^{pcr}(M) = \sum_{m=1}^{M} \hat{\theta}_m v_m \qquad (19.22)$$

As with the Ridge, the PCR coefficients should first be standardized to make them scale independent. *M=p* leads to the least squares estimates, while *M* < *p* leads to a reduced form similar to the Ridge; while Ridge shrinks all *p* toward zero, the PCR drops *p* – *M* smallest components.

A linear combination alternative to PCR is the *Partial Least Squares* (**PLS**) that uses *both* *x* and *y* variables, with the first combination as $\varphi_{1j} = x_j, y$ for each *j* and drives the first PLS as, $z_1 = \sum_j \hat{\varphi}_{1j} X_j$. Hence, each z_m is weighted by the extent of their univariate effect on y. Then, y is regressed on z_1 to orthogonize all x_j with respect to z_1 to obtain $\hat{\theta}_1$ and coninue until all $M \leq p$ directions are obtained. PLR seeks directions with high variance and high correlation with y. One way to view the Ridge shrinkage input vector is in terms of of the number of components used in PC. Fig. 19.9 displays PC for some data points of the input vectors X_1 & X_2. The largest PC is the dirction that maximizes the variance of the projected data; the smallest PC minimizes that variance. Ridge regression projects *y* onto these components, and then shrinks the coefficients of the low-varaiance components more than the high-varaince components, see Tibshrini et. al. (2016).

Fig. 19.9 PC Vectors



High Dimensional Problem: $p \geq M$. The traditional application of PCA regression or classification is in ~~low~~ *low* dimension $N > p$ where diminsion refers to the size of the p vector. We say a regression is in high dismension when $p \geq N$. Here are two examples of analyses in high dimenstional:

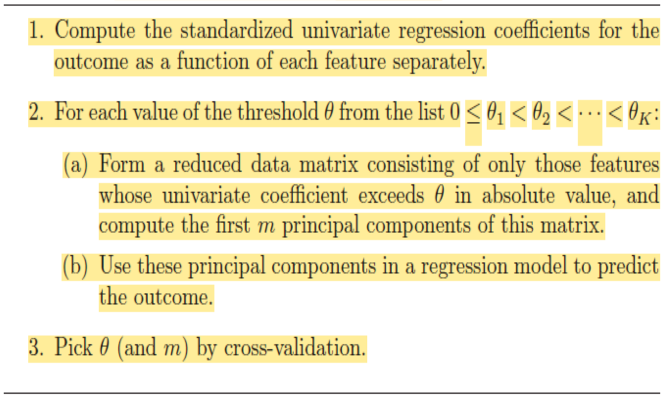
i. Instead of predicting blood pressure based solely on age, sex, etc. of the patient, we can include in the predictive model half a million rel~~l~~atively common individual DNA mutations with $N \approx 200$ but $p \approx 500{,}000$ to obtain more accurate predictions.

ii. Understading peoples' hopping patterns can employ the users' search terms in a "bag-of-words" model (see chapter 20). For each user, each of the p search terms is recorded as (0) for present and (1) for absent resulting in a large binary feature vector then with $N \approx 1000$, for example, p would be much larger.

PCA regression and classifications are called *supervised PCA* because the analysis is guided in terms of the relationship of featues to the outcome, in contrast to *unspervised PCA* to an analysis without a dependent application, used mainly to divide a large number of observations into ~~a~~ smaller homogeous clusters discussed below. If $N=p$, then *supervised PCA* regression will have a perfect fit. We would then have very poor prediction by the test set, as the regression line is too flexible and overfits the data. Moreover, the traditional model selections *AIC* and *BIC* do not work since estimated variance would be zero and $\hat{R}^2 = 1$. To aviod overfitting, we must shrink the p dimension by Lasso and Ridge methods or cluster the features into a very small number of PC predictors.

PCA should be modified for the context of high demensional regression, involving univariant regression of the outcome on each of the predictors and a method of aggregating them into a linear combination of a limited number of components with both (*a*) high variance and (*b*)significant correlation with the outcome, presented in the following algorithm as a procedure to find the combinations that meet the optimal (*a*) and (*b*):

19.1. Algorithm for Supervised Principal Components



```
1. Compute the standardized univariate regression coefficients for the
   outcome as a function of each feature separately.
2. For each value of the threshold $\theta$ from the list $0 \leq \theta_1 < \theta_2 < \cdots < \theta_K$:
    (a) Form a reduced data matrix consisting of only those features
        whose univariate coefficient exceeds $\theta$ in absolute value, and
        compute the first $m$ principal components of this matrix.
    (b) Use these principal components in a regression model to predict
        the outcome.
3. Pick $\theta$ (and $m$) by cross-validation.
```

The choice for step (1) and (2b) depends on the kind of regression outcome , commonly the univariant OLS coefficients for (1), and linear least squaers model for (2b). The relationship between features and the outcome in supervised PCA can also be expressed in terms of an underlying latent variable U :

$$y = \beta_0 + \beta_1 U + \varepsilon$$

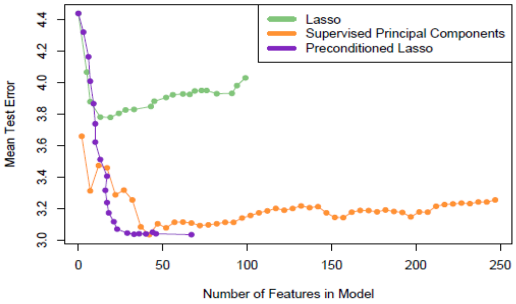
Consider also a j set of measurement observations for p for which

$$X_j = \alpha_{0j} + \alpha_{1j} + \varepsilon_j \, , j\epsilon p$$

With the mean zero error terms independent of all other random variables in the corresponding regression. The goal here is to identify $\mathcal{P}$, and estimate U in order to fit the prediction model. Then we can express the algorithm as; in step (1) we estimate the set $\mathcal{P}$; given $\widehat{\mathrm{P}}$, we use the argest PC in step (2a) to estimate U, and finally, fit the regression model in step (2b) to estimate the model's coefficients. The leading PC may be be consistent if it is affected by the presence of a large number of "noise" features, therefore we must also consider a procedure to reduce a set of approximate features for the model. PCA does not always produce a sparse model even if step (1) of the algorithm produces only a small number of predictors, as some of the omitted variables may have ~~a~~ large interaction effects with the selected components. On the other hand, highly correlated predictors are selected together, and therefore, contain a large number of redundent variables. One solution is to apply Lasso by relying on its sparsity assumption to select predictors. However, Lasso can perform poorly by the predicted test errors if the training set overfits due to the effects of a large number of "noi~~n~~se" inputs. *Preconditioning Lasso* by the PC selected variables provides a procedure for dealing with that problem. In this method, we first compute the PC supervised components to obtain $\hat{y}_i$ for each observation in the training set, selecting the threshold by CV; then apply the Lasso with $\hat{y}_i$ instead of y_i. We use all the features in the Lasso fit, not just the components used in the supervised PC threshold step. The effect of Precondition Lasso application is first to remove the noise varaibles affecting the outcome, thus preventing the adverse effects of the noise on it.

Example : Plot 19.1 com~~a~~pares the test errors set for the Lasso, supervised PC, and PC pre-conditioned Lasso for a gene mesurements data set. The supervised component path is trauncated at 250 gene features, while the Lasso self-truncates at 100. In this case, while the the Lasso starts to ovefit, the Preconditioned Lasso test error is as low as that for the PC employing fewer features;, hence, performs better than both.

Plot 19.1-Lasso and PC Test errors



19.7 Machine Learning Inference.

Machine learning models for prediction cannot be employed for inference estimation due to differences in their asymptotic distribution of model section methods. An asymptotic model selection method is consistent if it selects the correct model from a list of potential candidates; it is conservative if it always selects a model that nests the correct one. *BIC* is a consistent model-selection procedure while *AIC* is conservative based on their minimum values. A consistent method has an o*racle* property if it selects a consistent model with an estimator asymptotic equivalent to the one that would have chosen the unknown correct model. As an example, consider

$$y = \alpha x_{1i} + \beta x_{2i} + u_i \qquad (19.23)$$

the correct model is either β=0 or β≠0, a consistent method selects the correct model β=0 by first using an estimator $\hat{\alpha}$ for the true model estimator that decides if β=0, and then proceeds to estimate the selected model without having to go through the initial selection. Lasso is an example of a consistent model selection method that does not have the Oracle property because of its estimation bias. Other regularized methods such as adaptive Lasso have that property but that is not helpful for working with a finite size sample since Oracle is an asymptotic property. With more observations we can detect more variables with values close to zero, but then the estimator $\hat{\alpha}$ will have complex form for parameter estimation, typically with very large MSE, Lasso asymptotic convergence is not uniform with respect to parameters. Therefore, we cannot perform inference on Lasso and post-Lasso OLS models; we consider more demanding models required for machine learning inference in this section.

19.7.1 ***Partialing-out estimator***. Machine learning econometrics estimate non~~e~~-parametrically one or more variables of interest but controlling for other variables as with the standard microecomonetric models with $p<N$ requires dealing effectively with the dangers of data mining and overfitting. The machine learning inference for parameter estimation takes a semi-parametric method that estimates the principal parameters of interest that is also a function of other “nuisance” variables. If the estimation satisfies an orthogonality moment condition, then the approach allows for inference on the parameters of interest. The leading estimator employed is the Partial Linear Model (**PLM**), discussed in chapter 11, using Lasso with the sparsity assumption that only a few potential controls are relevant:

$$y = \mathrm{d}'\alpha + g(\mathbf{X_c}) + u \qquad (19.24)$$

Where $g(.)$ is a flexible functional form for the selected controls and x_c; α has a causal interpretation based on the selection-on-observable-only $E(u|\mathrm{d}, \mathrm{X}_c) = 0$. The PLM application produces √N consistent and asymptotically normal estimator of the α partial effect. However, the PLM consistent estimation leaves g(.) unspecified, and requires the model semi-parametrically to be estimated with only a few x_c controls to overcome the curse of dimensionality. Lasso modification for inference instead employs more complexity for g(.) specified as $g(X_{c)} \cong \mathrm{X}'\gamma + r$ that allows for flexibility of x_c with polynomials and interactions and r is an approximation error, we start with

$$y = \mathrm{d}'\alpha + \mathrm{X}'\gamma + r + u \tag{19.25}$$

Given a well-specified good set of controls, $\hat{\alpha}$ can be interpreted causally with the main assumption of sparsity, namely that only a limited number of x variables are relevant.

However, the literature has employed an alternative partialing-out estimator that is equivalent to the PLM.

In this approach, ***d*** scalar is the regressor of interest, typically a policy variable. First apply a Lasso of d on x to obtain its OLS residual $u_d$on the selected x, then apply a Lasso of y on x and the OLS residual u_y on the selected variables. At the final stage, we obtain $\hat{\alpha}$ from an OLS regression of u_y on $u_d$in a procedure equivalent to the PLM estimation. In general, with *K* main regressors, the partialing-out estimator performs *K* separate Lassos for each u_d and *K* least squared regressors for the second Lassos to finally obtain $\hat{\alpha}_1$, $\hat{\alpha}_2$, . . . , $\hat{\alpha}_k$. This procedure is equivalent to that of *PLM* but while the latter uses residuals from kernel regression, and employs the partialing-out sparsity assumption, that the number of p nonzero coefficients *s* in the true model is small relative to the sample size *N*, this is expressed as

$$s/(\sqrt{N}/\ln p)$$

that grows at a rate slower than √*N*; with an approximation error that satisfies

$$\sqrt{(\frac{1}{N})\sum_{i=1}^{N} r_i^2 \leq c\sqrt{(\frac{s}{N})}}$$

for *c*>0. It is common to use the plug-in formula for the tuning parameter.

$$\lambda = c\,\sqrt{N}\Phi(1 - \{\frac{\gamma}{2p}\}$$

with regression individual loadings $k_j = \sqrt{(1/N)\sum_{i=1}^{N}(x_{ij}\hat{\varepsilon}_i)^2}$and normalized x_j with mean zero and standard deviation of one. The formula is applicable under heteroskedasticity and homoskedasticity; *c*=1.1 and $\gamma = 0.1/\ln\{\max(p, N)\}$, this estimator provides good values for *c* and *γ*

19.7.2 ***Partially Penalized Estimator***.

There are some circumstances where the parameter of interest~~s~~ may not be part of the shrinkage model so there is no need to face the challenging application of the regularizer to the full range of the parameter vector. Let us define the regression equation in terms of two sub-vectors of zero and non-zero variables as

$$y = X_1\beta_1 + X_2\beta_2 + \text{u} \qquad (19.26)$$

Where $X_1$is $N \times p_1$ and X_2 is $N \times p_2$ matrices with $p= p_1 + p_2$, and β_1& β_2 are $p_1 \times 1$ and $p_2\times 1$ vectors, and assume only β_2 have zero elements, i.e. only $\beta_2$is sparse. Then then *Partially Linear (Regularized) estimator* is

$$(\hat{\beta}_1', \hat{\beta}_2')= arg_{\beta_1,\beta_2} \min(y - X_1\beta_1 - X_2\beta_2)'(y - X_1\beta_1 - X_2\beta_2)+ \lambda\, p\, (\beta_2) \quad (19.27)$$

Note that the penalty is imposed on $\beta_2$only; not on β_1. The equation is whether the asymptotic properties

of $\hat{\beta}_1$allow valid inference; it is possible to show that the Bridge estimator can provide valid parameter inferences that are not part of the shrinkage process. The idea can also be generalized. Consider

$$y = X_1\beta_1 + \text{u}$$

$X_1$is $N \times p_1$ containing some endogenous variables and we have a large number of p_2 potential instruments for them. With the 2SLS estimator, we construct the first stage instrumental variables by estimating

$X_1 = X_2\Pi + \text{v}$ given the set of $Z = X_2\widehat{\Pi}$. The estimation of Π is separate from β_1; moreover, we may be interested in Z as the main target in its own right for providing the best approximation for X_1, given X_2. It can be shown that PLM provides the best instruments for approximating X_1, given X_2, and it reduces the number of instruments using the sparsity of the shrinkage estimators to resolve the problem of too many instruments. This approach is feasible because it is possible to obtain optimal instruments Z based on post-Selection OLS (after applying a shrinkage procedure); then the IV type estimators follow standard asymptotic results. Recall that interest in this case is on the instrumental vector as the main target of estimation for the best approximation for X_1rather than the quality of estimator for Π.

Application 1: A recent application of the PPE is to the distributed lag model. Consider the challenge of estimating such a model as the number of observations increase; the potential numbers of L increases, creating identification problems. Suppose we are only interested in inference on non-lag parameters β and specify the model as

$$y_i = x_i'\beta + \sum_{j=1}^{L} x_{i-j}'\,\alpha_j + u_i \tag{19.28}$$

with $L < N$. Then one can apply PPE to estimate the model by

$$\left(\hat{\beta}\hat{\alpha}\right) = arg_{\beta,\alpha}min\sum_{i=1}^{N}(y_i - x_i'\beta - \sum_{j=1}^{L} x_{i-j}'\,\alpha_j)^2 + \lambda p(\alpha) \tag{19.29}$$

where the $p(\alpha)$ regularizer is applied to $\alpha = (\alpha_1', \alpha_2', \ \ldots, \ \alpha_L')$ only; since β is not a part of the shrinkage, under the above Bridge regularizer proposition, $\widehat{\beta}$ has an asymptomatically normal distribution allowing fruitful application of PPE in that context of inference on β. The basic idea on PPE applicable to other contexts is that we have to control for a large number of variables, potentially with *P*>*N*, and the parameters of interest do not constitute any components of the shrinkage, then valid inference on non-shrinkage coefficients by PPE is possible.

Application 2: A similar argument supports PPE application to penal data. Consider a fixed effect penal data model with dummy variable fixed effect control as

$$y_{it} = x_{it}'\beta + \alpha_i + u_{it} \tag{19.30}$$

When *N* is large, the number of dummy coefficients to estimate is also growing to an unacceptable number. However, if β is the vector of the coefficient interest, and α_i fixed effect vector is constant for *i*=1, 2, . . . , *N*, then PPE can provide a valid inference on β using

$$\hat{\beta}_{FE} = arg_{\beta}min = \sum_{i=1}^{N}\sum_{t=1}^{T}(\dot{y}_{it} - \dot{x}_{it}'\beta)^2 \tag{19.31}$$

Where $\dot{y}_{it}$and $\dot{x}_{it}$ are mean-deducted. The dummy variable approach suffers from the curse of dimensionality, so the above approach can provide a valid inference in that context.

19.7.3 ***Orthogonalization***. The PLM partialing-out model of α is a two -step estimator for which the second stage asymptotic distribution of α does not change by the first stage estimation because the former satisfies the moment orthogonality condition. Consider α as parameter of interest and η as the nuisance parameters. The Two-stage Partialing-out estimator first estimates $\hat{\eta}$ and then $\hat{\alpha}$ by solving for all variables w_i

$$\sum_{i=1}^{n} \psi\,(w_i, \alpha, \hat{\eta}) = 0 \qquad (19.32)$$

The asymptotic distribution of α is independent of the first stage estimation if $\psi(.)$ satisfies

$$E\{\partial\psi(w, \alpha, \eta)/\partial\eta\} = 0 \qquad (19.33)$$

(See Cameron 2005, 210). Then, if a change in η does not affect ~~in~~ expectation $\psi(.)$ which implies estimates $\hat{\eta}$ leaves the distribution of $\hat{\alpha}$ unchanged. To show this, consider *y*= α*d* + *g*(*X*) + *u* and define $\eta_1 = E\,(\frac{d}{X})$ and $\eta_2 = E\,(\frac{y}{X})$. The expectations of $\hat{\eta}_1$and $\hat{\eta}_2$ are from the OLS regression of ***d*** and ***y*** on the Lasso selected components of *x*. The Partialing-out $\hat{\alpha}$ is fron the OLS regression of (*y* - $\hat{\eta}_2$) on (*d* - $\hat{\eta}_1$), corresponding to the population moment condition

$$E\{\psi(w, \alpha, \eta_1, \eta_2)\} = 0 \qquad (19.34)$$

where $\psi(w, \alpha, \eta_1, \eta_2) = (d - \eta_1)\{(y - \eta_2) + \alpha(d - \eta_1)\}$; the term in front of the curly brackets corresponds to x_i, while that inside the curly brackets is the error term u_i; Therefore, the OLS estimator of *y* on *x* satisfies

$$\sum_i x_i u_i = \sum_i x_i(u_i - \beta x_i) \ = 0 \qquad (19.35)$$

with corresponding moment condition $E\{x(y - \beta x)\} = 0$.

19.7.4 ***cross-fit partialling-out.*** Another model for inference is an adaptive procedure for bias reduction that employs different samples for predictions of y from components of *d*, and the subsequent sample for α estimation. Combining cross-fit with orthogonalized moment is known as the *Debiased* or *Double Machine Learning* estimator. We divide the sample into a larger subsample of nuisance variables using Lasso components of *d* on *x* and *y* on *x*, and a small one for the variable of interest employing OLS regression of *d* and *y* on $\hat{d} = X'\hat{\pi}_d$ and $\hat{y} = X'\hat{\pi}_y$. These then use the residuals $\tilde{u}_d = d - X'\hat{\pi}_d$ and

$\tilde{u}_y = y - X'\hat{\pi}_y$ finally, the OLS estimate of $\tilde{\alpha}$ derives from $\tilde{u}_y$ and $\tilde{u}_d$. The double learning results in less restricted sparsity assumption that the nonzero coefficients grow at a rate no more than *N* rather than √*N*, that is, *s*/(*N*/ln *p*) small for *p* potential controls and *s* subset of them in the correct model. This estimator is asymptotically equivalent to the Partialing-out estimator. To make up for efficiency loss from using only a part of the original sample, we estimate a *K*-fold cross fitting double learning model hence obtaining from $\tilde{\alpha} = 1/K \sum_{k=1}^{K} \tilde{\alpha}_k$.

19.7.4-*Parialing-out IV estimator*.

The Partialing-out approach to reducing the number of input variables can be exploited to resolve the problem of over-identification for the case of many more instruments than endogenous regressors by IV estimate linear estimation of a subset of selected controls and instruments.

The Partialing-out *IV* estimates a model with ***d*** endogenous variables, **W** exogenous variables to be retained as always included and **X** controls variables; there are also Z instruments with *dim* [**Z**] ≥ [**d**]

$$y = \mathbf{d}'\alpha + \mathbf{W}'\delta + \mathbf{X}'\gamma + v \qquad (19.36)$$

The IV Partialing-out algorithm, simplified with δ=0 is

i. Obtain the Partialing-out residual $\hat{u}_{yi}$ with Lasso selected variables

ii. Calculate the scalar instrument $\breve{u}_{di}$from a Lasso regression of **d** on **X** and **Z** with selected variables $\tilde{X}_d$ and $\tilde{Z}_d$ used to predict $\hat{\mathbf{d}}$; finally calculate $\breve{u}_{di}$ and the $\breve{\beta}$ from the OLS regression of $\hat{\mathrm{d}}$ on $\breve{X}_{\hat{\mathrm{d}}}$ (the selected Lasso variables of $\hat{\mathrm{d}}$.

iii. Calculate the Partial-out endogenous regressor $\hat{u}_{di} = d_i - \breve{X}'_{\hat{\mathrm{d}}_i}$.

iv. Compute $\hat{\alpha}$ by IV regression of $\hat{u}_{yi}$ on $\hat{u}_{di}$ with $\breve{u}_{di}$ as the instrument.

19.8-Unsupervised Machine Learing.

Learning is supervised by an associated respose variable y. By contrast, unspervised learning analysis focuses on the features themselves without a response variable, and can provide important insights in some contexts. For example, in a sample of cancer patients, we may wish to look for subgroups of patients who share similar gene compositions for a better understanding of the disease. We cannot rely on the regression MSE or classification error rate to select the best sample sungroup divisions but we can still apply measures that can produce groups that achieve relative group homogeneity. Two types of unsupervised learning are common: *PCA* and C*łustering*. We first examine unsupervised PCA.

19.8.1 Unsupervised PC. Since with unsupervied we cannot check the outcone of the anaysis by the model prediction, the PCA identifies a large set of corelated varibles to summarise the data into a relatively smaller represenative groups the collectively explain most of the original data variations. To do so, PCA employs divisions that lead to the maximum within-group correlation.

Consider the original data set X of n x p , all centered to have mean zero; suppose we first want to find the group of varaibles that have the largest corr varaince subject. We obtain the linear combination of p features

$$Z_1 = \emptyset_{11}X_1 + \emptyset_{21}X_2 + \ldots + \emptyset_{p1}X_p \qquad (19.37)$$

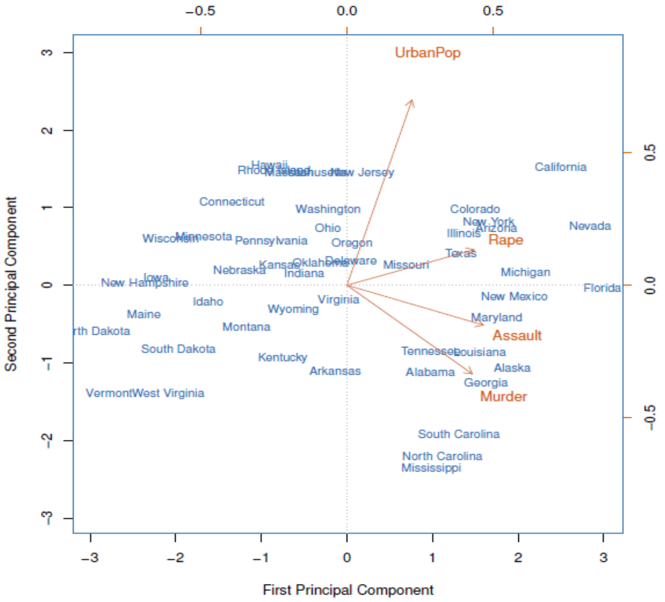
Then we can find the first component with the largest within group varaince, subjec to$\sum_{j=1}^{p} \emptyset_{j1}^2 = 1$

by solving for the maximixation of

$$minimize_{c_1,\ldots,c_k}\{\frac{1}{n}\sum_{i=1}^{n}\left(\sum_{j=1}^{p}\emptyset_{j1}x_{ij}\right)^2\} sub.to\ \sum_{j=1}^{p}\emptyset_{j1}^2 = 1 \qquad (19.38)$$

This is the sample varaince of the n values of z_{i1}; we call z_{11}, z_{21}. . . , z_{n1} the *scores* of the first principal component, and the lòoading vector $\emptyset_1$ defines the direction alone the feature axis for which the data variation is the largest. The second principla component Z_2has the linear combination of features with the largest within group variance among the remaining data that are uncorrelated with the first component Z_1, ect. This is equivalent to constrianing the direction of $\emptyset_2$ loading to be orthogonal to that of $\emptyset_2$ anf found by a second similar within group variance maximization. Then we represent group homogeneity by the vector sore graphicall by Z_1 aginst Z_2; Z_2 against Z_3 , ect. or equivelently by projecting the oroginal data onto the subspace of $\emptyset_1$, $\emptyset_2$, $\emptyset_3$, . . ., these are the ordered sequence of eigenvectors of the matrix X^TX, the variance of the components are the eigenvalues.

Example. USArrests data set have score length of n=50 and p=4. Plot 19. 2 shows the first two principal componetnts together with their scores and loadings, the fitrst loading put equal weight on Assault, Murder and Rape and much less on Urbanpop while the second loading does the opposite, hence the second PC roughly presents the level of urbanization.



Plot 19.2 PC with a two components

Another interpretaion of *PC* is that they represent~~s~~ the dimensions of the data as close as possible to all the data points in terms of average squared Euclidian distances, the first *M* score_vectors and loadings lead to the best *M* dimensional approximation to x_{ij} data points $x_{ij} \approx \sum_{m=1}^{M} z_{im}\phi_{jm}$.

$x_{ij} \approx \sum_{m=1}^{M}(z_{im}\emptyset_{jm})$

The first principal compenets are then found by minimization of the residual sum of squares out of all approximations of x_{ij} to obtain, for given scores $\emptyset_{jm}$, the solution as

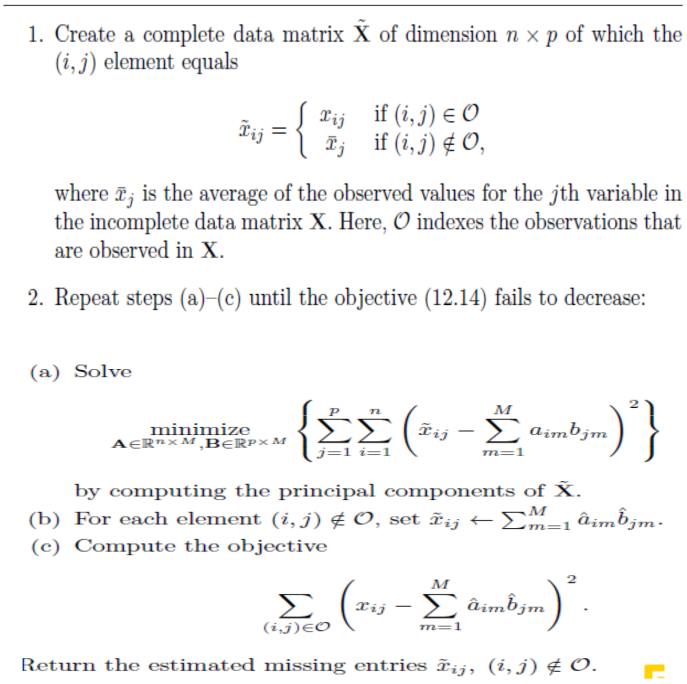
$$\sum_{j=1}^{p}\sum_{i=1}^{n}\left(x_{ij} - \sum_{m=1}^{M} z_{im}\phi_{jm}\right)^2$$

$$\sum_{j=1}^{p}\sum_{i=1}^{n}(x_{ij} - \sum_{m=1}^{M}(z_{im}\emptyset_{jm})^2$$

The question is how much of the variance in the data is left out by the first few PC? We wish to know the proportion of variance explained (***PVE***)by each PC realtive to the total variance in the data set. It turns out that we can decompose the total variance as that of the first M components, and the remaining residual and unsupervised PC as an equivalent minimization problem and PVE as the R^2 of the approximation.

An important application of PCA is for *imputation* for missing data by a procedure knaown as *matrix completion*. Imputation requires first to check if the missing data is randomly distributed. The procedure starts with the full data matrix, then obtains the mean values for nonmissing data, imputes these to the missing data and deducts the mean from nonmissing data, as explained in the following 19.2 algorithm:

19.2 Matrix Completion Algorithm



1. Create a complete data matrix $\tilde{\mathbf{X}}$ of dimension $n \times p$ of which the (i, j) element equals

$$\tilde{x}_{ij} = \begin{cases} x_{ij} & \text{if } (i,j) \in \mathcal{O} \\ \bar{x}_j & \text{if } (i,j) \notin \mathcal{O}, \end{cases}$$

where $\bar{x}_j$ is the average of the observed values for the jth variable in the incomplete data matrix $\mathbf{X}$. Here, $\mathcal{O}$ indexes the observations that are observed in $\mathbf{X}$.

2. Repeat steps (a)–(c) until the objective (12.14) fails to decrease:

(a) Solve

$$\underset{\mathbf{A} \in \mathbb{R}^{n \times M}, \mathbf{B} \in \mathbb{R}^{p \times M}}{\text{minimize}} \left\{ \sum_{j=1}^{p} \sum_{i=1}^{n} \left(\tilde{x}_{ij} - \sum_{m=1}^{M} a_{im} b_{jm} \right)^2 \right\}$$

by computing the principal components of $\tilde{\mathbf{X}}$.

(b) For each element $(i, j) \notin \mathcal{O}$, set $\tilde{x}_{ij} \leftarrow \sum_{m=1}^{M} \hat{a}_{im} \hat{b}_{jm}$.

(c) Compute the objective

$$\sum_{(i,j) \in \mathcal{O}} \left(x_{ij} - \sum_{m=1}^{M} \hat{a}_{im} \hat{b}_{jm} \right)^2 .$$

Return the estimated missing entries $\tilde{x}_{ij}$, $(i, j) \notin \mathcal{O}$.

Example using USA arrests data. The matrixx completion is applied to 10% of the matrix elements artificially set as missing and then imputed by 19.2 algorithm with only *M*=1 PC.

The true $x_{ij}$and imputed values $\hat{x}_{ij}$ for the standardized X values have an average correlation between the two sets of 0.63 with a~~n~~ standard deviation of 0.11, compared to an average correlation of 0.79 with an standard deviation of 0.08 if using the complete nonmissing data, hence the method proves a good solution for the missing data in this particular example.

19.9*Clustering.*

Clustering are a set of methods by which we establish similar subgroups of data, like PC, they are also unsupervised procedure for uncovering distinct clusters on the basis of the data but clustering differs in finding homogeneous subgroup~~d~~s rather than a few subgroups of dimensions. For example, clustering can identify the market segmented subgroups The two main clustering methods are *K-Means Clustering* and *Hierarchical Clustering* .

19.9.1 ***K-Means Custering*** . This method first partitions the data into a pre-specified number of clusters while the latter treats that number as an unkown using a tree-like representation called a *dendrogram* that views possible number of homogeneous clusters from 1 to *n*.

The idea on which K-Meeans Clustering is based is that within cluster vari~~in~~ation should be as small as possible to result in good clustering. Consider the *i*th observation in *k* clustering $i \in C_k$, then minimize the amount by which within a given cluster observations differ from each other by

$$\underset{C_1,\ldots,C_K}{\text{minimize}} \left\{ \sum_{k=1}^{K} W(C_k) \right\} \qquad (1939)$$

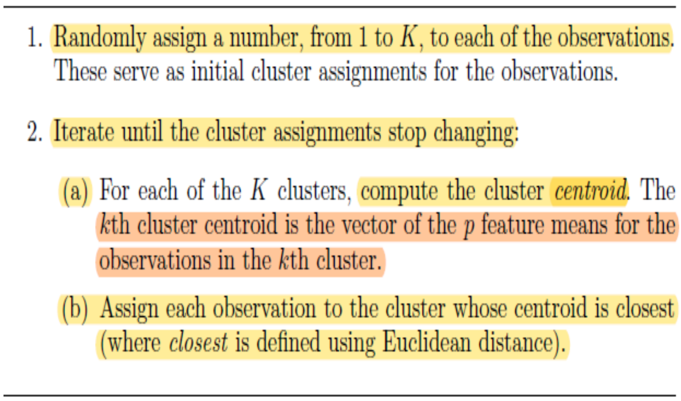
$$minimize_{c_1,\ldots,c_k}\{\textstyle\sum_{k=1}^{K} w(c_k)$$

choice of minimization is by squared Euclidian distance with which we define

$$minimize_{c_1,\ldots,c_k}\{\textstyle\sum_{k=1}^{K}\frac{1}{|c_K|}\sum_{i,i'\in c_k}\sum_{j=1}^{p}(x_{ij}-x_{i'j)}{}^{2}\} \qquad (19.40)$$

The solution employs the following *K-Means* algorithm.

19.3 *K-M*eans Algorithm

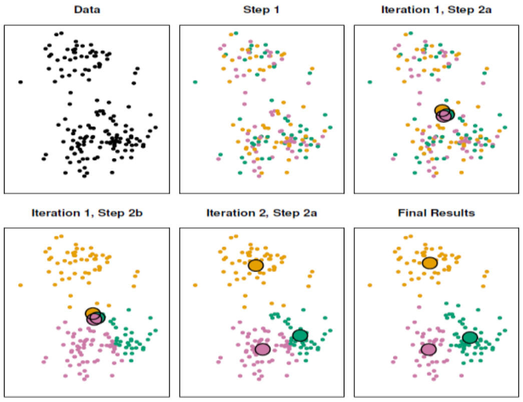


1. Randomly assign a number, from 1 to K, to each of the observations. These serve as initial cluster assignments for the observations.
2. Iterate until the cluster assignments stop changing:
   (a) For each of the K clusters, compute the cluster *centroid*. The kth cluster centroid is the vector of the p feature means for the observations in the kth cluster.
   (b) Assign each observation to the cluster whose centroid is closest (where *closest* is defined using Euclidean distance).

The step 2a computes the *cluster centroid*s as the mean of the observations in each cluster; hence the result depends on the initial randomly assigned clusters. To select the best solution the algorithm must be repeated with different initial values. Plot 19.3 presents an example of *K-M*eans Clustering for a simulated data set with *k*=3.

19.8.2 ***Hierachical Clustering***.

The disadvantge of *K-M*eans Clustering is its pre-specification of the number of cluster *K*. Hierachical Clustering has no such a requirement and the most common type aranges the data by a tree-based, bottom-up or agglomeerative clutstering known as a *dendrogram*. Plot 19.3 illustrates the construction and interpretation of a dendrogram with 9 obser~~s~~vations. Each leaf of the dendrogram on the left represents one of the observations of the plot on the right, as an unattached leaf; the lower in the tree fusions occur, the more similar the groups of observations are to each other. On the other hand, the vertical height of the fusions (on the left hand plot) indicate~~s~~ how different the two observations are. Therefore, the observations at the very bottom of the tree are very similar while those at the top are quite different. In general, there are 2^{n-1}possible reorderings of the dendrogram with *n* observations because at each of the *n*-1 fusion points, the position of the two fused branches could flip without affecting the dendrogram interpretation, Therefore, conclusions based on horizonal a~~x~~sis proximity cannot provide a measure of similarity, similarity must be based on the *vertical* ax~~s~~is where branches for two observations are first fused (left plot), making a dendrogram . In other words, the height of the cut to the dendrogram has the same function as *k* in *K-M*eans Clustering, making it a *Hierachical* method of controlling the number of clusters obtained.



Plot 19.3 *K-Means Clustering* with *k*=3

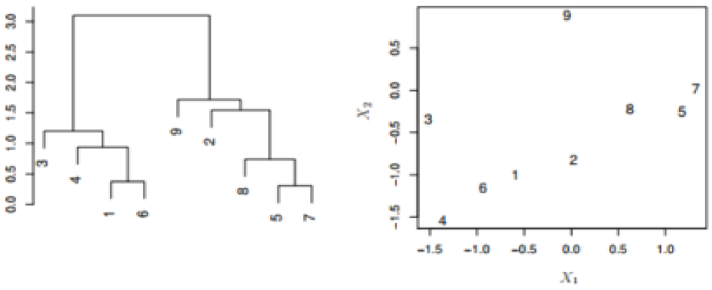
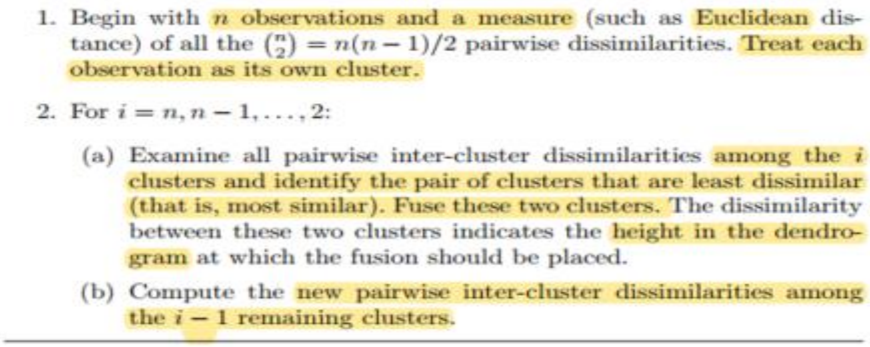


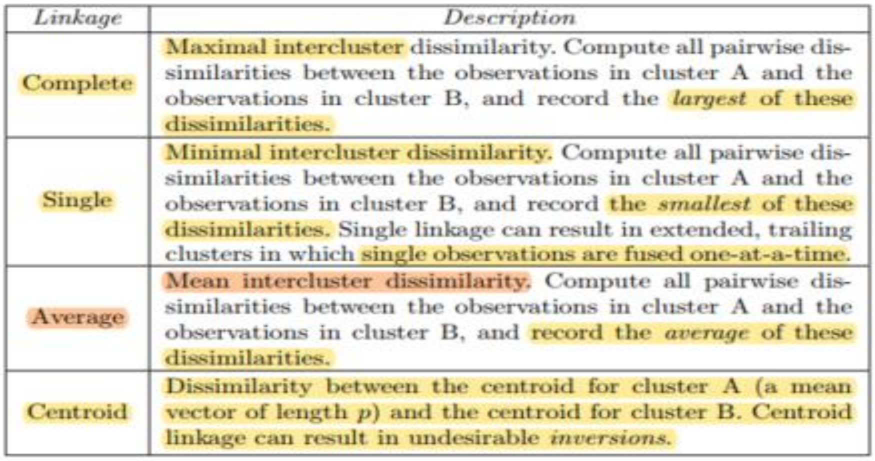
Fig. 19.10 Interpreting a Dendrogram

The dendrogran is defned by some measure of dissim*i*larity between each pair of observations; the most common *Euclidean distance*; at the bottom each *n* observation is treated as its own cluster, then the two most similar are fused, so now we have *n*-1 observations; then two most similar clusters are fused, leaving n-2 observations, etc. until the dendrogram is complete when all observations belong to a single cluster. The notion of *linkage* generalizes fu~~s~~sion of clusters containing multiple observations, for example {5, 7} wih {8}, is based on four common types−*Complete*, *Average, Single* and *Centroid* briefly described in the Algorithm 19.3.

19.3 Algorithm *Hierachical Clustering*



1. Begin with n observations and a measure (such as Euclidean distance) of all the $\binom{n}{2} = n(n-1)/2$ pairwise dissimilarities. Treat each observation as its own cluster.
2. For $i = n, n-1, \ldots, 2$:
   - (a) Examine all pairwise inter-cluster dissimilarities among the i clusters and identify the pair of clusters that are least dissimilar (that is, most similar). Fuse these two clusters. The dissimilarity between these two clusters indicates the height in the dendrogram at which the fusion should be placed.
   - (b) Compute the new pairwise inter-cluster dissimilarities among the $i-1$ remaining clusters.



Linkage	*Description*
Complete	Maximal intercluster dissimilarity. Compute all pairwise dissimilarities between the observations in cluster A and the observations in cluster B, and record the *largest* of these dissimilarities.
Single	Minimal intercluster dissimilarity. Compute all pairwise dissimilarities between the observations in cluster A and the observations in cluster B, and record the *smallest* of these dissimilarities. Single linkage can result in extended, trailing clusters in which single observations are fused one-at-a-time.
Average	Mean intercluster dissimilarity. Compute all pairwise dissimilarities between the observations in cluster A and the observations in cluster B, and record the *average* of these dissimilarities.
Centroid	Dissimilarity between the centroid for cluster A (a mean vector of length p) and the centroid for cluster B. Centroid linkage can result in undesirable *inversions*.

Note that the nested nature of Hierarchical Clustering can sometimes result in worse outcomes than *K-Means Clustering*. Suppose observations on men and women are evenly divided between Americans, Asians and Europeans, then split~~ting~~ into two groups by gender, and then divisions by continent; this procedure does not result in nested clusters.

19.9.3 ***Choice of dissimilarity*** . The observations may be far apart from each other by their Euclidean distance and yet highly correlated. Then *correlation-based distance* might be preferred that focuses on observation profiles rather than their magnitudes. For example, a Euclidean measure of online shoppers with infrequently purhased items using a {0, 1} binary variable would cluster together those with zero non-purchase, treating them as similar. That may not be desirable while those who bought items A and B but never item C or D, those with A and B purchased form better clusters using *correlation*-based distance. Therefore, choosing a dissimilarity measure is an important part of Hierarchical Clustering.

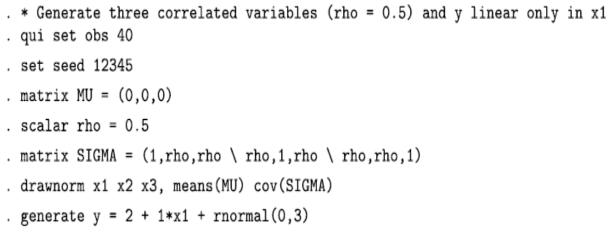
Selected Reading

James et. al. (2021) chapters 2 and 6 discuss linear Machine Learning models with many empirical examples in R; Hastie et.al (2001) chapters 3 and 4 coverr those models at greater depth and details. Cameron and Trivedi (2022), chapter 28 has several M.L. empirical examples of the linear models in Stata, Chapter 1 of the Chan and Ma'ty'as (2022) volume is a short and very good introduction to linear M.L. Tibshirani (1996) invented Lasso.

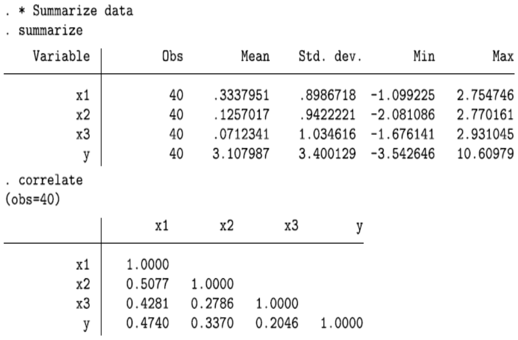
Lab Linear Shrinkage Models

Lab 1. Regress a continuous dependent variable y on three correlated normally distributed regressors, denoted x1, x2, and x3. The actual data-generating process (DGP) for y is the “true” linear model with an intercept and x1 alone (Many of the methods in this example can be adapted for other types of data such as binary outcomes and counts.)

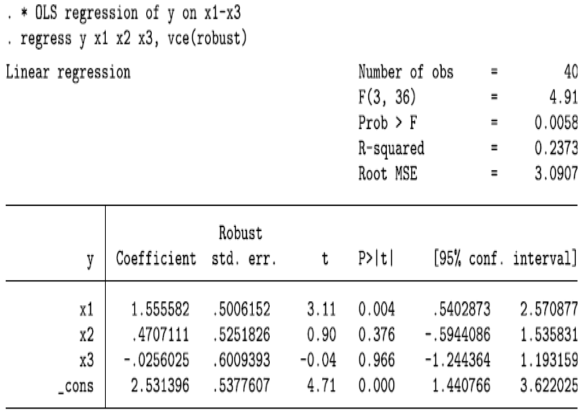
a) Obtain the descriptives and correlation among the variables and obtain all possible model estimations.



```
. * Generate three correlated variables (rho = 0.5) and y linear only in x1
. qui set obs 40
. set seed 12345
. matrix MU = (0,0,0)
. scalar rho = 0.5
. matrix SIGMA = (1,rho,rho \ rho,1,rho \ rho,rho,1)
. drawnorm x1 x2 x3, means(MU) cov(SIGMA)
. generate y = 2 + 1*x1 + rnormal(0,3)
```

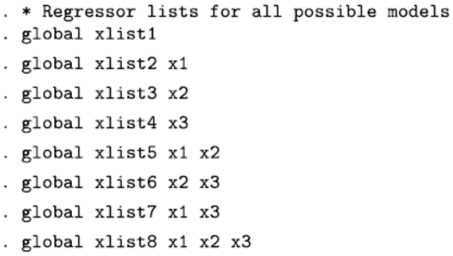


```
. * Summarize data
. summarize
    Variable |       Obs        Mean    Std. dev.       Min        Max
-------------+---------------------------------------------------------
          x1 |        40    .3337951    .8986718  -1.099225   2.754746
          x2 |        40    .1257017    .9422221  -2.081086   2.770161
          x3 |        40    .0712341    1.034616  -1.676141   2.931045
           y |        40    3.107987    3.400129  -3.542646   10.60979
. correlate
(obs=40)
             |       x1       x2       x3        y
-------------+------------------------------------
          x1 |   1.0000
          x2 |   0.5077   1.0000
          x3 |   0.4281   0.2786   1.0000
           y |   0.4740   0.3370   0.2046   1.0000
```



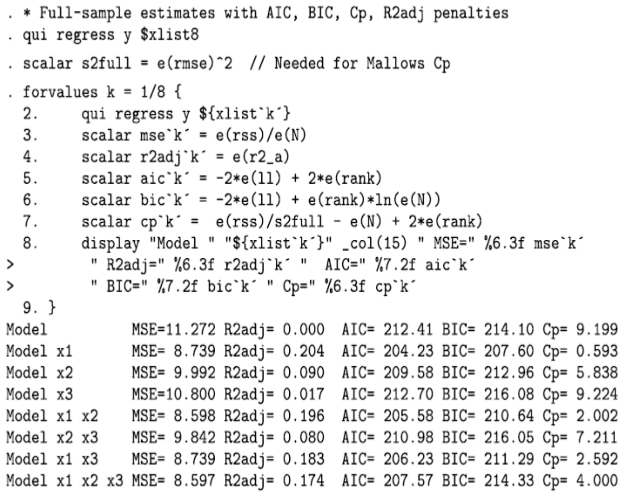
```
. * OLS regression of y on x1-x3
. regress y x1 x2 x3, vce(robust)

Linear regression                               Number of obs     =         40
                                                F(3, 36)          =       4.91
                                                Prob > F          =     0.0058
                                                R-squared         =     0.2373
                                                Root MSE          =     3.0907

------------------------------------------------------------------------------
             |               Robust
           y | Coefficient  std. err.      t    P>|t|     [95% conf. interval]
-------------+----------------------------------------------------------------
          x1 |   1.555582   .5006152     3.11   0.004     .5402873    2.570877
          x2 |   .4707111   .5251826     0.90   0.376    -.5944086    1.535831
          x3 |  -.0256025   .6009393    -0.04   0.966    -1.244364    1.193159
       _cons |   2.531396   .5377607     4.71   0.000     1.440766    3.622025
------------------------------------------------------------------------------
```

- All possible models.



```
. * Regressor lists for all possible models
. global xlist1
. global xlist2 x1
. global xlist3 x2
. global xlist4 x3
. global xlist5 x1 x2
. global xlist6 x2 x3
. global xlist7 x1 x3
. global xlist8 x1 x2 x3
```

b) Identify the best model by R^2, ajR^2, MSE, AIC, BIC.

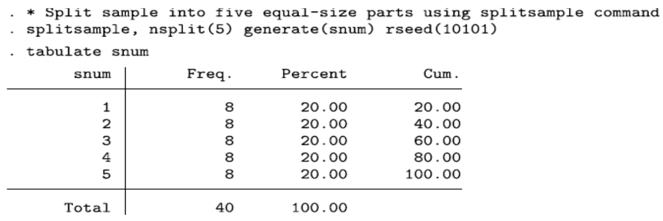


```
. * Full-sample estimates with AIC, BIC, Cp, R2adj penalties
. qui regress y $xlist8

. scalar s2full = e(rmse)^2  // Needed for Mallows Cp

. forvalues k = 1/8 {
  2.     qui regress y ${xlist`k´}
  3.     scalar mse`k´ = e(rss)/e(N)
  4.     scalar r2adj`k´ = e(r2_a)
  5.     scalar aic`k´ = -2*e(ll) + 2*e(rank)
  6.     scalar bic`k´ = -2*e(ll) + e(rank)*ln(e(N))
  7.     scalar cp`k´ =  e(rss)/s2full - e(N) + 2*e(rank)
  8.     display "Model " "${xlist`k´}" _col(15) " MSE=" %6.3f mse`k´
>         " R2adj=" %6.3f r2adj`k´ "  AIC=" %7.2f aic`k´
>         " BIC=" %7.2f bic`k´ " Cp=" %6.3f cp`k´
  9. }
Model         MSE=11.272 R2adj= 0.000  AIC= 212.41 BIC= 214.10 Cp= 9.199
Model x1      MSE= 8.739 R2adj= 0.204  AIC= 204.23 BIC= 207.60 Cp= 0.593
Model x2      MSE= 9.992 R2adj= 0.090  AIC= 209.58 BIC= 212.96 Cp= 5.838
Model x3      MSE=10.800 R2adj= 0.017  AIC= 212.70 BIC= 216.08 Cp= 9.224
Model x1 x2   MSE= 8.598 R2adj= 0.196  AIC= 205.58 BIC= 210.64 Cp= 2.002
Model x2 x3   MSE= 9.842 R2adj= 0.080  AIC= 210.98 BIC= 216.05 Cp= 7.211
Model x1 x3   MSE= 8.739 R2adj= 0.183  AIC= 206.23 BIC= 211.29 Cp= 2.592
Model x1 x2 x3 MSE= 8.597 R2adj= 0.174  AIC= 207.57 BIC= 214.33 Cp= 4.000
```

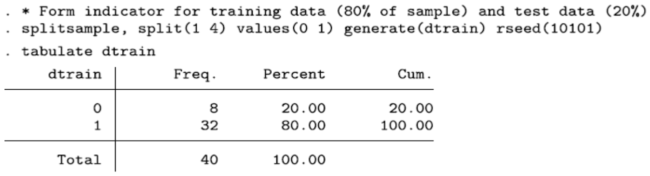
- the best model with intercept and x1 selected by the smallest MSE, BIC, etc.

Lab 2. Choosing a model by CV. Following from Lab1 data set, Sample split and select the best model by CV.

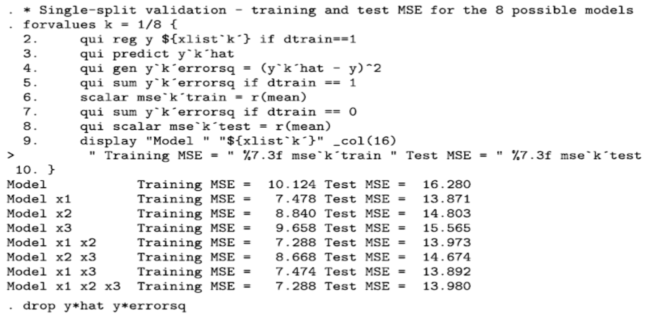
a) Split the sample observations into 80% training subsample and 20% test sample (out-of-sample) and identify the best model with the smallest MSE.



```
. * Split sample into five equal-size parts using splitsample command
. splitsample, nsplit(5) generate(snum) rseed(10101)

. tabulate snum

       snum |      Freq.     Percent        Cum.
------------+-----------------------------------
          1 |          8       20.00       20.00
          2 |          8       20.00       40.00
          3 |          8       20.00       60.00
          4 |          8       20.00       80.00
          5 |          8       20.00      100.00
------------+-----------------------------------
      Total |         40      100.00
```

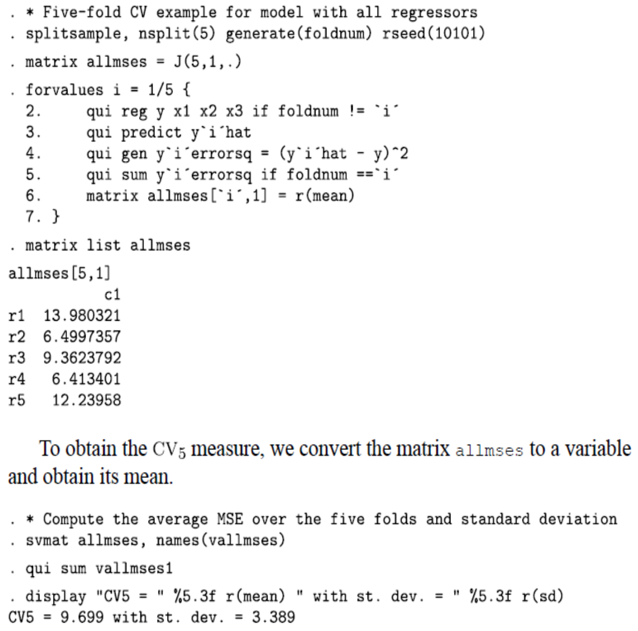


```
. * Form indicator for training data (80% of sample) and test data (20%)
. splitsample, split(1 4) values(0 1) generate(dtrain) rseed(10101)

. tabulate dtrain

     dtrain |      Freq.     Percent        Cum.
------------+-----------------------------------
          0 |          8       20.00       20.00
          1 |         32       80.00      100.00
------------+-----------------------------------
      Total |         40      100.00
```



```
. * Single-split validation - training and test MSE for the 8 possible models
. forvalues k = 1/8 {
  2.     qui reg y ${xlist`k´} if dtrain==1
  3.     qui predict y`k´hat
  4.     qui gen y`k´errorsq = (y`k´hat - y)^2
  5.     qui sum y`k´errorsq if dtrain == 1
  6.     scalar mse`k´train = r(mean)
  7.     qui sum y`k´errorsq if dtrain == 0
  8.     qui scalar mse`k´test = r(mean)
  9.     display "Model " "${xlist`k´}" _col(16)
>         " Training MSE = " %7.3f mse`k´train " Test MSE = " %7.3f mse`k´test
 10. }
Model          Training MSE =  10.124 Test MSE =  16.280
Model x1       Training MSE =   7.478 Test MSE =  13.871
Model x2       Training MSE =   8.840 Test MSE =  14.803
Model x3       Training MSE =   9.658 Test MSE =  15.565
Model x1 x2    Training MSE =   7.288 Test MSE =  13.973
Model x2 x3    Training MSE =   8.668 Test MSE =  14.674
Model x1 x3    Training MSE =   7.474 Test MSE =  13.892
Model x1 x2 x3 Training MSE =   7.288 Test MSE =  13.980

. drop y*hat y*errorsq
```

b) Split the sample by K-fold CV with K=5 using *crossfold* Stata command.

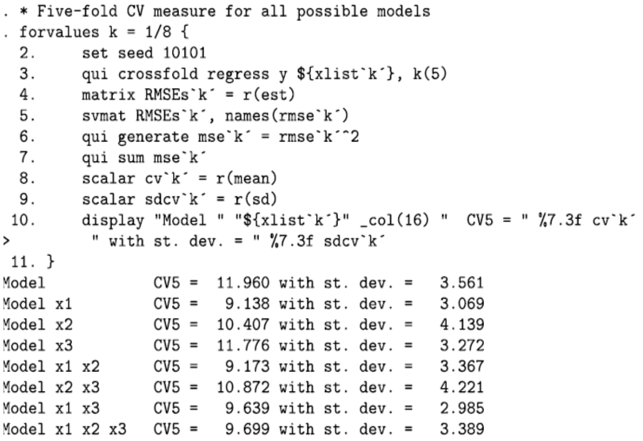


```
. * Five-fold CV example for model with all regressors
. splitsample, nsplit(5) generate(foldnum) rseed(10101)

. matrix allmses = J(5,1,.)

. forvalues i = 1/5 {
  2.      qui reg y x1 x2 x3 if foldnum != `i´
  3.      qui predict y`i´hat
  4.      qui gen y`i´errorsq = (y`i´hat - y)^2
  5.      qui sum y`i´errorsq if foldnum ==`i´
  6.      matrix allmses[`i´,1] = r(mean)
  7. }

. matrix list allmses

allmses[5,1]
           c1
r1  13.980321
r2  6.4997357
r3  9.3623792
r4   6.413401
r5   12.23958
```

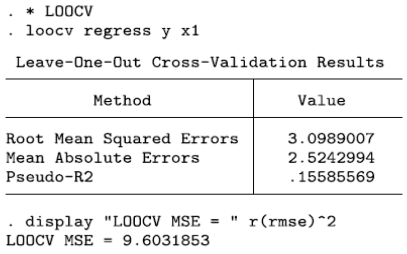
To obtain the CV_5 measure, we convert the matrix `allmses` to a variable and obtain its mean.

```
. * Compute the average MSE over the five folds and standard deviation
. svmat allmses, names(vallmses)

. qui sum vallmses1

. display "CV5 = " %5.3f r(mean) " with st. dev. = " %5.3f r(sd)
CV5 = 9.699 with st. dev. = 3.389
```

With all 8 models:



```
. * Five-fold CV measure for all possible models
. forvalues k = 1/8 {
  2.      set seed 10101
  3.      qui crossfold regress y ${xlist`k´}, k(5)
  4.      matrix RMSEs`k´ = r(est)
  5.      svmat RMSEs`k´, names(rmse`k´)
  6.      qui generate mse`k´ = rmse`k´^2
  7.      qui sum mse`k´
  8.      scalar cv`k´ = r(mean)
  9.      scalar sdcv`k´ = r(sd)
 10.      display "Model " "${xlist`k´}" _col(16) "  CV5 = " %7.3f cv`k´
>          " with st. dev. = " %7.3f sdcv`k´
 11. }
Model            CV5 =  11.960 with st. dev. =   3.561
Model x1         CV5 =   9.138 with st. dev. =   3.069
Model x2         CV5 =  10.407 with st. dev. =   4.139
Model x3         CV5 =  11.776 with st. dev. =   3.272
Model x1 x2      CV5 =   9.173 with st. dev. =   3.367
Model x2 x3      CV5 =  10.872 with st. dev. =   4.221
Model x1 x3      CV5 =   9.639 with st. dev. =   2.985
Model x1 x2 x3   CV5 =   9.699 with st. dev. =   3.389
```



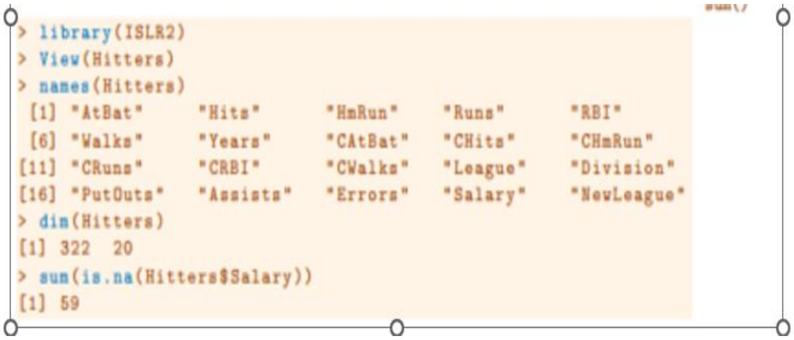
```
. * LOOCV
. loocv regress y x1

 Leave-One-Out Cross-Validation Results
 ---------------------------------------
          Method         |     Value
 ------------------------+--------------
 Root Mean Squared Errors|   3.0989007
 Mean Absolute Errors    |   2.5242994
 Pseudo-R2               |   .15585569
 ---------------------------------------

. display "LOOCV MSE = " r(rmse)^2
LOOCV MSE = 9.6031853
```

Lab 3. Stepwise selection. Use the previous linear model for stepwise select using Stata *vselect* command based on ajR2, AIC, BIC, or AICC (a bias-corrected version of AIC).

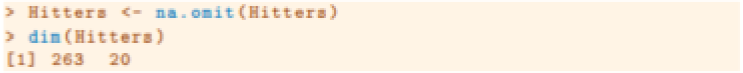
```
. * Best subset selection with community-contributed command vselect
. vselect y x1 x2 x3, best

Response :             y
Selected predictors:   x1 x2 x3

Optimal models:

   # Preds     R2ADJ         C       AIC      AICC       BIC
         1  .2043123  .5925225  204.2265  204.8932  207.6042
         2  .1959877  2.002325  205.5761  206.7189  210.6427
         3  .1737073         4  207.5735  209.3382   214.329

predictors for each model:

1  :  x1
2  :  x1 x2
3  :  x1 x2 x3
```

Lab 4. Best Subset Selection. Open hitters data on baseball players' *salary*, report the descriptives after removing the missing values.

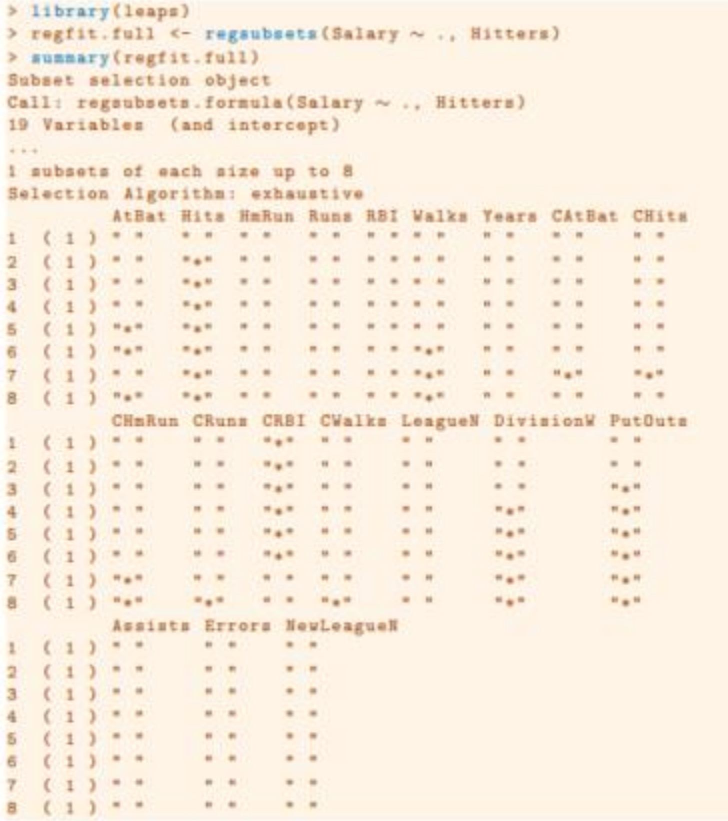
a) Select the best model for a given number of predictors using the selection methods in Lab1-3 and output the best set of variables for each model size.



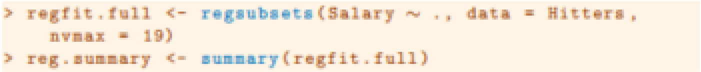
```
> library(ISLR2)
> View(Hitters)
> names(Hitters)
 [1] "AtBat"     "Hits"      "HmRun"     "Runs"      "RBI"
 [6] "Walks"     "Years"     "CAtBat"    "CHits"     "CHmRun"
[11] "CRuns"     "CRBI"      "CWalks"    "League"    "Division"
[16] "PutOuts"   "Assists"   "Errors"    "Salary"    "NewLeague"
> dim(Hitters)
[1] 322  20
> sum(is.na(Hitters$Salary))
[1] 59
```



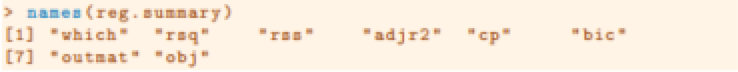
```
> Hitters <- na.omit(Hitters)
> dim(Hitters)
[1] 263  20
```



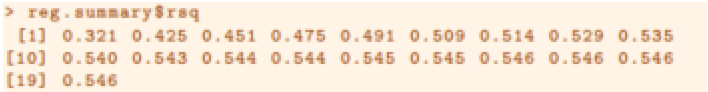
```
> library(leaps)
> regfit.full <- regsubsets(Salary ~ ., Hitters)
> summary(regfit.full)
Subset selection object
Call: regsubsets.formula(Salary ~ ., Hitters)
19 Variables  (and intercept)
...
1 subsets of each size up to 8
Selection Algorithm: exhaustive
         AtBat Hits HmRun Runs RBI Walks Years CAtBat CHits
1  ( 1 ) " "   " "  " "   " "  " " " "   " "   " "    " "
2  ( 1 ) " "   "*"  " "   " "  " " " "   " "   " "    " "
3  ( 1 ) " "   "*"  " "   " "  " " " "   " "   " "    " "
4  ( 1 ) " "   "*"  " "   " "  " " " "   " "   " "    " "
5  ( 1 ) "*"   "*"  " "   " "  " " " "   " "   " "    " "
6  ( 1 ) "*"   "*"  " "   " "  " " "*"   " "   " "    " "
7  ( 1 ) " "   "*"  " "   " "  " " "*"   " "   "*"    "*"
8  ( 1 ) "*"   "*"  " "   " "  " " "*"   " "   " "    " "
         CHmRun CRuns CRBI CWalks LeagueN DivisionW PutOuts
1  ( 1 ) " "    " "   "*"  " "    " "     " "       " "
2  ( 1 ) " "    " "   "*"  " "    " "     " "       " "
3  ( 1 ) " "    " "   "*"  " "    " "     " "       "*"
4  ( 1 ) " "    " "   "*"  " "    " "     "*"       "*"
5  ( 1 ) " "    " "   "*"  " "    " "     "*"       "*"
6  ( 1 ) " "    " "   "*"  " "    " "     "*"       "*"
7  ( 1 ) "*"    " "   " "  " "    " "     "*"       "*"
8  ( 1 ) "*"    "*"   " "  "*"    " "     "*"       "*"
         Assists Errors NewLeagueN
1  ( 1 ) " "     " "    " "
2  ( 1 ) " "     " "    " "
3  ( 1 ) " "     " "    " "
4  ( 1 ) " "     " "    " "
5  ( 1 ) " "     " "    " "
6  ( 1 ) " "     " "    " "
7  ( 1 ) " "     " "    " "
8  ( 1 ) " "     " "    " "
```



```
> regfit.full <- regsubsets(Salary ~ ., data = Hitters,
    nvmax = 19)
> reg.summary <- summary(regfit.full)
```

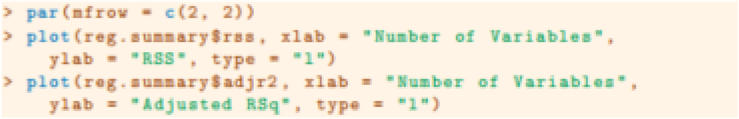


```
> names(reg.summary)
[1] "which"  "rsq"    "rss"    "adjr2"  "cp"     "bic"
[7] "outmat" "obj"
```

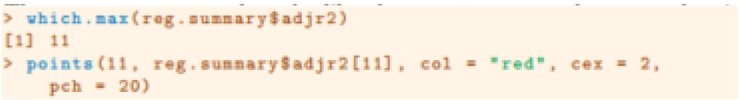


```
> reg.summary$rsq
 [1] 0.321 0.425 0.451 0.475 0.491 0.509 0.514 0.529 0.535
[10] 0.540 0.543 0.544 0.544 0.545 0.545 0.546 0.546 0.546
[19] 0.546
```

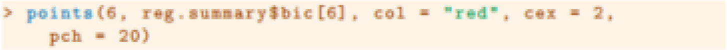
b) Plot all of the models at once by the selection methods to identify the best overall model,



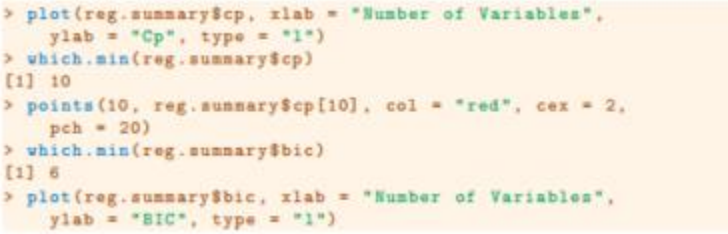
```
> par(mfrow = c(2, 2))
> plot(reg.summary$rss, xlab = "Number of Variables",
    ylab = "RSS", type = "l")
> plot(reg.summary$adjr2, xlab = "Number of Variables",
    ylab = "Adjusted RSq", type = "l")
```



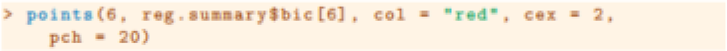
```
> which.max(reg.summary$adjr2)
[1] 11
> points(11, reg.summary$adjr2[11], col = "red", cex = 2,
    pch = 20)
```



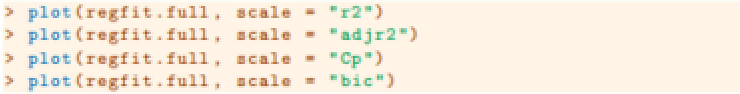
```
> points(6, reg.summary$bic[6], col = "red", cex = 2,
    pch = 20)
```



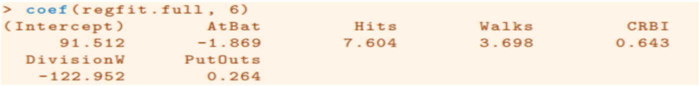
```
> plot(reg.summary$cp, xlab = "Number of Variables",
    ylab = "Cp", type = "l")
> which.min(reg.summary$cp)
[1] 10
> points(10, reg.summary$cp[10], col = "red", cex = 2,
    pch = 20)
> which.min(reg.summary$bic)
[1] 6
> plot(reg.summary$bic, xlab = "Number of Variables",
    ylab = "BIC", type = "l")
```



```
> points(6, reg.summary$bic[6], col = "red", cex = 2,
    pch = 20)
```

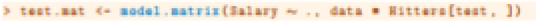


```
> plot(regfit.full, scale = "r2")
> plot(regfit.full, scale = "adjr2")
> plot(regfit.full, scale = "Cp")
> plot(regfit.full, scale = "bic")
```

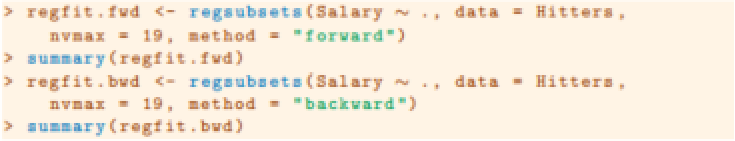


```
> coef(regfit.full, 6)
(Intercept)        AtBat         Hits        Walks         CRBI
     91.512       -1.869        7.604        3.698        0.643
  DivisionW      PutOuts
   -122.952        0.264
```

- The lowest BIC is the six-variable model resulting in the above coefficient estimates.



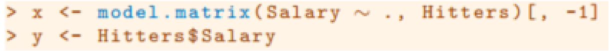
```
> test.mat <- model.matrix(Salary ~ ., data = Hitters[test, ])
```



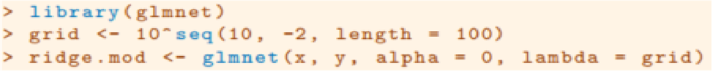
```
> regfit.fwd <- regsubsets(Salary ~ ., data = Hitters,
    nvmax = 19, method = "forward")
> summary(regfit.fwd)
> regfit.bwd <- regsubsets(Salary ~ ., data = Hitters,
    nvmax = 19, method = "backward")
> summary(regfit.bwd)
```

Lab 5. Ridge and Lasso. Open data file Hitters

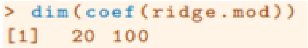
a) Estimate a Ridge regression at small and large values of λ.



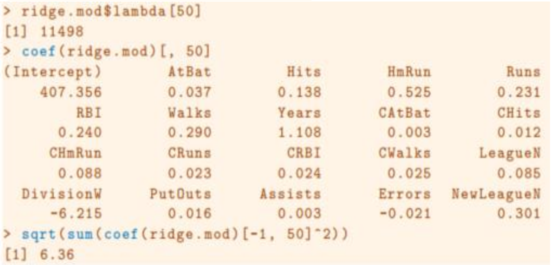
```
> x <- model.matrix(Salary ~ ., Hitters)[, -1]
> y <- Hitters$Salary
```



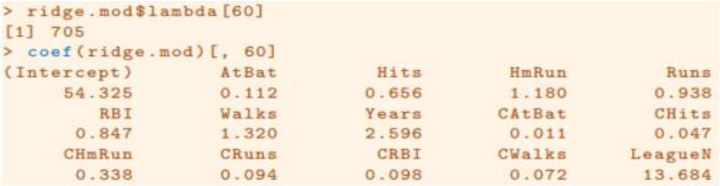
```
> library(glmnet)
> grid <- 10^seq(10, -2, length = 100)
> ridge.mod <- glmnet(x, y, alpha = 0, lambda = grid)
```



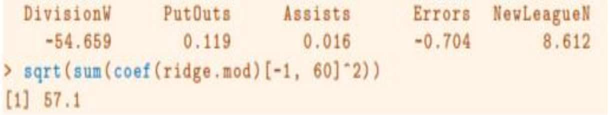
```
> dim(coef(ridge.mod))
[1]  20 100
```



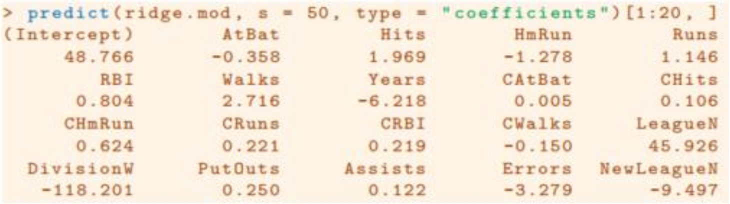
```
> ridge.mod$lambda[50]
[1] 11498
> coef(ridge.mod)[, 50]
(Intercept)       AtBat        Hits       HmRun        Runs
    407.356       0.037       0.138       0.525       0.231
        RBI       Walks       Years      CAtBat       CHits
      0.240       0.290       1.108       0.003       0.012
     CHmRun       CRuns        CRBI      CWalks     LeagueN
      0.088       0.023       0.024       0.025       0.085
  DivisionW     PutOuts     Assists      Errors  NewLeagueN
     -6.215       0.016       0.003      -0.021       0.301
> sqrt(sum(coef(ridge.mod)[-1, 50]^2))
[1] 6.36
```



```
> ridge.mod$lambda[60]
[1] 705
> coef(ridge.mod)[, 60]
(Intercept)        AtBat         Hits        HmRun         Runs
     54.325        0.112        0.656        1.180        0.938
        RBI        Walks        Years       CAtBat        CHits
      0.847        1.320        2.596        0.011        0.047
     CHmRun        CRuns         CRBI       CWalks      LeagueN
      0.338        0.094        0.098        0.072       13.684
```

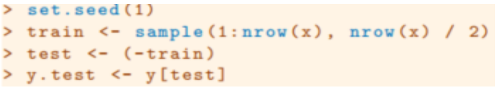


```
  DivisionW     PutOuts     Assists      Errors  NewLeagueN
    -54.659       0.119       0.016      -0.704       8.612
> sqrt(sum(coef(ridge.mod)[-1, 60]^2))
[1] 57.1
```

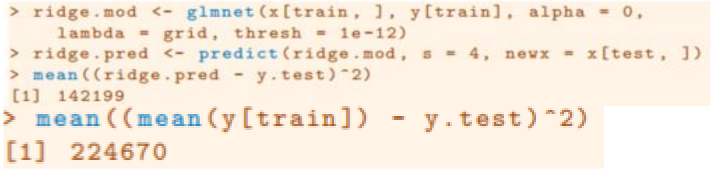


```
> predict(ridge.mod, s = 50, type = "coefficients")[1:20, ]
(Intercept)        AtBat         Hits        HmRun         Runs
     48.766       -0.358        1.969       -1.278        1.146
        RBI        Walks        Years       CAtBat        CHits
      0.804        2.716       -6.218        0.005        0.106
     CHmRun        CRuns         CRBI       CWalks      LeagueN
      0.624        0.221        0.219       -0.150       45.926
  DivisionW      PutOuts      Assists       Errors   NewLeagueN
   -118.201        0.250        0.122       -3.279       -9.497
```

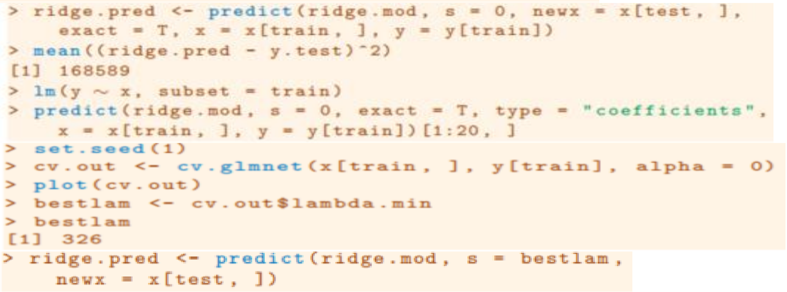
b) Split the sample into training and test samples in order to estimate the test error without and cross-validation of λ to obtain the lowest predicted error



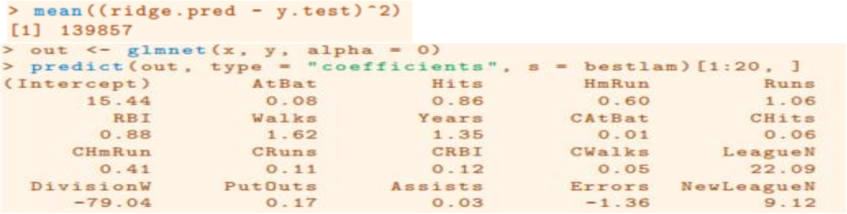
```
> set.seed(1)
> train <- sample(1:nrow(x), nrow(x) / 2)
> test <- (-train)
> y.test <- y[test]
```



```
> ridge.mod <- glmnet(x[train, ], y[train], alpha = 0,
    lambda = grid, thresh = 1e-12)
> ridge.pred <- predict(ridge.mod, s = 4, newx = x[test, ])
> mean((ridge.pred - y.test)^2)
[1] 142199
> mean((mean(y[train]) - y.test)^2)
[1] 224670
```

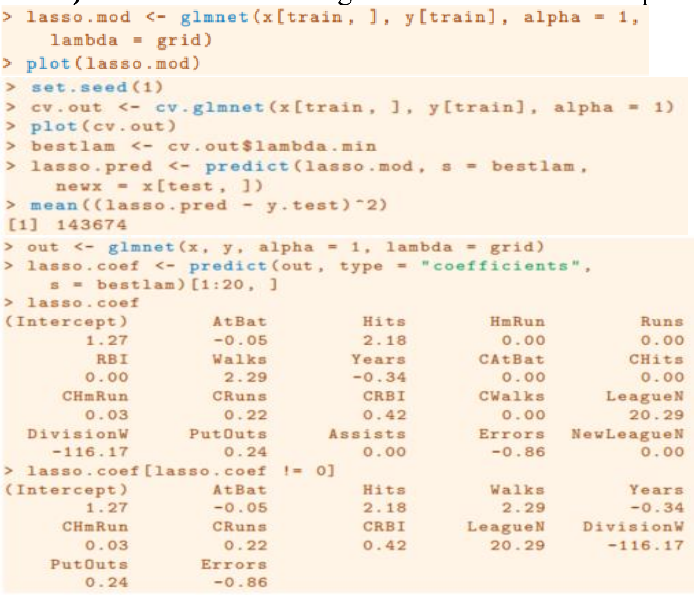


```
> ridge.pred <- predict(ridge.mod, s = 0, newx = x[test, ],
    exact = T, x = x[train, ], y = y[train])
> mean((ridge.pred - y.test)^2)
[1] 168589
> lm(y ~ x, subset = train)
> predict(ridge.mod, s = 0, exact = T, type = "coefficients",
    x = x[train, ], y = y[train])[1:20, ]
> set.seed(1)
> cv.out <- cv.glmnet(x[train, ], y[train], alpha = 0)
> plot(cv.out)
> bestlam <- cv.out$lambda.min
> bestlam
[1] 326
> ridge.pred <- predict(ridge.mod, s = bestlam,
    newx = x[test, ])
```



```
> mean((ridge.pred - y.test)^2)
[1] 139857
> out <- glmnet(x, y, alpha = 0)
> predict(out, type = "coefficients", s = bestlam)[1:20, ]
(Intercept)       AtBat        Hits       HmRun        Runs
      15.44        0.08        0.86        0.60        1.06
        RBI       Walks       Years      CAtBat       CHits
       0.88        1.62        1.35        0.01        0.06
     CHmRun       CRuns        CRBI      CWalks     LeagueN
       0.41        0.11        0.12        0.05       22.09
  DivisionW     PutOuts     Assists      Errors  NewLeagueN
     -79.04        0.17        0.03       -1.36        9.12
```

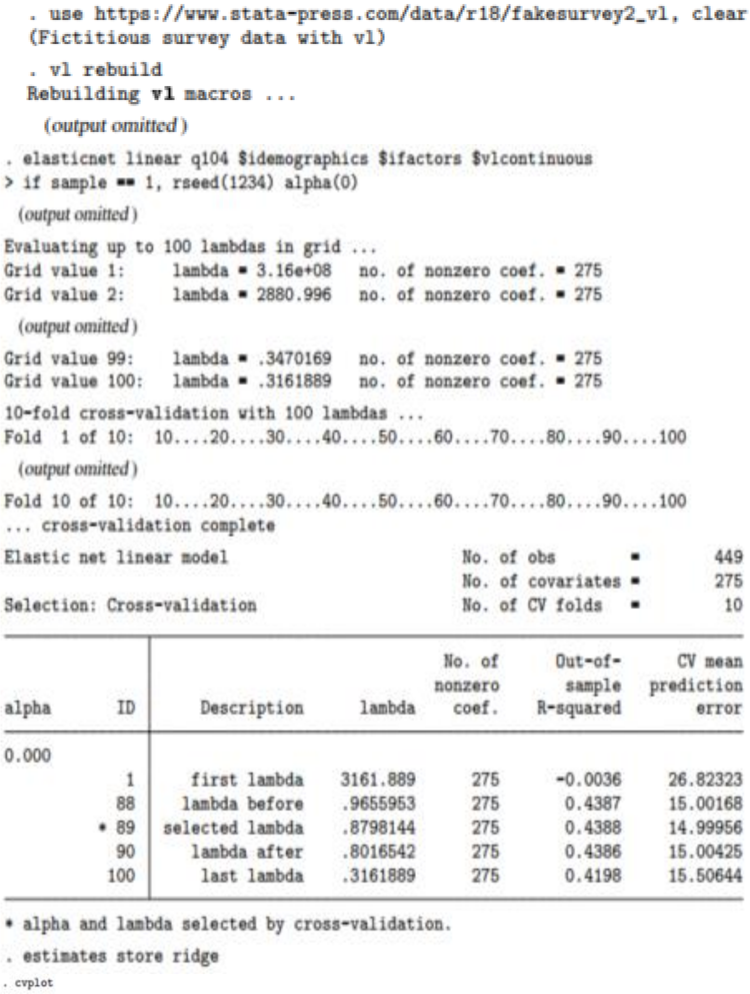
c) Now run a Lasso regression for the same sample.



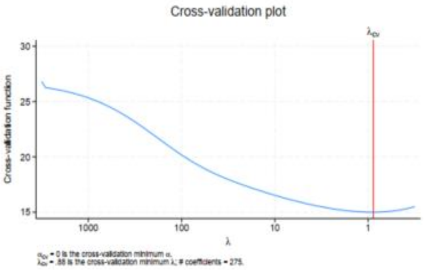
```
> lasso.mod <- glmnet(x[train, ], y[train], alpha = 1,
    lambda = grid)
> plot(lasso.mod)
> set.seed(1)
> cv.out <- cv.glmnet(x[train, ], y[train], alpha = 1)
> plot(cv.out)
> bestlam <- cv.out$lambda.min
> lasso.pred <- predict(lasso.mod, s = bestlam,
    newx = x[test, ])
> mean((lasso.pred - y.test)^2)
[1] 143674
> out <- glmnet(x, y, alpha = 1, lambda = grid)
> lasso.coef <- predict(out, type = "coefficients",
    s = bestlam)[1:20, ]
> lasso.coef
(Intercept)       AtBat        Hits       HmRun        Runs
       1.27       -0.05        2.18        0.00        0.00
        RBI       Walks       Years      CAtBat       CHits
       0.00        2.29       -0.34        0.00        0.00
     CHmRun       CRuns        CRBI      CWalks     LeagueN
       0.03        0.22        0.42        0.00       20.29
  DivisionW     PutOuts     Assists      Errors  NewLeagueN
    -116.17        0.24        0.00       -0.86        0.00
> lasso.coef[lasso.coef != 0]
(Intercept)       AtBat        Hits       Walks       Years
       1.27       -0.05        2.18        2.29       -0.34
     CHmRun       CRuns        CRBI     LeagueN   DivisionW
       0.03        0.22        0.42       20.29     -116.17
    PutOuts      Errors
       0.24       -0.86
```

Lab 6. Lasso-Ridge-elastic comparison.

a) Open data file fakesurvey2-vi.dta, apply elastic net regression with 10-fold CV.

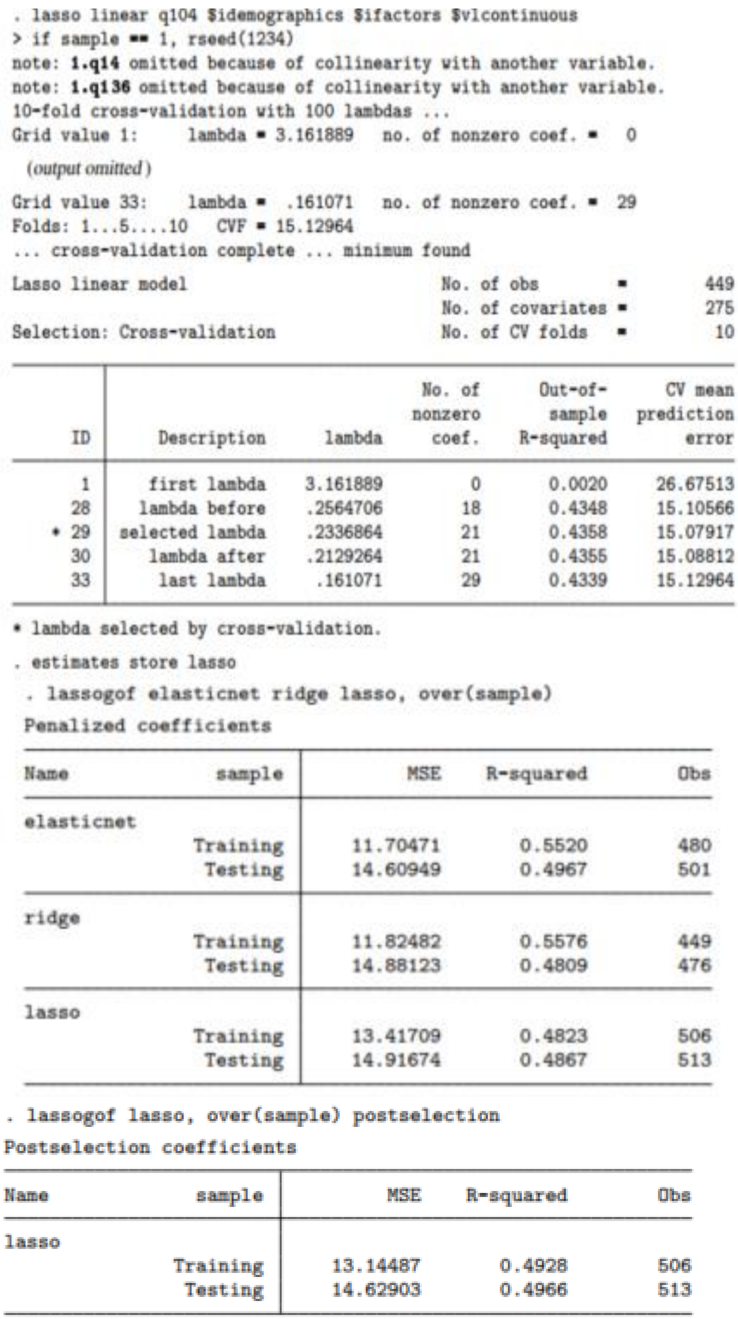


```
  . use https://www.stata-press.com/data/r18/fakesurvey2_vl, clear
  (Fictitious survey data with vl)

  . vl rebuild
  Rebuilding vl macros ...
    (output omitted)

. elasticnet linear q104 $idemographics $ifactors $vlcontinuous
> if sample == 1, rseed(1234) alpha(0)
  (output omitted)

Evaluating up to 100 lambdas in grid ...
Grid value 1:     lambda = 3.16e+08   no. of nonzero coef. = 275
Grid value 2:     lambda = 2880.996   no. of nonzero coef. = 275
  (output omitted)

Grid value 99:    lambda = .3470169   no. of nonzero coef. = 275
Grid value 100:   lambda = .3161889   no. of nonzero coef. = 275

10-fold cross-validation with 100 lambdas ...
Fold  1 of 10:  10....20....30....40....50....60....70....80....90....100
  (output omitted)

Fold 10 of 10:  10....20....30....40....50....60....70....80....90....100
... cross-validation complete

Elastic net linear model                   No. of obs        =       449
                                           No. of covariates =       275
Selection: Cross-validation                No. of CV folds   =        10

-----------------------------------------------------------------------------
                                                No. of    Out-of-     CV mean
                                               nonzero     sample  prediction
alpha         ID     Description      lambda     coef.  R-squared       error
-----------------------------------------------------------------------------
0.000
               1     first lambda   3161.889       275    -0.0036    26.82323
              88    lambda before   .9655953       275     0.4387    15.00168
            * 89  selected lambda   .8798144       275     0.4388    14.99956
              90     lambda after   .8016542       275     0.4386    15.00425
             100      last lambda   .3161889       275     0.4198    15.50644
-----------------------------------------------------------------------------
* alpha and lambda selected by cross-validation.

. estimates store ridge

. cvplot
```



α_{CV} = 0 is the cross-validation minimum α.
λ_{CV} = .88 is the cross-validation minimum λ; # coefficients = 275.

b) Compare elastic net, Ridge and Lasso regressions test MSE and report Lasso coefficient estimates.

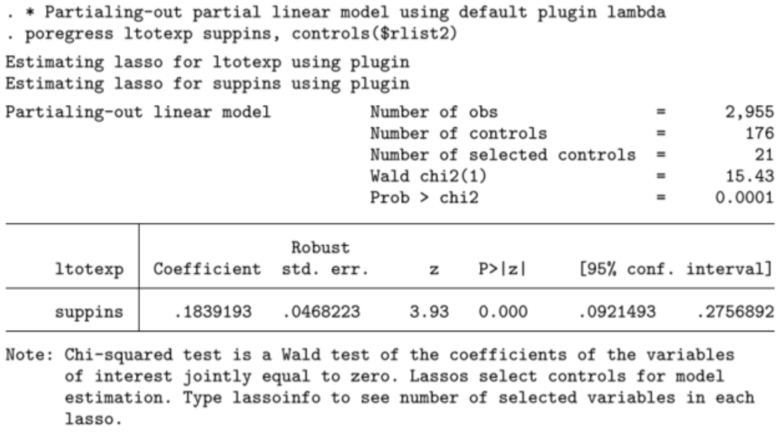


```
. lasso linear q104 $idemographics $ifactors $vlcontinuous
> if sample == 1, rseed(1234)
note: 1.q14 omitted because of collinearity with another variable.
note: 1.q136 omitted because of collinearity with another variable.
10-fold cross-validation with 100 lambdas ...
Grid value 1:     lambda = 3.161889   no. of nonzero coef. =   0
  (output omitted)
Grid value 33:    lambda =  .161071   no. of nonzero coef. =  29
Folds: 1...5....10   CVF = 15.12964
... cross-validation complete ... minimum found

Lasso linear model                         No. of obs         =       449
                                           No. of covariates  =       275
Selection: Cross-validation                No. of CV folds    =        10

--------------------------------------------------------------------------
         |                                No. of    Out-of-      CV mean
         |                               nonzero     sample   prediction
      ID |     Description      lambda     coef.  R-squared        error
---------+----------------------------------------------------------------
       1 |    first lambda    3.161889         0     0.0020     26.67513
      28 |   lambda before    .2564706        18     0.4348     15.10566
    * 29 | selected lambda    .2336864        21     0.4358     15.07917
      30 |    lambda after    .2129264        21     0.4355     15.08812
      33 |     last lambda     .161071        29     0.4339     15.12964
--------------------------------------------------------------------------
* lambda selected by cross-validation.

. estimates store lasso

 . lassogof elasticnet ridge lasso, over(sample)
 Penalized coefficients

 -----------------------------------------------------------------------
 Name            sample |         MSE    R-squared        Obs
 -----------------------+-----------------------------------------------
 elasticnet             |
               Training |    11.70471       0.5520        480
                Testing |    14.60949       0.4967        501
 -----------------------+-----------------------------------------------
 ridge                  |
               Training |    11.82482       0.5576        449
                Testing |    14.88123       0.4809        476
 -----------------------+-----------------------------------------------
 lasso                  |
               Training |    13.41709       0.4823        506
                Testing |    14.91674       0.4867        513
 -----------------------------------------------------------------------

. lassogof lasso, over(sample) postselection
Postselection coefficients

-----------------------------------------------------------------------
Name            sample |         MSE    R-squared        Obs
-----------------------+-----------------------------------------------
lasso                  |
              Training |    13.14487       0.4928        506
               Testing |    14.62903       0.4966        513
-----------------------------------------------------------------------
```

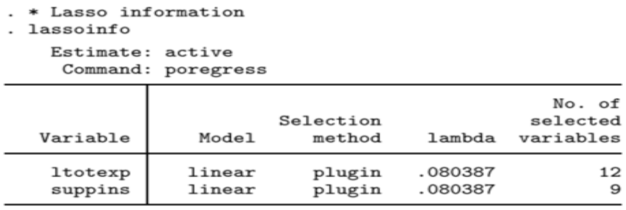
***Lab** 7*. M.L. Inference. Partialing-out- see also below, *Lab_x 4.*

Open data file mus203empdmdexp.dta, the sample contains log of total health expenditure and other 19 basic variables-14 binary and 5 continuous variables.

a) Divide the sample into 80% training sample and a 20% test or hold-out sample, fit a Lasso partialing-out regress of *ltotexp* on *suppins,* controlling remaining Lasso selected control variables using Stata command *poregression* with a penalty parameter λ selected by a plugin formula.

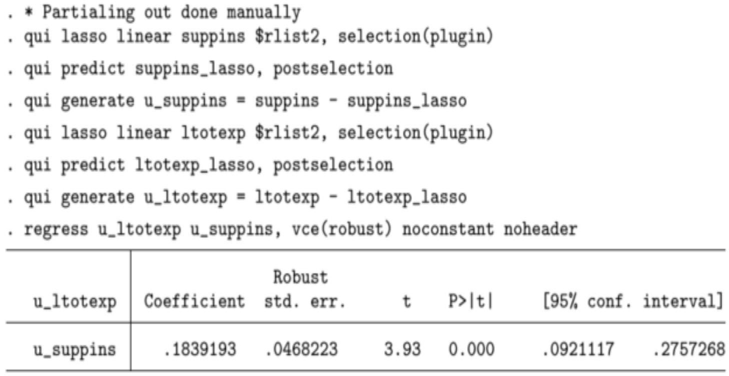


```
. * Partialing-out partial linear model using default plugin lambda
. poregress ltotexp suppins, controls($rlist2)

Estimating lasso for ltotexp using plugin
Estimating lasso for suppins using plugin

Partialing-out linear model          Number of obs               =      2,955
                                     Number of controls          =        176
                                     Number of selected controls =         21
                                     Wald chi2(1)                =      15.43
                                     Prob > chi2                 =     0.0001

------------------------------------------------------------------------------
             |               Robust
     ltotexp | Coefficient  std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
     suppins |   .1839193   .0468223     3.93   0.000     .0921493    .2756892
------------------------------------------------------------------------------
Note: Chi-squared test is a Wald test of the coefficients of the variables
      of interest jointly equal to zero. Lassos select controls for model
      estimation. Type lassoinfo to see number of selected variables in each
      lasso.
```



```
. * Lasso information
. lassoinfo

    Estimate: active
     Command: poregress

    ------------------------------------------------------------
                 |                                      No. of
                 |              Selection              selected
        Variable |      Model      method     lambda  variables
    -------------+----------------------------------------------
         ltotexp |     linear      plugin    .080387         12
         suppins |     linear      plugin    .080387          9
    ------------------------------------------------------------
```

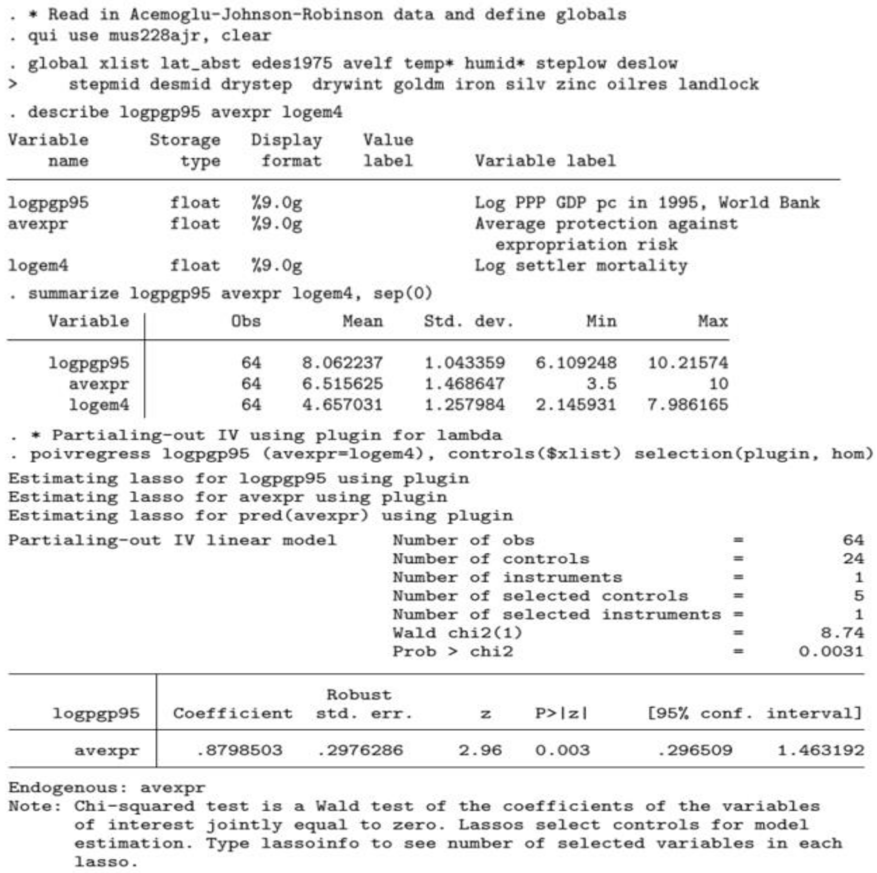
b) Repeat the above manually.



```
. * Partialing out done manually
. qui lasso linear suppins $rlist2, selection(plugin)

. qui predict suppins_lasso, postselection

. qui generate u_suppins = suppins - suppins_lasso

. qui lasso linear ltotexp $rlist2, selection(plugin)

. qui predict ltotexp_lasso, postselection

. qui generate u_ltotexp = ltotexp - ltotexp_lasso

. regress u_ltotexp u_suppins, vce(robust) noconstant noheader

------------------------------------------------------------------------------
             |               Robust
   u_ltotexp | Coefficient  std. err.      t    P>|t|     [95% conf. interval]
-------------+----------------------------------------------------------------
   u_suppins |   .1839193   .0468223     3.93   0.000     .0921117    .2757268
------------------------------------------------------------------------------
```

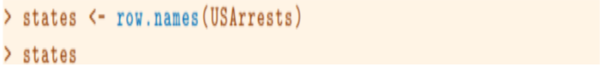
Lab 8. Fit Cross Partialing-out IV & Double selection, see also below, *lab_x 5***.**

Open cross-sectional data file *mus228ajr.dta*, define globals for the variables and fit a partialing-out Lasso_IV model of per capita income *loggdp95* to shrink the number of potential instruments and exogenous variables using plugin λ.

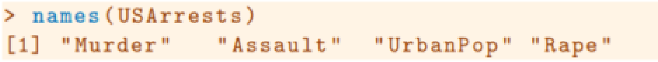


```
. * Read in Acemoglu-Johnson-Robinson data and define globals
. qui use mus228ajr, clear
. global xlist lat_abst edes1975 avelf temp* humid* steplow deslow
>     stepmid desmid drystep  drywint goldm iron silv zinc oilres landlock
. describe logpgp95 avexpr logem4
Variable      Storage   Display    Value
    name         type    format    label      Variable label
-----------------------------------------------------------------
logpgp95        float   %9.0g                 Log PPP GDP pc in 1995, World Bank
avexpr          float   %9.0g                 Average protection against
                                                expropriation risk
logem4          float   %9.0g                 Log settler mortality
. summarize logpgp95 avexpr logem4, sep(0)
    Variable |        Obs        Mean    Std. dev.       Min        Max
-------------+---------------------------------------------------------
    logpgp95 |         64    8.062237    1.043359   6.109248   10.21574
      avexpr |         64    6.515625    1.468647        3.5         10
      logem4 |         64    4.657031    1.257984   2.145931   7.986165
. * Partialing-out IV using plugin for lambda
. poivregress logpgp95 (avexpr=logem4), controls($xlist) selection(plugin, hom)
Estimating lasso for logpgp95 using plugin
Estimating lasso for avexpr using plugin
Estimating lasso for pred(avexpr) using plugin
Partialing-out IV linear model      Number of obs                =         64
                                    Number of controls           =         24
                                    Number of instruments        =          1
                                    Number of selected controls  =          5
                                    Number of selected instruments =        1
                                    Wald chi2(1)                 =       8.74
                                    Prob > chi2                  =     0.0031
-----------------------------------------------------------------------------
             |               Robust
    logpgp95 | Coefficient  std. err.      z    P>|z|     [95% conf. interval]
-------------+---------------------------------------------------------------
      avexpr |   .8798503   .2976286     2.96   0.003      .296509    1.463192
-----------------------------------------------------------------------------
Endogenous: avexpr
Note: Chi-squared test is a Wald test of the coefficients of the variables
      of interest jointly equal to zero. Lassos select controls for model
      estimation. Type lassoinfo to see number of selected variables in each
      lasso.
```

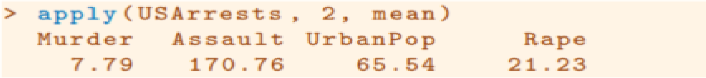
Lab 9. Unsupervised ML & PCA. Perform PCA on the USArrests data set for the 50 US states.



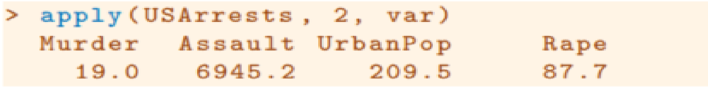
```
> states <- row.names(USArrests)
> states
```



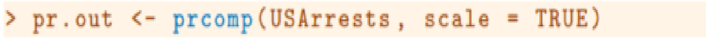
```
> names(USArrests)
[1] "Murder"   "Assault"  "UrbanPop" "Rape"
```



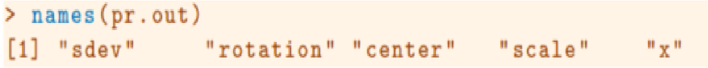
```
> apply(USArrests , 2, mean)
  Murder  Assault UrbanPop     Rape
    7.79   170.76    65.54    21.23
```



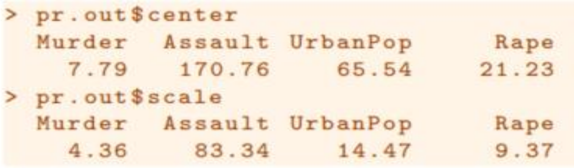
```
> apply(USArrests , 2, var)
  Murder  Assault UrbanPop     Rape
    19.0   6945.2    209.5     87.7
```



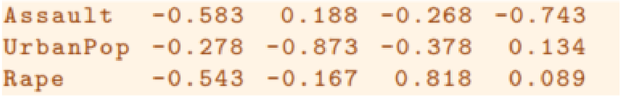
```
> pr.out <- prcomp(USArrests , scale = TRUE)
```



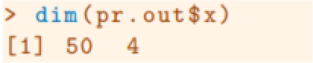
```
> names(pr.out)
[1] "sdev"     "rotation" "center"   "scale"    "x"
```



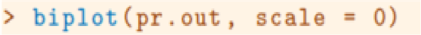
```
> pr.out$center
  Murder  Assault UrbanPop     Rape
    7.79   170.76    65.54    21.23
> pr.out$scale
  Murder  Assault UrbanPop     Rape
    4.36    83.34    14.47     9.37
```



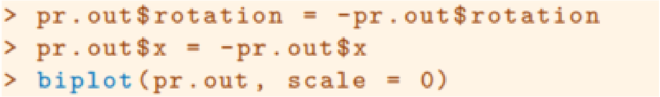
```
Assault   -0.583  0.188 -0.268 -0.743
UrbanPop  -0.278 -0.873 -0.378  0.134
Rape      -0.543 -0.167  0.818  0.089
```



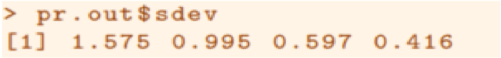
```
> dim(pr.out$x)
[1] 50  4
```



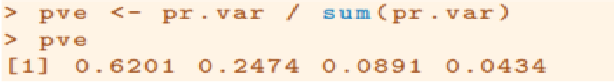
```
> biplot(pr.out, scale = 0)
```



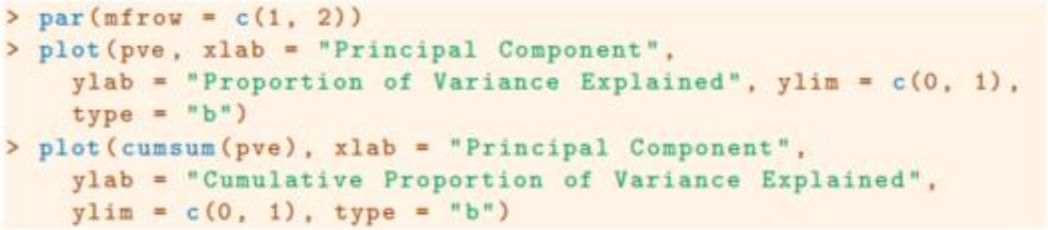
```
> pr.out$rotation = -pr.out$rotation
> pr.out$x = -pr.out$x
> biplot(pr.out, scale = 0)
```



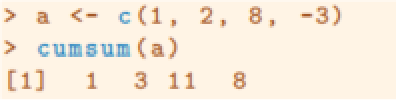
```
> pr.out$sdev
[1] 1.575 0.995 0.597 0.416
```



```
> pve <- pr.var / sum(pr.var)
> pve
[1] 0.6201 0.2474 0.0891 0.0434
```



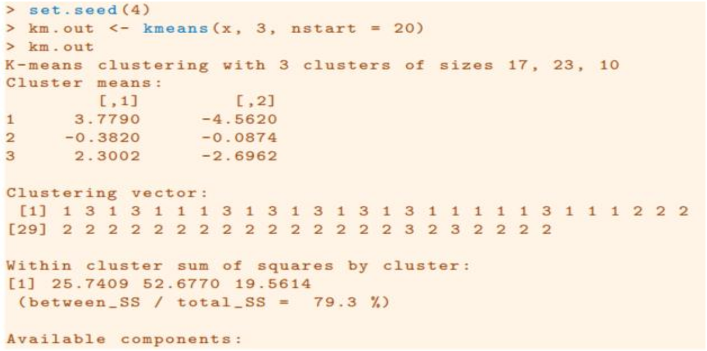
```
> par(mfrow = c(1, 2))
> plot(pve, xlab = "Principal Component",
    ylab = "Proportion of Variance Explained", ylim = c(0, 1),
    type = "b")
> plot(cumsum(pve), xlab = "Principal Component",
    ylab = "Cumulative Proportion of Variance Explained",
    ylim = c(0, 1), type = "b")
```



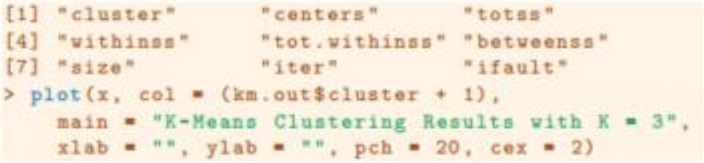
```
> a <- c(1, 2, 8, -3)
> cumsum(a)
[1]  1  3 11  8
```

Lab 10. Clustering

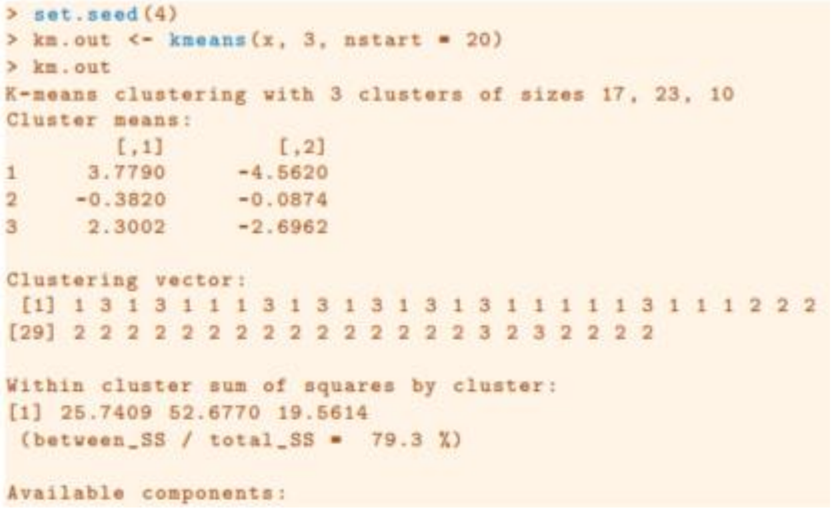
k-means with n=50 and k=3



```
> set.seed(4)
> km.out <- kmeans(x, 3, nstart = 20)
> km.out
K-means clustering with 3 clusters of sizes 17, 23, 10
Cluster means:
        [,1]        [,2]
1     3.7790     -4.5620
2    -0.3820     -0.0874
3     2.3002     -2.6962

Clustering vector:
 [1] 1 3 1 3 1 1 1 3 1 3 1 3 1 3 1 3 1 1 1 1 1 3 1 1 1 2 2 2
[29] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 2 3 2 2 2 2

Within cluster sum of squares by cluster:
[1] 25.7409 52.6770 19.5614
 (between_SS / total_SS =  79.3 %)

Available components:
```



```
[1] "cluster"      "centers"      "totss"
[4] "withinss"     "tot.withinss" "betweenss"
[7] "size"         "iter"         "ifault"
> plot(x, col = (km.out$cluster + 1),
    main = "K-Means Clustering Results with K = 3",
    xlab = "", ylab = "", pch = 20, cex = 2)
```



```
> set.seed(4)
> km.out <- kmeans(x, 3, nstart = 20)
> km.out
K-means clustering with 3 clusters of sizes 17, 23, 10
Cluster means:
       [,1]        [,2]
1    3.7790     -4.5620
2   -0.3820     -0.0874
3    2.3002     -2.6962

Clustering vector:
 [1] 1 3 1 3 1 1 1 3 1 3 1 3 1 3 1 3 1 1 1 1 1 3 1 1 1 2 2 2
[29] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 2 3 2 2 2 2

Within cluster sum of squares by cluster:
[1] 25.7409 52.6770 19.5614
 (between_SS / total_SS =  79.3 %)

Available components:
```

LAB LINEAR SHRINKAGE EXERCISES

Lab_x 1. choosing Validation set by CV.

a) Use hitters data set to select models of different sizes by *BIC, ajR*^{2}, etc. with Validation method.

b) Now select the best model by cross-validation method

Lab_x **2**. Comparison of Ridge, Lasso and elastic net.

a) Use the simulated data of lab 1 for y x1, x2 and x3 to run OLS, Lasso, Lasso adaptive, Lasso plugin, and elastic net with alpha =0 and 0.9.

b) Compare post-estimation coefficients.

Lab_x **3.** Regression under $p>N$ to select the best Lasso and adaLasso based on CV and BIC.

a) Run Stata vl to build the variable list, and split the sample into training and test subsamples and fit a linear Lasso.

b) Fit an adaptive Lasso

c) Select smoothing parameter λ by CV.

d) Select the best model by BIC

Lab_x 4. Fit a cross Partialing-out & double selection, see also above, Lab 7.

a) Use mus203mepsmedexp.dta to fit a cross-partial-out Lasso regression for *ltotexp* on *suppins*, controlling for all interaction variables generated from the original variables.

b) Fit a double selection Lasso

Lab_x 5. Manual application of Partialing-out Lasso IV, see also Lab 8 above.

Use Use mus228ajr.dta to fit *manually* a partialing-out Lasso IV model of lab 8 above.

Lab_x 6. Lasso, elastic-net

a) Use fakesurvey2_vl.dta to build var. list for Ridge regression.

b) Fit an elastic net model to auto.dta

Lab_x *7.* Partialing-out Lasso

Q-Use breathe.dta to measure the effect of nitrogen dioxide on the reaction time of school children by cross-fit partialing-out *xporegression*.

Lab_x 8. Clustering

Q-Apply clustering to the simulated data (x1.x2) with K = 2, 3 and 4 clusters identified.

Lab_x 9. Hierarchical Clustering

Q-Use Lab_x 9 data with n=50 to apply a hierarchical clustering

CHAPTER 20 Non-linear Machine Learning and Deep Neural Nets Models

20- *Nonlinear ML Models*. Econometrics employs non-linear models to deal with quite common types of data nonlinearities; the main types are when the response variable is discrete, or it is continuous but limited in scope, e.g. takes only positive values, or it is continuous with a nonlinear stepwise function. Machine learning offers techniques for modelling them when we have more predictors than the number of observations, similar to nonlinear models in econometrics that assume $p < N$ context. One solution is to employ shrinkage with regularization. However, the extension of linear ML regression to nonlinear models is challenging. In the linear ML models the least squares or likelihood methods are usually employed as the objective function modified with different regularizers such as Lasso, Ridge or AdaLasso. A common approach is to employ a nonlinear least squares or non-linear log likelihood objective function. That is a challenging task since constrained nonlinear optimization has to rely numerical solutions. If the log-likelihood objective function is concave or the least squares function is convex, or the class of models based on the Generalized Linear Model (GML) are applicable, then efficient algorithms are available for some regularizers, such as the convex Bridge and its special cases. The tuning parameter λ is critical for nonlinear LM models as for the linear ML applications. In the discrete response variable, the minimization of the least squares residual is not necessarily convex; the cross validation modified with shrinkage regularization based on likelihood objective. The identification of the optimal tuning parameter obtains a sequence of $(\lambda_1 > \lambda >_2 . . . > \lambda_T)$ each time excluding one-fold and calculate*s* the average coefficient estimates for each λ folder; it obtains the average ratio of the error to standard deviation across included folds, a measure known as the *deviance,* and select the best λ_t based on the deviance. Moreover, in the linear case, valid inference is achievable with PPE for the non-shrinkage parameters, the same approach is possible with nonlinear ML models, see Chan et.al. (2014, 2.2.3).

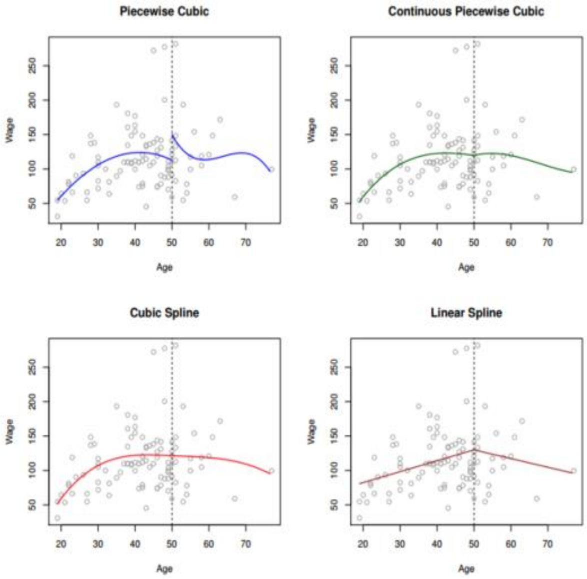
The focus of nonlinear ML models is on improving upon linear ML predictions by reducing further the complexity and thus the variance estimation of the ML linear models by extending them by a variety of polynomial regressions; these include spline and generalized additive regressions. P*olynomial regression* adds extra predictors to the original such as cubic regression with X, X^2, and X^3. *Step function regression* uses K different regions to generate a qualitative predictor to fit a piecewise constant regression. These models are special case of a *Basis function* approach that fits fixed known functions to X; for polynomials by $b_j(x_i) = x_i^j$ and for step functions by $b_j(x_i) = I(c_j \leq x_i < c_{j+1})$. *Regression S~~s~~pline* combine~~s~~ these two methods for more flexibility by fitting different polynomials to K different regions, while Smoothing Splines employ regression that improves complexity my minimizing a residual sum of squares subject to a smoothness penalty. Generalized additive models extend the above to multiple predictors.

20.1 *Spline Models*

Fig. 20.1 shows the fits for the subset of a US wage data set with a knot at age=50; cubic polynomials unconstrained (top right), constrained to be continuous at age=50 (bottom right), constrained to have continuous first and second derivatives (bottom left) and linear spline constrained to be continuous (bottom right). In this example we have only one knot, but in general there may be more than one. The top left looks unnatural with a jump and the constraints improve continuity of the fitted curve; each time imposing a constraint frees up one degree of freedom, so a degree-*d* spline is a piecewise polynomial of degree *d*. We use the basis model to represent a cubic regression spline with K knots by

$$y_i = \beta_0 + \beta_1 b_1(x_i) + \beta_2 b_2(x_i) + \ldots + \beta_K + 3b_K + 3(x_i) + \varepsilon_i$$

(20.1)



Plot 20.1 Polynomials for the wage dataWe can fit (20.1) by least squares with an appropriate choice of basis functions $b_1, b_2 \ldots, b_{K+3}$. In order to fit a cubic spline with K knots, we must

perform least squares regressions with an intercept plus 3+*K* predictors, and estimate a total of 4+*K* coefficients with *d*=4+*K*. However, splines can have high variance because the predictors are either too big or too small and we need to impose boundary constraints, suggesting to place more knots where the function might change rapidly and fewer where it seems stable, or equivalently pre-specify the degrees of freedom for the spline.

Smoothing spline provides a different approach to smoothness which is, to fit some specified function *g*(*x*) to the data to make $RSS = \sum_{i=1}^{n}(y_i - g(x_i))^2$ small. Without any constraints on *g*(*x*), we can always reduce RSS to zero, but that overfits the model. A natural alternative to overcome this problem is by adding a penalty term to RSS that minimizes

$$RSS = \sum_{i=1}^{n}(y_i - g(x_i))^2 + \lambda \int g''(t)^2 dt \tag{20.2}$$

Where λ is a nonnegative tu~~r~~ning parameter similar to those employed by Ridge and Lasso, the function *g* that minimizes (20.2) is known as a s*moothing Spline*. The first derivative of the penalty term $g'(t)$ measures the slop of a function at t. while its second derivative $g''(t)$ in (20.2) measures its *roughness*. A large value produces a very curvy plot near *t*, close to zero otherwise; $\int g''(t)^2$dt is a measure of total change in the function $g'(t)$. Therefore, the larger *λ*, the smoother *g* will be; the penalty term has no effect on (20.1.2) at *λ*=0 so the function will be jumpy but as *λ*→ ∞, *g* will be perfectly smooth; *λ* controls for the spline smoothness, and as *λ* increases from 0 to ∞ with *n* the nominal degrees of freedom, but with the spline parameters heavily constrained, the *effective degrees of freedom* df_λ goes from *n* to 2. We can choose an efficient value of *λ* by cross-validating RSS with the Leave-One-Out CV error (LOOCV), see chapter 12 for details.

20.2_Generalized Additive Models

An advanced automated non-linear method with greater flexibility is the *Generalized Additive Models* (**GAM**). The GAM with unspecified smooth, non-parametric functions f_j (.)'s, *have*~~s~~ ~~a~~ regression equation s aof the form

$$E(Y|X_{1,} X_2, \ldots X_p) = \alpha + f_1(X_1) + f_2(X_2) + \cdots + f_p(X_p) \tag{20.3}$$

We fit each function using a scatterplot smoother such as a kernel smoother with an algorithm for simultaneously estimating all *p* functions. In a classification context, the mean of the binary response in a two-class classification, $\mu(X) = P_r(Y = 1|X)$is related by a linear regression to the *logit* function:

$$\log\left(\frac{\mu(X)}{1-\mu(X)}\right) = \alpha + \beta_1 X_1 + \cdots + \beta_p X_p \tag{20.4}$$

This is modified for the *additive* logistic with a more general unspecified functional form *f*(.) for each linear term:

$$\log\left(\frac{\mu(X)}{1-\mu(X)}\right) = \alpha + f_1(X_1) + \cdots + f_p(X_p) \tag{20.5}$$

The additive logistic regression is an example of the GAM, relating the conditional mean response to an additive function via a link function *g*:

$$g[\mu(X)] = \alpha + f_1(X_1) + \cdots + f_p(X_p) \tag{20.6}$$

An example of the classical link function~~s~~ is the Gaussian response model. The identity link, $g(\mu) = \mu$ is used with a linear additive Gaussian response, $g(\mu) = logit(\mu)$ or $g(\mu) = probit(\mu)$ for binominal probabilities modeling with the latter as the inverse Gaussian cumulative function , $probit(\mu) = \Phi^{-1}(\mu)$ and additive log-linear models $g(\mu) = \log(\mu)$ for Poisson count data. Together with the gamma and negative-binominal distributions, they are all members of the exponential group of functions. However, not all functions are required to be exponential; the GAM can also handle a mixture of the following input terms functional forms: a semi-parametric model $g(\mu) = X^T\beta + \alpha_k + f(Z)$ with linear predictors for X, α_k for the *k*th qualitative input vector and the effect of Z input is modeled non-parametrically. $g(V,Z) = gk(Z)$ allowing interactive terms between Z and qualitative input vector, $g(\mu) = f(X) + gk(Z)$ and $g(\mu) = f(X) + g(Z,W)$ where the non-parametric function has two predictors, Z and W. In time series, the additive models can decompose the series trend and seasonal components.

Fitting Additive Models- The building block~~s~~ of the GAM is the scatter plot smoother. Here we employ the cubic smoothing spline which has the form

$$Y = \alpha + \sum_{j=1}^{p} f_j\left(X_j\right) + \varepsilon \tag{20.7}$$

We can specify penalty sum of squares terms with tuning parameters $\lambda_j \geq 0$, and apply the following backfitting algorithm with k iterations for additive models until $\hat{f}_j$ change less than a specified threshold.

1. Initialize: $\hat{\alpha} = \frac{1}{N}\sum_1^N y_i, \hat{f}_j \equiv 0, \nvdash i,$
2. Cycle: $j = 1,2,\ldots,p,\ldots.,1,2,\ldots.,p,\ldots,$

$$\hat{f}_j \leftarrow s_j[\{y_i - \hat{\alpha} - \sum_{k\neq j}\hat{f}_k(x_{ik})\}_1^N]$$

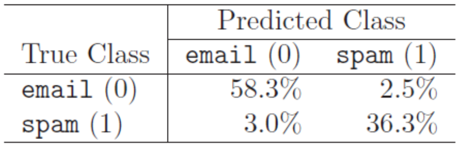
$$\hat{f}_j \leftarrow f_j - \frac{1}{N}\sum_{i=1}\hat{f}_j(x_{ij})$$

However, this does not produce a unique solution because the constant term α *is* unidentified (redefined by adding or subtracting constants). To overcome this problem, we make the standard assumption that the functions average to zero over the data. Then $\hat{\alpha} = ave(y_j)$ and provided the input matrix X entries has full column rank so it is a strictly convex function; minimization will then be unique. An iterative procedure can then solve for a minimization solution by the algorithm outlined above and known as *backfitting*. An example is the widely used logistic binary model; its generalized, additive logistic, has the form

$$\log\frac{P_r(Y=1|X)}{P_r(Y=0|X)} = \alpha + f_1(X_1)+\ldots+f_p(X_p)$$

This function is the most frequently employed model, especially in medical research, for example, in risk screening. This model can be estimated with the backfitting algorithm using the Newton iterative procedure. Another example is provided by Email Spam data from *ftp.ici.uci*, with a response coded 0 for *email* and coded 1 for *spam*, and 57 quantitative predictors, 48 predictors are defined as the percentage of words in the email matching a given word such as business, address, free, etc., and 6 as standing for the percentage of characters that match a given type, e.g. ch!. The remaining three predictors are based on sequences of uninterrupted capital letters: the average length, longest length, and the sum of the length. Using a randomly selected test set size of 1536, 30065 for the training observations, and employing a smoothing-spline tuning parameter (with the trace of its matrix 4=4) produced the outcome presented in table 20.1.

Table 20.1 GLM for heart data



	Predicted Class	
True Class	email (0)	spam (1)
email (0)	58.3%	2.5%
spam (1)	3.0%	36.3%

The detained estimates (not shown here) suggest many non-linear effects account for strong discontinuity at zero. Hence, replacing the predictors with an indicator for a zero count and applying a linear logistic model gives a test error of 7.4%, but including the linear frequencies reduces the error to 6.6%. Therefore, the nonlinear additive model has ~~an~~ additional predictive power.

20.3_Tree-based Methods

Tree-based methods segment the predictor space into a number of regions, and typically use the mean response training observation values in each region to which it belongs. Since the rules used for the predictor data segmentation are summed up in a tree, they are called *decision tree* methods and applicable to both regression and classification problems. Tree based methods, introduced by Breiman at. el. (1984), have hierarchical structure using a series of sequential division of the inputs *x* to reach a conclusion about *y* output, typically a prediction of *y*. They consist of two types of *Classification and Regression Trees* (**CART**); both have many similarities except for the output *y* being categorical for classification and numerical for regression, or qualitative v. quantitative. Trees are easy to visualize and understand without statistical interpretation and need little preprocessing to generate them.

20.3.1 *Single Tree Regression.*

Building a tree consists of a set of recursive, non-overlapping partitioning of the input data into *j* regions $R_1, R_2, \dots, R_j$ to obtain a prediction for *y* corresponding to all observations in each *j* region. The partitioning method is known as *Recursive Binary Splitting* that selects X_j with a threshold value *s* to divide the input space into regions such that $\{X| X_j < s\}$ and $\{X| X_j \geq s\}$ produce the largest fall in the prediction error, defined by the residual sum of squares (**RSS**) for numerical response, and the classification error rate (**CER**)- fraction of observations that do not belong to the most common class or category. Classification trees use two other error measures, the *Gini index* and *Cross Entropy* discussed below.

For a large set of variables, the processing tree-based approach leads to an overfitted model and the jump across different binary splitting points produces variances that differ in different regions, i.e. potentially heteroscedastic errors. A large tree may overfit and a small tree may miss important features of the data and lead to an inaccurate fit. Therefore, the size of the tree $|T|$, the number of regions that are not split any further ("leaves" of the tree), should be taken into account to limit overfitting. A measure of both tree size and accuracy is defined as

$$\mathbf{RSS+\alpha|\mathit{T}|}$$

Where **α** is a tun~~n~~ing parameter that penalizes the numerical regression model for additional splits**;** replaced with the Gini index for classification regression. The top of a tree for the first split is called the *root node*; a *branch* is a region resulting from ~~a~~ partitioning the input variable, and a node without any branch is the *terminal node* or a *leaf*. The process of adding more branches is known as *growing* the tree, while that of cutting the number of brunches as *pruning* the tree.

Let us use ∧ for the *and* operator, an indicator function $I(.)=1$ if its argument is true, zero otherwise, and define a response variable that depends on two covariates x_1 and x_2 as

$$y_i = \beta_1 I(x_{1i} < c_1 \wedge x_{2i} < c_2) + \beta_2 I(x_{1i} < c_1 \wedge x_{2i} \geq c_2) +$$

$$\beta_3 I(x_{1i} \geq c_1 \wedge x_{2i} < c_2) + \beta_4 I(x_{1i} < c_1 \wedge x_{2i} \geq c_2) + u_i$$

Fig. 2.1 shows the regression tree that partitions (x_1, x_2) into four regions. Within each region the response variable is the same for all values of x_1 & x_2 that belong to that region.

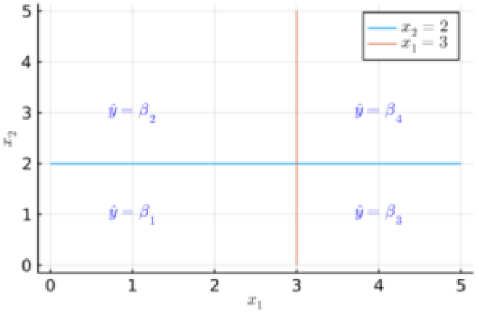


Fig. 2.1 Regression Tree Partition

For instance, with c_1=2 and c_2=3, the four branches are

$$\{(x_1, x_2): x_1 < 3 \wedge x_2 < 2\}$$

$$\{(x_1, x_2): x_1 < 3 \land x_2 \geq 2\}$$

$$\{(x_1, x_2): x_1 \geq 3 \land x_2 < 2\}$$

$$\{(x_1, x_2): x_1 \geq 3 \land x_2 \geq 2\}$$

The predicted response in regions in this example, given the corresponding β, is shown Fig. 20.2 tree.

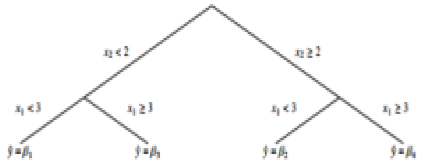
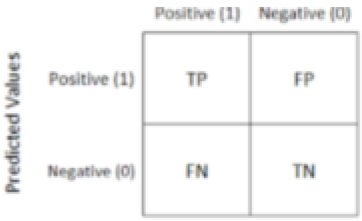


Fig. 20.2 Regression Tree View

We can also evaluate a binary classification tree by a *Confusion* matrix that compares each predicted value with its corresponding actual value, classifying all observations into four categories: the first two as True Positives (TP) and True Negatives (TN) if predictions are in line with the actual values; the remaining two as False Positives (FP) and False Negatives (FN) when the predictions contradict the actual values. Table 20.3.



Predicted Values		Positive (1)	Negative (0)
	Positive (1)	TP	FP
	Negative (0)	FN	TN

Table 20.3 Confusion Matrix

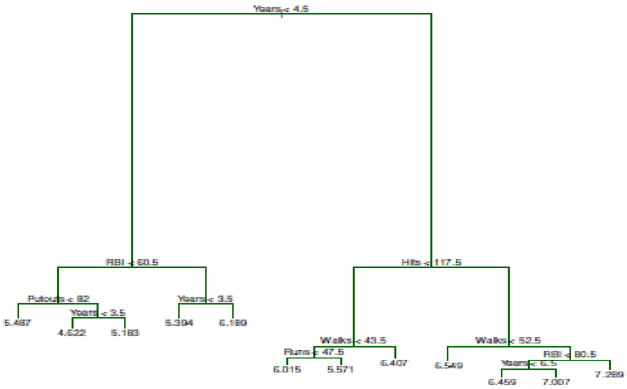
The Confusion Matrix allows comparison of different versions of the same model and comparison of different models such as logit and classification tree models.

At a given internal node (branch), the label splits the node into a left and right branch according to $X_j < t_k \& X_j \geq t_k$, and the tree is represented as down side up; each leaf is the mean response value of the observations that fall in that region. A regression tree divides the *p* predictor space $X_1, X_2, \dots, X_p$ into$R_1, R_2, \dots, R_j$ non-overlapping regions, for instance for *J*=3 and R_3= $\{X | Years \geq 4.5 \,, Hits \geq 117.5\}$ corresponds to $\overline{Y}_j$ for *J*=3 (three terminal node or leaves, with two internal nodes or branches). The decision tree starts at the top and recursively splits each internal node by a binary variable with the goal to minimize RSS:

$$\sum_{j=1}^{J} \sum_{i \in R_j} (y_i - \widehat{y_{Rj}})^2 \tag{20.8}$$

In doing this, the decision tree method minimizes the RSS at the level of the *curren*t split rather than looking at future splitting until RSS is minimized. This approach, called greedy decision tree, simplifies the impractical process of considering all possible trees and still produces a very large tree. Nonetheless, the greedy method can produce a complex tree, resulting in regression overfit and high variance, and an output that is difficult to interpret. On the other hand, building a small tree risks missing important non-linear features of the data. The approach taken in ML regression is to build a large tree first and then *prune* it back into a smaller *subtree*, the selection process known as *cost complexity pruning* or *weakest link pruning* that considers a sequence of trees based on a tuning parameter α each value of which corresponds to a subtree $T \subset T_0$.

For a better understanding of regression tree-based methods, consider a tree with m binary split regions and j splits to predict a quantitative response Y for a region-specific constant c_m and modeled as



Plot. 20.2 Full Tree for hitters data

$$f(x) = \sum_{m=1}^{M} c_m \mathrm{I}\,(x \in R_m) \tag{20.9}$$

Then the best $\hat{c}_m$ is just the average over m regions

$$\hat{C}_m = ave(y_i | x_i \in R_m) \tag{20.10}$$

Since finding the best binary partitioning in terms of min. sum of squares is computationally infeasible, we first proceed with a greedy tree. How large should we build the tree before pruning? Given *m* regions and *T*, let

$$N_m = \neq \{x_i \in \ R_m\}, \qquad (20.11)$$

$$\hat{C}_m = \frac{1}{N_m} \sum_{x_i \in R_m} y_{i,} \qquad (20.12)$$

$$Q_m(T) = \frac{1}{N_m} \sum_{x_i \in R_m} (y_i - \widehat{c_m})^2 \qquad (20.13)$$

We then define cost complexity pruning criterion by

$$C_\alpha(T) = \sum_{m=1}^{|T|} N_m Q_m\,(T) + \alpha|T| \qquad (20.14)$$

Where *T* is the number of terminal nodes. At α=0 we produce the full tree T_0 equal to the training sample error, but as α increases, branches get pruned from that tree in a nested manner with the value of α selected using *CV*. The approach to finding the best tree is to use the tuning α≥0 for each subtree $T_\alpha$to minimize $C_\alpha(T)$ with α=0 leading to the full tree T_0.

20.3.2-*Classification Single Tree.*

A decision tree for classification follows a similar method for qualitative response, predicting each training data observation belongs to the most commonly occurring class in the region to which it belongs. In this case, we are not only interested in prediction in a region but also the *class proportions* of the training data that fall in that region. We use a recursive binary splitting method to grow a classification tree but cannot use RSS as splitting criterion, instead we use the *classification error rate*: allocate an observation in a region to the *most commonly occurring class* of training data (true {0, 1} observations) in that region, then the fraction of the training observations in that region that do not belong to the most common class is the classification error rate.

Let the proportion of observations in node *m* ~~with~~ for region R_m; N_m number of observations be

$$\hat{p}_{mk} = \frac{1}{N_m} \sum_{x_i \in R_m} I(y_{i,} = k) \qquad (20.15)$$

Then the classification error rate is defined by

$$E = 1 - \max_{k} \ (\hat{p}_{mk}) \tag{20.16}$$

A measure of node impurity is a discrete misspecification error defined by

$$\text{Misclassification error: } \frac{1}{N_m} \sum_{x_i \in R_m} I(y_{i,} \neq k(m)) = 1 - \hat{p}_{mk(m)} \tag{20.17}$$

In practice, the classification error is not sufficiently sensitive to tree growing because of its discrete nature, and we rely on two other alternative continuous measures of node impurity. The *Gini* index is defined by

$$G \ = \sum_{k=1}^{K} \hat{p}_{mk} \ (1 - \hat{p}_{mk}) \tag{20.18}$$

It is a measure of total variance across K classes; the index has a small value for all class proportions are near 0 or 1, hence the Gini is regarded as a measure of *node purity*: a small value indicates a node mainly holds observations from a single class. The other index, known as *entropy*, also called *cross-entropy* or *deviance*, is defined as

$$D \ = - \sum_{k=1}^{K} \ \hat{p}_{mk} log \ \hat{p}_{mk} \tag{20.19}$$

As $0 \leq \hat{p}_{mk} \leq 1$ it follows that $0 \leq \hat{p}_{mk} log \hat{p}_{mk}$. If the proportions are all close to 0 or 1, then entropy and Gini have small values if the *m*th node is pure; they both produce similar values.

Fig. 20.3 shows a two-class plot with K=2 and ~~the~~ p as the proportion in the second class, cross-entropy scaled to pass through (0.5, 0.5). In this K=2 case, we have

$$1 - \max(p, 1-p) \, , 2p(1-p) \ and \ - plogp - (1-p)\log\,(1-p)$$

All three are similar but Gini and entropy are more suitable to optimization because of their differentiability.

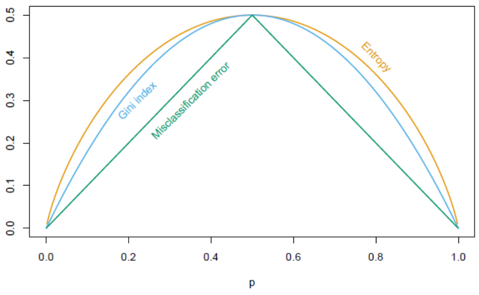
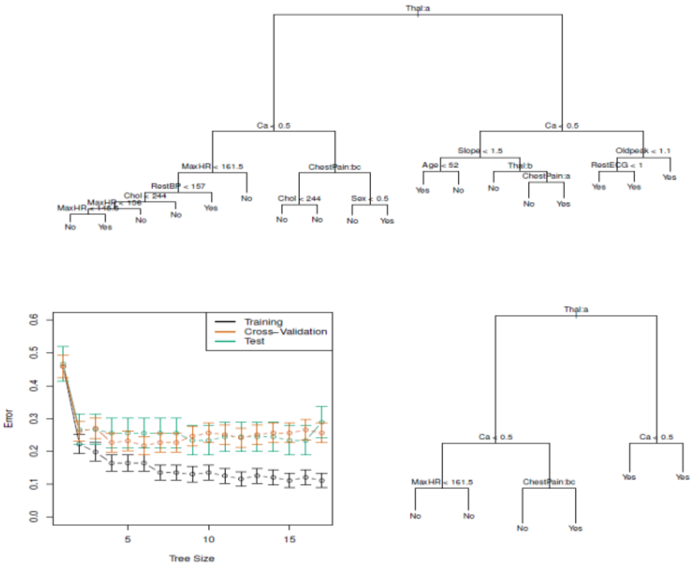


Fig. 20.3 Classification Error Rate

Example using the South African Heart attach data. The response here is coded as Yes indicates the presence of heart disease, *No* means no heart disease, 13 predictors include *Age*, *Sex* and *Chol*. level, and selection by CV leads to a tree with six terminal nodes. In this example we have both quantitative and qualitative, for instance *ChestPain*, predictors. Fig. 20.4 shows the full tree at the top, and pruned tree at the bottom right, and the CV, training and test errors for different sizes of the pruned trees. Note that for *RestECG* split (full tree bottom right), regardless of its value, the same response *Yes* is predicted and yet a split is undertaken because doing so leads to higher node purity (values close to either 0 or to 1). Although that does not reduce the classification error, it promotes node purity by the Gini and entropy indices that are sensitive to node purity.



Plot 20.4 Greedy and Pruned Trees, heart data set

20.2.3 Missing Data.

Missing data is a common problem with input data sets, and the solution depends on whether the missing variables have distorted the non-missing or are independent of the non-missing observations and randomly distributed, so the omission mechanism is independent of the unobserved value. More precisely, assume a y response vector and inputs $\boldsymbol{N}$ x p matrix of X with X_{obs} entry in X but also some missing values; let Z=(y, X), Z_{obs} =(y, X_{obs}) and $\boldsymbol{R}$ indicator matrix with ijth entry 1 if x_{ij} is missing, 0 otherwise. Then we say the data is *Missing At Random* (MAR) if $\boldsymbol{R}$ depends on the data Z only through Z_{obs}:

$$\Pr(R|Z, \theta)= \Pr(R|Z_{obs}, \theta) \qquad (20.20)$$

Where θ is any parameter in the distribution of $\boldsymbol{R}$. A stronger assumption more commonly employed for missing data is *Missing Completely At Random* (***MCAR***),

$$\Pr(R|Z, \theta)= \Pr(R|\theta) \qquad (20.21)$$

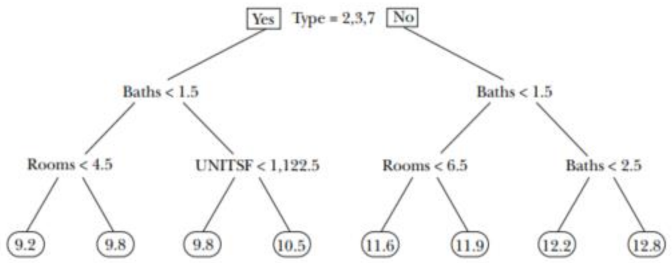
Most imputations rel~~a~~y on ***MCAR***. A procedure to identify the nature of the missing data is to code "missing" as an additional and then check to see if "missing" can predict the response. Given MCAR, there are some solutions. The tree-based models that contain many simple trees as their "building block", obtained by bootstrapping or another procedure, are known as *ensemble tree* methods. Ensemble tree approaches attempt to address the high variance of tree-based models. We discuss three such approaches, each offering a different solution for lowering tree-based variance. First, if the relative amount of missing data is small, disregard the missing data. Second, use ML methods like CART through surrogate splits. The *Surrogate V*ariables method uses only the observations on non-missing predictors; chooses the best primary predictors and split points, and then forms a list of surrogate predictors and splits as follows. The first surrogate predictor and its split point is that which best mimics the splits obtained on the training data by the primary split, the second surrogate is the second-best mimic in that regard, etc. So, if the primary split is missing as we send observations down the tree, we use the surrogate splits in order. The method of surrogate splits makes use of the correlations between predictors to correct for missing variables; the higher the correlation between missing predictor and other predictors, the smaller is the loss of information due to the missing value. Third, impute all missing values before training, as, for most learning methods, imputation is necessary, usually with the mean or median of the non-missing input data.

20.3.4-Comparison of the least squares and tree-based predictive models

Since the least squares regression already produces predictions, the value of employing a ML predictive method consists of the ability to deal with a very large number of predictors, hard to deal with using a linear regression. Mullianathan and Spiess (2016) consider a random sample of 10,000 owner-occupied units from the 2011 American Housing Surveys with values of each unit and 150 predictors such as size and location, number of rooms, etc.; using 41,808 units from the same sample as the hold-out subset. They report tree-based methods, esp. random forests, perform much better in prediction than the least squares.

The least squares application must make some choices whether to include all regressors, esp. dummies for location, green space, etc. which alone generates many more variables; we could also include interactions among the regressors since the number of rooms or their sizes depend on the unit's land area, etc. That word be infeasible if it leads to the number of regressors greater than the number of observations, at least very inefficient predictive values even if the sample size is less than the number of regressors. By contrast, ML searches for the interactions. A typical linear regression tree maps each vector of characteristics to a predicted value; the prediction function is in a tree form that divides into two parts at each node of the tree, for instance, two or less bathrooms (left) and more than two bathrooms (right), and continues down the list of regressors until reaching a terminal node, or a leaf, as a prediction. Fig.1 provides an example where each leaf corresponds to a product of several dummy variables ($x_1 = 1_{TYPE=2,3.7} \times 1_{BATH<1.5} \times 1_{ROOMS<4.5}$ for the left most leaf) with the corresponding coefficient of α_1= 9.2.

A Shallow Regression Tree Predicting House Values



Note: Based on a sample from the 2011 American Housing Survey metropolitan survey. House-value predictions are in log dollars.

Plot 20.5 Depth of a Tree

The fit of a tree can improve by making it deep[27], increasingly adding more branches to it until each observation would represent its own leaf and the fit would become perfect! This in fact would

[27] The depth of a tree is equal to its height- the length of its longest path from the root node to the terminal node, that is the number of nodes contain in its height- three in this example. A shallow tree has a lower depth, that is, using fewer variables because it uses fewer splits; a shallow tree may underfit the data, capturing too few of the data's critical features. A deep tree may overfit the data, picking on noise in the training data, not critical to its important features. Therefore, the depth of a tree in nonlinear ML corresponds to regularization to control for the number variables in linear ML such as Lasso.

be an overfit; overfitting is not just a feature of tree regression but affects all ML models. That flexibility to fit a variety of data structure implies that the best in-sample fit is a poor choice, the ML must rely on out-of-sample prediction. The overfitting problem is in part resolved by regularization: controlling the depth of the tree is an example of regularization. The smaller the tree, the worse will be the in-sample it, but that also means less overfitting, so ML imposes depth on the tree, and would choose the best among those of a certain depth. The other ML procedure is in using empirical *tuning* by fitting one part of the sample, and examine what level of depth produce the best out-of-sample result with the other part of the sample (CV). We can further improve efficiency by random folding of the sample and successively hold out one of the folds for out-of-sample evaluation using leaving-one-out CR. Then pick the parameter with the best estimated average performance. This procedure works well because it offers both the predicted and actual values of the response variable to assess the quality of prediction. While observability of the response values still leaves a large number of functions, regularization reduces this problem to a much lower dimension. By contrast, inference on parameter estimation typically relies on an assumption about the DGP, often normality, to produce consistent coefficient estimates.

20.3.5-Bagging model.

The main problem with tree models is that they are unstable; a small change in the training sample can lead to a drastic change in the tree structure, that in turn implies response prediction with a large variance and high degree of inaccuracy. The models discussed below offer solutions that can overcome the high variance of tree-based models.

The decision tree model has high variance because splitting the training data into two parts randomly results in very different outcomes. One solution is to split the tree nodes that produce similar outcomes and hence ensure low variance. We know linear models have low variance, so we can apply linear regression repeatedly to distinct data for a relatively large data set *N* compared to the number of predictors *p*, and then average the outcomes. Given *n* independent observations for Z with variance σ^2, obtain the mean of $\bar{Z} = \frac{\sigma}{n}$; that is averaging from distinct data repeatedly using bootstarp samples lowers high variance. This is the solution proposed by *bootstrap aggregation* or, the *Bagging M*ethod. We build a separate prediction model using each training set and average the predictions, that is, compute $\hat{f}^1, \hat{f}^2, \dots, \hat{f}^B$ using B separate training set, then average them to obtain a low variance learning model.

$$\hat{f}_{avg}(x) = \frac{1}{B}\sum_{b=1}^{B} \hat{f}^b(x) \qquad (20.22)$$

Since we do not have multiple training sets, we can train with B bootstrap

$$\hat{f}_{bag}(x) = \frac{1}{B}\sum_{b=1}^{B} \hat{f}^{*b}(x) \qquad (20.23)$$

Bagging hugely improves variance reduction (accuracy) because it *combines many thousands of trees* into a single process. The bagging model can extend to the classification by majority vote method. This method records the class predicted by each of the B trees and take a majority vote. The overall prediction is the most commonly occurring class among the B predictions. On average, each bagged tree around uses 2/3 of the observations for the training error calculation, and the remaining 1/3 left out, are called *Out-Of-Bag* (***OOB)*** set, B/3 trees used for testing; applying majority vote if bagging is for classification data. We note that with bagging, the simple tree structure of the model is lost; for interpretation this is a drawback; it is no longer clear which predictors are important if we employ a large number of trees, therefore bagging improves predictive accuracy at the expense of interpretability. More stable methods like nearest neighborhood are not influenced by this kind of drawback, however, bagging is most helpful with the unstable (high variance) tree-based models but then we lose the ability to interpret the estimation. Since each bagged tree is identically distributed, the expectation of an average of B such trees is the same as that of any of them. That is, the bias of bagged trees is the same as that of each bootstrap tree, so improvement comes through variance reduction for an unbiased estimate.

The relationship between bootstrap and Bayes methods establishes that the bootstrap mean is approximately the Bayesian posterior function, Bagging exploits this relationship to improve accuracy. Bagged estimate is an approximate posterior Bayesian mean, since the posterior mean minimizes squared error loss, this implies that bagging can reduce mean squared error.

20.3.6-*Random Forests*

The bagging process employs all *p* features without selecting any, as a result, the trees are highly correlated, reducing the scope for variance reduction. Random Forest (***RF***) is an alternative model of variance reduction based on the idea of *decorrelating* the bootstrap bagged trees by removing the bagging correlations through *de-coupling* them by randomly selecting $m< p$ features, thereby reducing tree correlation and hence, estimated reduced variance. Each time we consider a split, a random sample of *m* predictors is selected from the full set of *p* predictors, the split is allowed to use only ~~of~~ the m predictors. A fresh sample m is used at each split, typically for $m \approx \sqrt{p}$, for example for the Heart data above, 4 out of 13, so unlike bagging, RF does not even consider a majority vote procedure. To understand the rationale behind this approach, consider one very strong and a number of moderate predictors in the data set. Then the bagging collection will use the strong predictor at the top of each tree, making the trees similar to each other and highly correlated. By contrast, on average, $(p-m)/p$ of the RF splits will not consider this strong predictor, so other*s* in the set have a better set of being selected, resulting in less variable trees and hence, lower variance. Building an RF model with $m=p$ simply reproduces a bagged model.

An average of *B i.i.d.* random variables, each with variance σ^2, has variance (1/B) σ^2. If the variable are only *i.d.* but not independent, with positive pairwise correlation ρ, the average variance is

$$p\sigma^2 + \frac{1-p}{B}\sigma^2 \qquad (20.24)$$

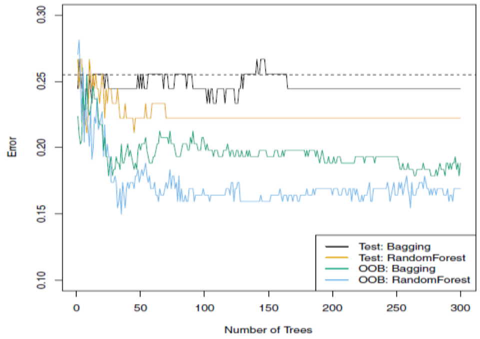
As *B* increases, the second term becomes negligible, but not the first, thereby limiting the benefits of averaging. The RF procedure reduces the correlation between trees through random selection of the input variables. Specifically, before each split, we select $m\leq p$ of the input variables at random used for splitting, usually for for $m \approx \sqrt{p}$, or even as small as 1. The RF regression prediction after B such trees $\{T(x;\theta_b)\}_1^B$ are grown, is

$$\hat{f}_r^B(x) = \frac{1}{B}\sum_{b=1}^{B} T(x;\theta_b) \qquad (20.25)$$

For regression, the predictors are averaged as above. For classification, RF employs a class vote from each tree, and then classifies based on majority vote, as in bagging. In addition, the application is recommended to use the following: $m= \lfloor \sqrt{p} \rfloor$ and minimum node size of one for classification, and $m=$

$\lfloor p/3 \rfloor$ and minimum node size of five for regression. For each observation (x_j, y_j), construct its RF predictor by averaging only those bootstrap trees for which that observation was *not* included. Note that when the input numbers are large with a small fraction of relevant variables, random forests will probably have poor results for small *m* because then at each split, the probability of selecting the relevant variables can be small.

Example For the Heart data set, Plot 20.6 compares the results for the test error rate by bagging and by RF as a function of the number of trees built with B bootstrap training data. The bagging test error is slightly smaller than that from a single tree and larger B does not lead to overfitting. However, RS test error (brown and blue curves) are clearly less than the ones for bagging, so the procedure here is almost identical to *One-Leave-Out* CV.



Plot 20.6 Bagging and RF test error rate

20.3.7 *Boosting*

Boosting is a non-bootstrap method that grow trees sequentially, each tree uses information from the previous tree to fit a tree on the modified version of the earlier step in the sequence, hence trees are grown not through random bootstrap but rather through adjustment to the current regression using the sequence of residuals by combining a number of trees, as in bagging. Boosting combines the outputs of many "weak" classifiers into a very effective "committee", its most popular two-class algorithm known as ***AdaBoost.M1***. In contract to building a very large tree which amounts to *fitting the data hard* with risk of overfitting, boosting *learns slowly* by fitting a tree with the residuals rather than the response Y, then adds this residual-based decision tree into the fitted function to update the next residual. Each tree is usually small with a few terminal nodes determined by the number of splits parameter *d* that slows the learning process, combined with a shrinkage parameter λ to even further slow~~ing~~ learning, but unique to this model, learning heavily depends on previous trees. Boosting has three tuning parameters: i) the number of trees B selected by CV, ii) the shrinkage parameter, typically with λ=0.01 or 0.001, iii) the complexity control parameter *d* for the number of splits in each tree, often *d*=1 is effective resulting in a *stump*, a tree with a single split.

For a two-class AdaBoost.M1problem, consider the output coded as Y€ (-1, 1) and a classifier G(x) for a vector of predictors X taking two values [-1, 1]. Boosting works through fitting a tree to the error of the training sample, and then using that to update the subsequent training error:

$$\overline{err} = \frac{1}{N}\sum_{i=1}^{N} I(y_i \neq G(x_b)) \qquad (20.26)$$

With the expected error rate in the future as $E_{XY}I(Y \neq G(X))$. Boosting uses a weak classifier, one only marginally better random guessing, repeatedly applied to modified versions of the data to obtain a sequence of weak classifiers G_m(x), with *m*=1, 2, . . . , *M*. The predictions from all of them are then combined through a weighted majority vote to obtain the final prediction by

$$G(x) = sign\ (\sum_{m=1}^{M} a_m G_m(x)) \qquad (20.27)$$

The Boosting algorithm computes $\alpha_1, \alpha_2, \dots, \alpha_M$ and weigh the contribution of each G_m(x) as demonstrated in

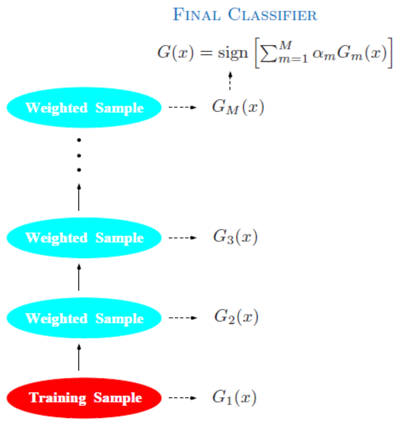
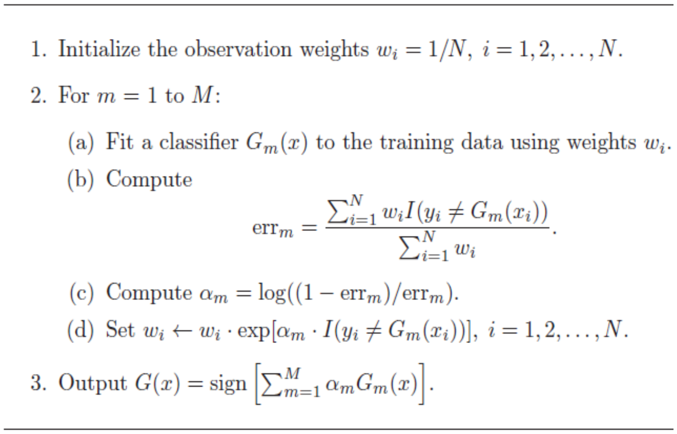


Fig. 20.4 Boosting Classifier

At each boosting step, weights $w_1, w_2, \dots, w_N$ are applied to each training observation (x_i, y_i), all the weights at the first step w_i = 1/*N* to train the classifier on the data, in each subsequent iteration *m*=1, 2, . . . , *M*, the weights are individually modified and the classification algorithm reapplied to the weighted observations. At each step *m*, a misclassified observation by the classifier from the previous step, that is, G_{m-1}(x) have their weights increased, while correctly classified observations have their weights decreased, thereby forcing boosting iteration to focus on the observations missed in earlier steps. Therefore, unlike Bagging and Random Forests, the boosting algorithm improves prediction by concentrating on the residuals, inducing them to move in the direction that reduces classification error rate as illustrated in the following algorithm

AdaBoost.M1 Algorithm

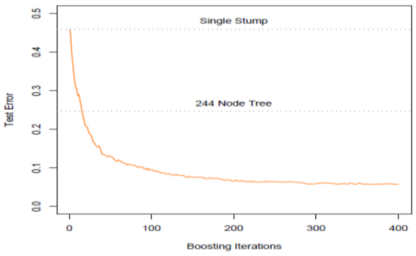


```
1. Initialize the observation weights $w_i = 1/N,\ i = 1, 2, \ldots, N$.
2. For $m = 1$ to $M$:
    (a) Fit a classifier $G_m(x)$ to the training data using weights $w_i$.
    (b) Compute
            $\text{err}_m = \frac{\sum_{i=1}^{N} w_i I(y_i \neq G_m(x_i))}{\sum_{i=1}^{N} w_i}$.
    (c) Compute $\alpha_m = \log((1 - \text{err}_m)/\text{err}_m)$.
    (d) Set $w_i \leftarrow w_i \cdot \exp[\alpha_m \cdot I(y_i \neq G_m(x_i))],\ i = 1, 2, \ldots, N$.
3. Output $G(x) = \text{sign}\left[\sum_{m=1}^{M} \alpha_m G_m(x)\right]$.
```

Here the individual weights are updated at the second step, and misclassified observations have their weights scaled by a factor of $exp(\alpha_m$) to increase their importance in the sequence for G_{m+1}(x). Boosting often improves the impact of a weak classifier very considerably as shown in Plot 20.7 for a simulated example test error rate with independent Gaussian features and the deterministic target Y using the median of a chi-squared random variable with ten degrees of freedom $x_{10}^2(0.5) = 9.35$ (sum of ten standard Gaussian squares) with stumps weak classifier defined as

$$Y = \{ \begin{matrix} 1 & if\ \sum_{j=1}^{10} X_j^2 \ > \ X_{10}^2(0.5) \\ -1 & otherwise \end{matrix}$$

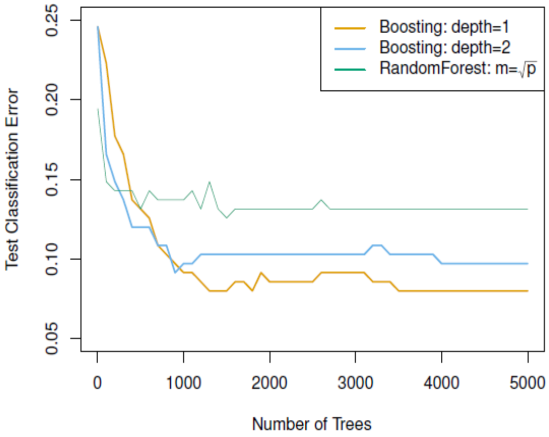
Applying this classifier on its own results in only a slight improvement of the test error of 45.5% compared to 50% by random guessing. However, as shown in Fig. 19.9.4, after 400 iterations, the boosting error rate slowly but sharply declines to reach a minimum of 5.8% despite having a weak classifier, with most of the dramatic impact coming from the base learner" G(x).



Plot 20.7 node classification tree and bootstrap stumps

Example: Plot 20.8 shown in boosting application to the 15-class gene expression data set cancer, displaying the test error for cancer v. non-cancer as a function of the total number of trees and the interaction depth *d*. In this case, using stumps for good enough, outperforming *d*=2 , and boosting *d*=1 or 2 outperform RF. The two boosted models have λ=0.01. The differences between the three are small, around0.02. the single tree error rate is 24%.

Plot 20.8- Boosting v. RF



20.2.7-*Bayesian Additive Regression Trees* (**BART**).

The BART is another ensemble methos related to both Bagging and Random Forests but differs from both in how its trees are generated.

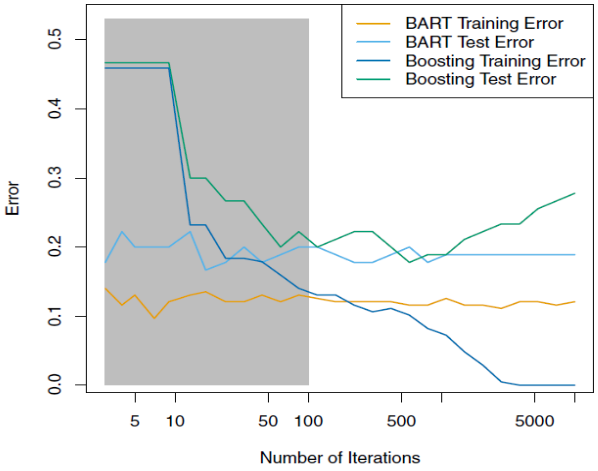
Let $\hat{f}_k^b(x)$ be the prediction at x for the *k*th regression tree used in the *b*th iteration, summing up the K trees a the end of each iteration for b=1, 2, . . . , B as $\hat{f}^b(x) = \sum_{k=1}^K \hat{f}_k^b(x)$. In the first iteration all trees are set to have a single root node with $\hat{f}_k^1(x) = \frac{1}{nK}\sum_{i=1}^n y_i$ hence $\hat{f}^b(x) = \sum_{k=1}^K \hat{f}_k^1(x) = \frac{1}{n}\sum_{i=1}^n y_i$. In subsequent BART updates each of the *k*th trees, in iteration *b*th, BART updates the kth tree by subtracting each response value from the predictions from all except the kth tree. This produces a *partial residual* rather than building a new tree by randomly modifying the tree from the previous iteration to improve the fit to the partial residual:

$$r_i = y_i - \sum_{k'<k} \hat{f}_{k'}^{b}\,(x_i) - \hat{f}_{k'}^{b-1}(x_i) \qquad (20.28)$$

We disregard the first few predictions known as the *burn-in* period. The important step in the BART algorithm is improving the fit by modification to the current partial residual of the tree from previous iterations rather than fitting a new tree by limiting how hard we fit the data. In effect, random modification of a tree to fit the residual is drawing a new tree from the posterior distribution.

Fig. 19.5.8 shows a BART application to the Heart data, with 100 in-burn iterations, K=200, B=1,000, and the number of in-burn period interactions L=100. BART performs well with minimal tuning.

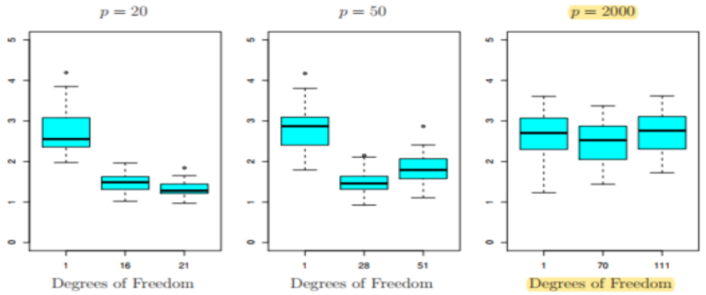
Plot 20.9 BART ensemble method genes data



20.4 *Models with* $p \geq N$.

An example $p \geq N$ is the technology of gene expression in biology, a p matrix of 2308 genes by N samples of 63 from a set of experiments. When the number of features is much larger than the number of observations, for example, in computational biology, overfitting and high variance become major estimation concerns and regularilization methods provide the key to resolve the high-dimensionality problem with $p \geq N$.

The Simulated example ~~of~~ in Plot 20.10 shows the problem with box-plots of relative test-error (test error $/\sigma^2$) over 100 simulations for *N*=100 using three values of the ridge regularization λ=0.001, 100, 1000 (from left to right). The average effective d.f. of the fit and the number of features are shown at the bottom and top of the plot. The average number of $t_j = \frac{\widehat{\beta}_j}{\widehat{se}_j}$ higher than 2 were 9.8, 1.2, and 0.0, At *p*=20 most of the significant coefficients are identified, at *p*=*N*=100 some of them are but at *p*=1000>*N* none of them are, even though many are zero. Even higher values of the tuning parameter λ do not help when *p*>*N*. There are two types of solutions for the case of $\boldsymbol{p} \geq \boldsymbol{N}$ either to modify the regularized procedure for *p*<*N*, or to employ a non-regularized one such as PCA.



Plot 20.10 PCA with Genes Data

We cannot apply LDA for $\boldsymbol{p} \geq \boldsymbol{N}$ but we can modify the procedure by imposing regularization on it. A common regularization is that features with each class are independent, that is, the within-cell covariance matrix is diagonal, with the diagonal covariance LAD as the discriminant for class k given by

$$\delta_k(x^*) = -\sum_{j=1}^{p} \frac{\left(x_j^* - \bar{x}_{kj}\right)^2}{s_j^2} + 2log\pi_k \qquad (20.29)$$

Where $x^* = (x_1^*, x_2^*, \dots, x_p^*)^T$ is a vector of expression for a test observation, $s_j$is the pooled with-class standard deviation of the *j*th gene, and $\bar{x}_{kj} = \sum_{i \in C_k} x_{ij}/N_k$ is the mean of N_k values of gene j in class forC_k set for class *k*. $\tilde{x}_k = (\tilde{x}_{k1}, \tilde{x}_{k2}, \dots \tilde{x}_{kp})^T$is called the centriod of class k. The drawback of the diagonal LDA is the employment of all of the features (genes) and must therefore be modified with $\boldsymbol{p} \geq \boldsymbol{N}$ applications so that the features that contribute negliligibly to the class predictions are excluded. This is done here by a procedure that shrinks the classwise mean toward the overall mean for each feature separately, called the *Nearest Shrunken Centeriods* (NSC) defined as

$$d_{kj} = \frac{\bar{x}_{kj}-\bar{x}_j}{m_k(s_j+s_0)} \tag{20.30}$$

Where $\bar{x}_j$ is the aggregate genes mean, $m_k^2 = \frac{1}{N_k} - \frac{1}{N}$, and s_0 is a small positive constant, typically the median value of s_j, to degrade large d_{kj} for expression values close to zero, shrinking them toward zero by a soft threshold

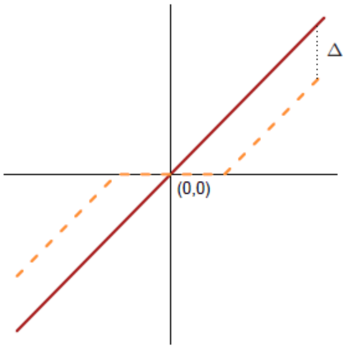
$$d'_{kj} = sign(d_{kj})(|d_{kj}| - \Delta)_+ \tag{20.31}$$

Where each d_{kj} is lowered by an amount of $|\Delta|$, set to zero if the value is less than zero, using a soft threshold function explaed in Fig. 20.4.

We note that the genes with nonzero d_{kj} contribute to the test error prediction, a very great majority of genes are just ignored. For example, in an application to the shrunken covariance LDA with soft threshold to the gene expression data set, 43 out 63 features (genes) were dropped out. This procedure can also be applied to a two-class classification when $\boldsymbol{p} \geq \boldsymbol{N}$. With K coefficient vectors of log-odds parameters, we regularize a symmetric multiclass logistic model by maximizing the penalty log-likelihood

$$\min_{(\beta_{0k}\beta_k)_1^K}[\sum_{i=1}^N logP_r(g_i|x_i) - \frac{\lambda}{2}\sum_{k=1}^K ||\beta_k||_2^2\,] \tag{20.32}$$

Fig. 20.4 Shrinking Predictors by NSC



The regularization automatically resolves the excess parameter problem with its binary outcome variable by forcing $\sum_{k=1}^K \hat{\beta}_{kj} = 0, j = 1, \dots, p$.

Regularized discriminant analysis (RDA) is an alternative procedure that employs the inversion of a very big p x p within-covariance singular matrix with largest rank of $N < p$. RDA resolves the singularity problem by applying regularization to the estimated within-covariance $\hat{\sum}$ shrinking it toward its diagonal

$$\hat{\Sigma}(\gamma) = \gamma\hat{\Sigma} +(1-\gamma)diag(\hat{\Sigma}), with\, \gamma \in [0,1] \qquad (20.33)$$

Here γ =0 corresponds to diagonal LDA, that is, the version of nearest shrunken centroids without shrinking. This procedure is like a Ridge regression that shrinks the total input covariance matrix towards a (scale) diagonal matrix, hence employing an ℓ_2 parameters penalty regularization of all nonzero coefficient estimates without features selection. We can also select $\ell_1$penalty regularization with a Lasso procedure that sets a portion of the coefficients exactly equal to zero by

$$\min_{\beta} \frac{1}{2} \sum_{i=1}^{N} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j))^2 + \lambda \sum_{j=1}^{p} |\beta_j| \qquad (20.34)$$

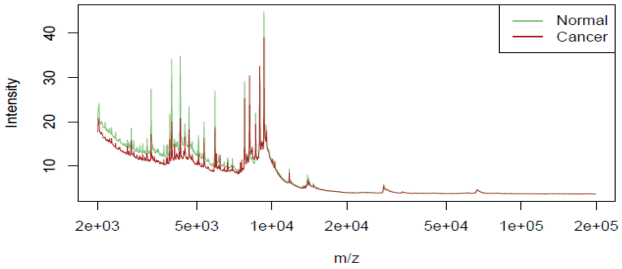
Lasso can also be applied to a two-class classification with the outcome as +, - 1, and zero cutoff point to the predictions. Lasso regularization is rather drastic. A compromise between L_2 and L_1penalties is provided by the elastic net penalty with its α parameter determining the mix of L_2 and L_1

$$\sum_{j=1}^{p} (\alpha\left|\beta_j\right| + (1-\alpha)\beta_j^2 \qquad (20.35)$$

Compared to Lasso, this procedure has the potential advantage of resulting in less drastic shrinkage, more non-zero coefficients with $p > N$.

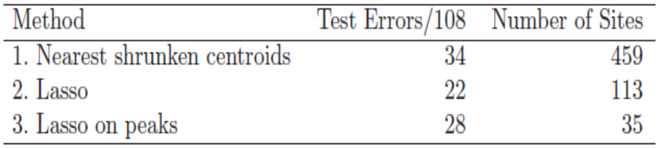
Example. Protein mass spectrometry, ~~is~~ a technology used for analyzing the protein in blood is an example of a Lasso application when $\boldsymbol{p \geq N}$**.**

Plot 20.11 shows an example with the average spectra for healthy individuals and those with prostate cancer. There are 16,898 number of *m/z*- the mass over charge ratios, ranging from 2000 to 40,000 for the sample size of 157 healthy and 167 cancer patients. The aim is to find m/z sites that can tell apart the two groups with the predictors as a function of *m/z*.



Plot 20.11-Test error for cancer data

A less smooth, harder Lasso fit achieves a noticeably small test error rate penalty. That may not be a helpful solution though since here the technology must discriminate peaks for the spectra for a sample of healthy and cancer patients. One way to resolve this issue is to apply a peak-extraction procedure to hierarchical clustering to the positions of these peaks along the *m/z* axis and cut the resulting dendrogram horizontally at height log (0.005)-assuming that peaks 5% apart are considered a part of the same cluster, and compute the peak positions in each cluster leading to 728 common clusters. The results are shown in table 20.4 for the number of test errors for 728 peaks and corresponding protein sites. Lasso hard fit does better, however, Lasso on 35 peaks is still more useful providing 35 sites for a follow-up study.



Method	Test Errors/108	Number of Sites
1. Nearest shrunken centroids	34	459
2. Lasso	22	113
3. Lasso on peaks	28	35

Table 20.4 Shrinkage with Protein data

20.5 *Relation of ML nonlinear Models to Nonlinear Econometrics and their Application*

The nonlinear ML regression and classification has some advatages compared to traditional nonlinear econometric models. Tree models are easy to interpret, ans easy to explore their partitioning ability, that is, to classify observations correctly using covariates, they can classify qualitative variables without having to specify dummies, and can estimate means and interaction effects without prior specification, and, finally, deal with hetroeskadasticity of non-constant variance. On the other hand, single trees have a tendancy to overfit, and have relatively low predictive accuracy. Importantly, trees do not produce any regression coefficient estimates, hence inference for quantitification of the relationship bewteen the response and covariates is problematic, although the non-linear ML tree-based response prediction does not face the same problem. Another issue is computational time intensity of the nonlinear tree-base models as the number of covariates increase, although parallel Boosting, Bagging and Random Forests models can be employed to reduce computational time.

There are two issues that connect ML tree models ~~and~~ to nonlinear econometric models, namely nonparametric regression and threshold regression. Tree models can be seen as the simple form of a nonparametric model

$$y_i = m(x_i) + \varepsilon_i \qquad (20.36)$$

where $m(x_i) = E[\frac{y_i}{x_i}]$ is the conditional mean and ε_i are random errors; m assumes no particlar parametric function, and estimated at particular local values of x. A common method is to aveage all y values or some local window of x, By sliding that window along the domain of x one can estimate the entire regression function over the size of the window, also known as the bandwidth; further improvements can be achieved by using kernel functions that empoly weights based on the distance of the observations wihin a bandwidth from the particular value of x under consideration, see chapter 11. Given a suitably selected kernel and some adjustment to the bandwidth, the nonparametric econometric models can replace CART. Trees can also be seen as step functions, a piece-wise constant functions with the step functions approximating splits (tree branch nodes); however, the covariates must be processed prior to nonparametric application, for example, by transforiming categorical variables to a set of dummy *va*riables. However, nonparametric and threshold econometric models have a rich theoratical basis for rate of convergance and their asymptotic properties that allow inference and causal analysis. This confines the ML nonlinear models with little theoretical basis to response prediction applications by side-stepping inference.

20.6 *Inference with tree-based models*

For a small tree, it is possible to visualize the role of each variable in tems of its partitioning ability that corresponds to the importance or significance of that variable. The ML nonlinear *importance score* becomes harder to obtain as the number of variables increase or if Bagging and/or Random Forests are employed. Still, the imprtance/significance of a variable can be determined by quantifying the increase in regression *RSS*, or classification *Gini* if that variable was excluded from the tree, by repeating this process for B trees and averaging the increse in *RSS* or *Gini* as the variable’s importance score. This process is repeated for all the variables to obtain an importance score. A large score indicates that the variable is important and should not be excluded. Fig. 20.12 shows an example of an importantce score plot.

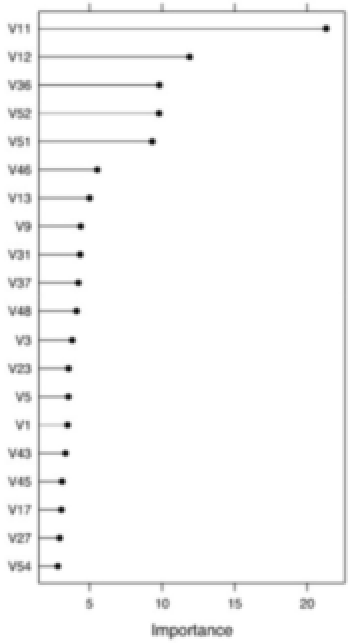


Fig. 20.12 Importance score with 54 predictors

Where variable V11 is the most imprtant, implying that its removal from the tree(s) results in the largest average increase in the Gini index; the second largest is V12, etc. The importance varaible score keeps the largest ones with the largest values and excludes those with low values. However, there is no theoretical basis to the score plot and that makes its use of limited value in econometrics, particularly since coefficient test statistics ate missing.

Even with a single binary treatment, we can obtain the conditional av~~a~~erage treatment effect $y(x)$ by

$$Y\,(x)=E[y|d=1,\, x]\text{-}\ E[y|d=0,\, x] \tag{20.37}$$

Rather than partitioning based on minimizing RSS or Gini, Athey and IMbens (2016) propose to choose a variable left and right split that maximizes the squared difference between the estimated treatment effects by

$$\sum_{left}(\,\bar{y}_1-\bar{y}_0)^2+\sum_{right}(\,\bar{y}_1-\bar{y}_0)^2 \tag{20.38}$$

Where the bar above the variables indicates the mean of the observations, providing a causal interpreation of the treatment effects. However, treatment models constitute a relatively ~~a~~ small part of econometric models, and it remains to be seen how generalizable this proposed approach. It is fair to say that non-linear ML models are mainly for prediction, and are still in early stages of development for inference compared to the linear ML inference examined in chapter 19.

20.7: Deep learning Networks

ML neural networks are nonlinear models that extract linear combinations of inputs and then model the response as a nonlinear function of the derived features. A network (or graph) is a pair of *V*= (1, . . . *n*) set of nodes (intersections; angles) and *E* set of edges between them presented by an *N*-dimensional adjacency matrix (*V* × *E*) that shows the closeness of two corresponding nodes, usually measured by the Euclidian distance between them. Networks are graphs that represent economic activities, while neural networks are techniques used to identify hidden patterns in the data; networks are often used as inputs into neural-based ML models including deep network, convolutional and recurrent networks. Neural Networks (*NN*) extract~~s~~ a linear combinations of inputs and transform them into non-linear functions of the response. The *NN* models are classfied into a single hidden layer *Shallow NN*, also called *Valina NN*, and multiple hidden layers *Deep NN*; the most common Shallow *NN* is the *Feed Forward NN*. Deep learning has the neutral-network as its core regression with two special cases: *Recurrent Neual Net* (*RNN*) with sequential data and time-series, and *Convolutional Neual Net* (*CNN*) for classification, usually applied to image classification based on Euclidean distances; examples are economic closeness in migration or international trade. However, we start with a simpler non-NN regression known as *Projection Pursuit Regression* (**PPR**).

The *PPR* is a deep learning non-linear regression used to specify the direction of variables along a line through the origi~~o~~n using *unit vectors*.[28] Let ω_m for *m*=1, 2, . . ., M be the unit *p*-vectors of unknown parameters. Using those, the PPR is an addative model of the form

$$f(X) = \sum_{m=1}^{M} g_m(\omega_m^T X) \tag{20.39}$$

[28] A unit vector *v* is defined to have a magnitude of 1; *magnitude* of a unit vector *v*=(a, b) is ||v||=$\sqrt{a^2} + b^2$=1, so *v*=(1, 3)≠1 is not a unit vector while *v*=(0, 1) is; unit *v* has the same direction, same angle formed with the horizontal X-axis, of the vector *v* but a magnitude of 1. The division of u=v/||v|| results in the unit-vector v.

However, the model uses derived features$V_m = \omega_m^1 X$rather than the oroginal features, the function g_m is unspecified and estimated, together with ω_m directions. We wish to estimate ω_m by a suitably flexible smooth regression so that the model "projection pursuit" ~~or~~ fits well; $g_m = \omega_m^1 X$ is called a Ridge function in $\mathbb{R}^p$. PPR is a general model that can form many nonlinear functions of linear combinations, for example, the product $X_1 . X_2$can be alternatively presented additively by PPR as $[\ (X_1 + X_2)^2 - \ (X_1 - X_2)^2]/4$.

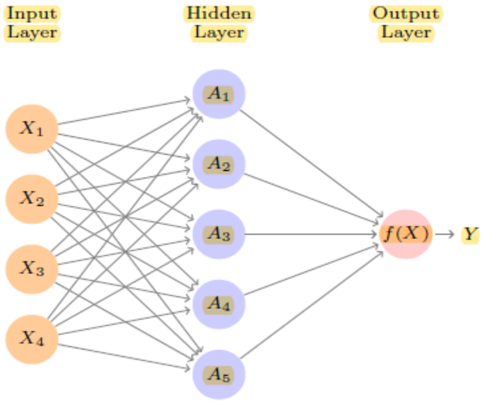
The fitted PPR is hard to interpret because the inputs enter the model nonlinearly e.g. as interactions, so it is usually used for prediction purposes, except when *M*=1, called the *single index* model in econometrics. PPR approximately minimizes the error function by

$$\sum_{i=1}^{N} [y_i - \sum_{m=1}^{M} g_m(\omega_m^T x_i]^2 \qquad (20.40)$$

We impose complexity restrictions on g_m to prevent overfitting. The number of *M* is part of the PPR model building estimation; cross-validation is used to determine *M*; additional steps are required when the next term fails to improve the fit noticeably.

Single Hidden Layer Neural Net. A neu~~t~~ral network employs a vector of predictors *p* for *X*=(X_1, $X_2, \dots, X_p$) to construct a non-linear function *f*(*X*) in order to predict the response variable *y*; the approach differs from the machine learning tree-based on non-linear regressions of Boosting, Random Forests in the model structure. Fig. 20.1 explains the simplest vanila neu~~t~~ral net with a single hidden layer, B*ack-Propagation Net*, known as the Feed-F*orward Neural Network*. The plot on the right shows a quantitative response function and that on the left a qualitative classification response function.

Fig 20.13 Single Layer NN



The two figures present a very similar model *structure* though with a different response variable. On the left we have p=4 predictors or *features* that make up the *input layer*; each p feeding the *K* pre-selected hidden units obtained from linear combination of the input layer. The unobservable hidden layer units are then also combined linearly and used to predict the response function. The model is of the following form

$$f(X) = \beta_0 + \sum_{k=1}^{K} \beta_k h_k(X) = \beta_0 + \sum_{k=1}^{K} \beta_k g\,(\omega_{k0} + \sum_{j=1}^{p} \omega_{kj}\, X_j) \tag{20.41}$$

Model construction is a two-step model process. First, we compute the *K activations*$A_k$as a function of input features; $A_k$is a different transformation of the original X_1, . . . ,X_p

$$A_k = h_k(X) = g(\omega_{k0} + \sum_{j=1}^{p} \omega_{kj}\, X_j) \tag{20.42}$$

Where g(z) is the non-linear pre-specified *activation function;* the *K*=5 activations are then feed into the output layer leading to the linear regression

$$f(X) = \beta_0 + \sum_{k=1}^{K} \beta_k\, A_k \tag{20.43}$$

The model must provide estimates from data of all parameters $\beta_0, \dots \beta_K$ and $\omega_{10}, \dots, \omega_{Kp}$. The common activation functions employed are sigmoid (also employed in logistic functions to convert a linear function to 0-1 probabilities. Early neu~~t~~ral net applications favored the sigmoid activation function, also used in logistic regression to transform a linear function into 0-1 probabilities.

$$g(z) = \frac{e^x}{1+e^z} = \frac{1}{1+e^{-z}} \tag{20.44}$$

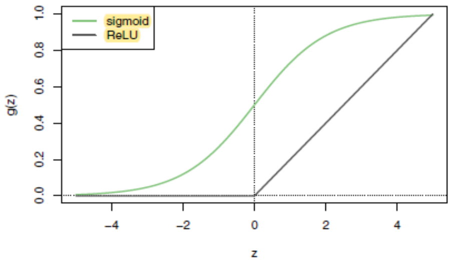
However, the same outcome can be obtained without the sigmoid activation function by the radial-basis function~~s~~ (RBF) defined as

Radial Basis: $S(x) = exp(x^{-2})$ (20.45)

The modern preference is for the *Rectified Linear Unit* (**ReLU**) which thresholds at zero and allows more efficient computation, Fig. 20.2 shows the piece-wise linear ReLU behavior.

$$g(z) = (z)_+ = \{ \begin{matrix} 0 \;\; if\; z < 0 \\ z \;\; otherwise \end{matrix} \} \tag{20.48}$$

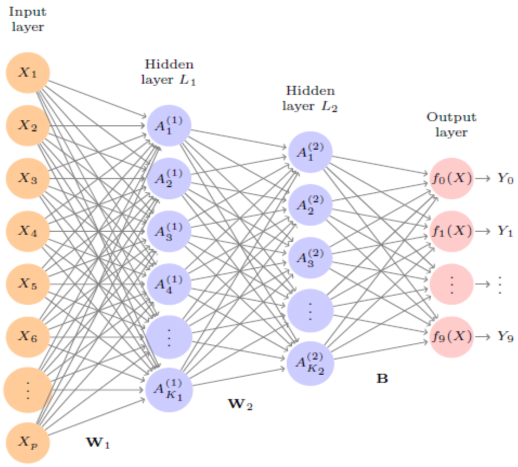
Fig. 20.14 sigmoid v. ReLu functions



Without the non-linear activation function, the model is reduced to a simple linear one; however, non-linearity captures complex interaction patterns.

20.8 ***Deep Learning- M****ultiple Hidden Layers Neural Net*. Modern Neu~~t~~ral Net regression employs more than one hidden layer; usually many units per layer. Fig. 20.15 presents a multilayer network with two hidden layers L_1&$L_2$and ten output variables representing a single qualitative variable.

Fig. 20.15 Multilayer NN



The first hidden layer with an activation function as

$$A_k^{(1)} = h_k^{(1)}(X) = g(\omega_{k0}^{(1)} + \textstyle\sum_{j=1}^{p} \omega_{kj}^{(1)} X_j) \qquad (20.49)$$

For k=1, 2, . . . ,K_1. The second hidden layer takes the $A_k^{(1)}$ activations. The second hidden layer takes the $A_k^{(1)}$activations as input to compute the new activations for ℓ=1. 2. . . $K_2$with

$$A_k^{(2)} = h_k^{(2)}(X) = g(\omega_{k0}^{(2)} + \textstyle\sum_{k=1}^{K_1} \omega_{\iota k}^{(2)} A_k^{(1)}) \qquad (20.50)$$

The process makes the second layer a function of X via the first layer activations$A_t^{(2)} = h_t^{(2)}(X)$; and the network builds complex transformations of X as features fed into the output layer. The final step is to compute ten different linear models

$$Z_m = \beta_{m0} + \sum_{\ell=1}^{K_2} \beta_{m\ell} h_\ell^{(2)}(X) = \beta_{m0} + \sum_{\ell=1}^{K_2} \beta_{m\ell} A_\ell^{(2)} A_\ell^{(2)}, \qquad (20.51)$$

for m=0, 1. . . . ,9. Define $f_m(X) = P_r(Y = m|X)$ as class probabilities and use the special *Softmax* activation function that secures ten non-negative probabilities that sum up to one.

$$f_m(X) = P_r(Y = m|X) = \frac{e^{Z_m}}{\sum_\ell^9 e^{Z_\ell}} \qquad (20.52)$$

More generally, the derived featuresZ_m are generated from the activations as a function of the transformation function for Z_m hidden units by

$$Z_m = \sigma(\alpha_{0m} + \alpha_m^T X), m = 1, \dots, M$$

$$T_k = \beta_{0k} + \beta_k^T Z, \qquad k = 1, \dots., K,$$

$$f_k(X) = gk(T), \qquad k = 1, \dots K,$$

Where $Z = (Z_{1,} Z_2, \dots, Z_M), and\ T = (T_1, T_2, \dots T_K)$. We note that with the transformation function σ(v) as the identity function, the entire model reduces to linearity in inputs, making it clear that the neu~~t~~ral network model is a nonlinear generalization of the linear model by the transformation function. We also note that the single layer network is identical to the PPR model; though the latter employs a nonparametric $g_m(v)$ form compared to the simpler networks with three free parameters σ(v). Hence, using the neu~~t~~ral network model to express PPR leads to

$$g_m(\omega_m^T X) = \beta_m \sigma(\alpha_{om} + \alpha_m^T X) = \beta_m \sigma(\alpha_{0m} + \| \alpha_{0m} \| (\omega_m^T X)) \qquad (20.53)$$

Where $\omega_m = \alpha_m / \| \alpha_m \|$ is the mth unit vector; since $\sigma_{\beta.a0.s}(U) = \beta_\sigma(\alpha_0 + sv)$ has lower complexity than g(v), the network~~s~~ model uses many more activation functions than the PPR.

We use the identity $g_k^{(T)} = T_k$ for the regression transformation while for classification, it is more common to use the *Softmax* function; the same transformation is also used with the multilogit model.

$$gk(T) = \frac{e^{T_k}}{\sum_{\ell=1}^{K} e^{T_\ell}} \qquad (20.54)$$

With the qualitative responses, we obtain coefficient estimates that minimize the negative multivariate log-likelihood called the *cross-entropy*.

$$-\sum_{i=1}^{n}\sum_{m=0}^{9} yim \log\left(f_m\,(x_i)\right) \qquad (20.55)$$

We would have minimized squared error loss function if the responses were quantitative. Typically, there are many times more parameters to estimate in neural networks than the number of observations; to avoid overfitting some regularization must be imposed on the network regression models.

Example 1-Simulated Data

Generate data from $Y = f(X) + \varepsilon$ for two additive Networks models.

Sum of Sigmoids: $Y = \sigma(\alpha_1^T X) + \sigma(\alpha_2^T X) + \varepsilon_1$ (20.56)

Redial: $Y = \prod_{m=1}^{10} \emptyset(X_m) + \varepsilon_2$ (20.57)

Where $X^T = (X_1, X_2, \ldots, X_p)$; p=2 with $\alpha_1 = (3, 3)$and $\alpha_2 = (3, -3)$in the first model and p=10 for the radial model. Both models have Gaussian errors, with variance chosen so that the signal-to-noise ratio

$$\frac{Var(E(\frac{Y}{X})}{Var(Y - E\left(\frac{Y}{X}\right))} = \frac{Var(f(X))}{Var(\varepsilon)}$$

is 4 in both models. We fit networks with weight decay and different numbers of hidden units to obtain the average test error $E_{T_{est}}(Y - f(X))^2$for each of the 10 random starting weights. Fig. 20.4 shows that the neu~~t~~ral network (on the left) works well with the Sigmoid model, and the two-unit model has the best performance, achieving an error close to the Bayes rate (the regression error variance). We note that with more hidden units, overfitting quickly becomes a problem and with some starting weights, the model does worse than the linear one. The Radial function (on the right) does not do well; it has an error greater than the Bayes error. Fig.20.5 repeats the exercise for the Sigmoid model with no weight decay and the Radial model with weight decay (λ=0.1); the former results in severe overfitting, while the latter produces good outcomes at all number of hidden units with no evidence of overfitting. In short, there are two free parameters to select: weight decay λ and the number of hidden units.

Fig. 20.16 Nonlinear Transformation functions

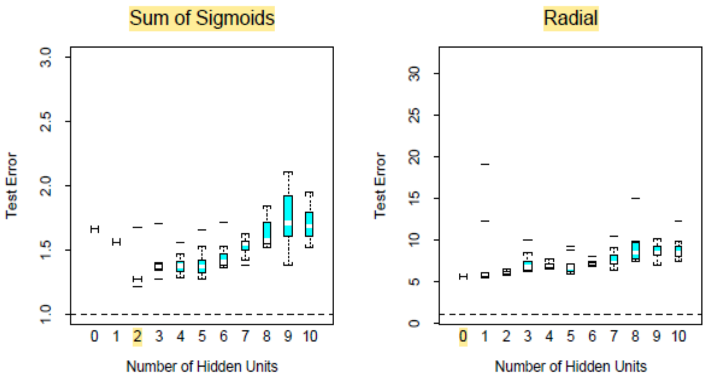
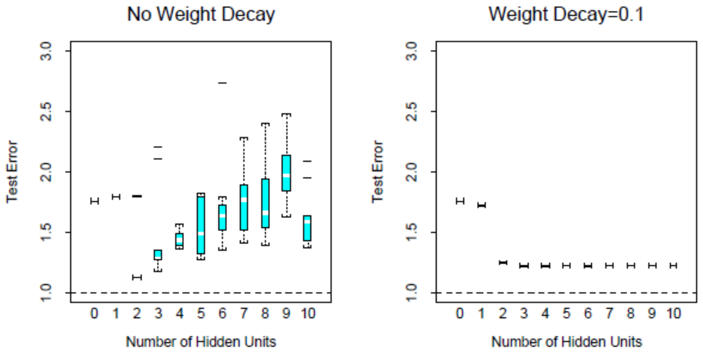


Table **20.17** Declining Weight function



Example: The Handwritten Digit Recognition MNIST is a famous data set MNIST handwritten-digit prediction problem. shown in Fig. 20.3 displays a small segment of the MNIST data. We have two hidden layers; and K_1=256 and K_2=128 hidden units, and the output layer with 10 units. This network has 235,146 parameters or *weights*, including the intercepts called *biases*.The data for this comes from the handwritten ZIP codes on the envelopes from the US Postal mail. Each image is a segment from a five-digit ZIP code, isolating a single digit. The images are 16 x 16 eight-bit grayscale maps, with each pixel ranging from 0 to 255. Plot 20.12 displays some sample images. The goal is to predict from the 16 x 16 matrix of pixel intensities the identity of each image (0, 1, . . . , 9) accurately, a classification problem that requires a low error rate to prevent misdirection of mail. In this study, there are 320 digits in the training set, 60 in the test set.

Plot 20.12 Section of Handwritten Digital MNIST

20.9 *Fitting a Neu~~t~~ral Network*-

Since *NN* models must minimize a non-linear function, whose curvature has often multiple minimization solutions; obtaining a global solution that dominates local ones faces a non-convexity problem; two strategies usually are employed as a solution. The first, called *slow learning*, uses the slow iterative gradient descent method for the second derivative of the minimum. function and stops the iteration when the prediction error stops decrease:

$$\underset{(w_k)_1^K,\beta}{\text{minimize}} \frac{1}{2}\sum_{i=1}^{n}(y_i - f(x_i))^2 \qquad (20.58)$$

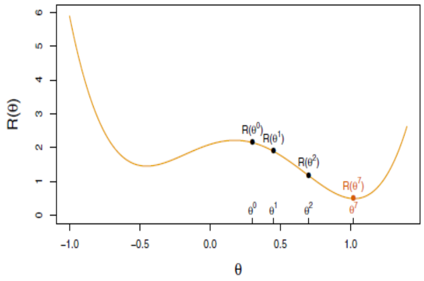
Where $f(x_i) = \beta_0 + \sum_{k=1}^{K}\beta_k g(w_{k0} + \sum_{j=1}^{p} w_{kj}x_{ij})$.

Plot 20.13 explains that procedure visually with two minimum solutions of the objective function *R*(θ) with θ = -0.46 for local and θ=1.02 for global. At each step, θ moves downhill against the gradient, setting *t* ← *t*-1until it cannot go down any further; in this example the global solution is reached in 7 steps.

We solve by the chain rule of differentiation that shows a fraction of the residual assigned to each parameter (via chain rule)- a process known as *NN Backpropagation*. However, the slow learning method involves many steps and with a large *n* number of observations; we should work with a small minibatch of *n* observations each time to compute a gradient step, a process known as S*tochastic Gradient Descent* (*SGD*), the current deep learning state-of-the-art minimization method.

The second procedure is to regularize the minimization by imposing Ridge or Lasso penalties on the parameters. For example, for regularization, essential to avoid overfitting, we can use the earlier MNIST data as an example to augment the minimization function with a Ridge penalty term

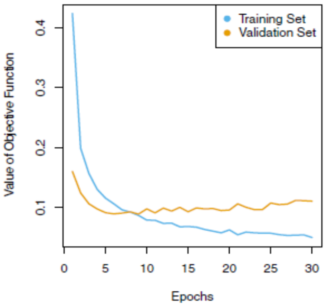
Plot 20.13 local v. global monimization



$$R(\theta;\, \lambda) = -\sum_{i=1}^{n}\sum_{m=0}^{9} y_{im} \log\left(f_m\,(x_i)\right) + \lambda\sum_j \theta_j^2 \,(20.59)$$

using different tuning λ values for different group layer weights as shown for the MNSIT network in Fig. 20.12 for training and validation errors as a function of training epochs for the log-likelihood objective.

Plot 20.14 Validation error



For *classification NN*, we work with the *Softmax* activation function and the Cross-*Entropy Error* function, this neutral net is identical to a linear logistic regression with *n* hidden units; estimating all parameters by maximum likelihood. A solution that minimizes R(θ) globally is likely to be an overfit, hence we either regularize directly by adding a penalty term, or indirectly by adopting an early stopping rule. The general rule to the minimization of R(θ) is by the G*radient Descent* method, derived using the chain rule of differentiation. The solution involves two processes. In the *Forward Pass*, the current weights are fixed and the predicted values are$\hat{f}_k(x_i)$ are computed; in the *Backward Pass*, the δ_k activation functions are computed and Backpropagated to obtain s_{mi}, and then use both errors to compute the gradient by iteration (unclear). This two-pass process is called the D*elta Rule*, each hidden unit passes and receives information only from units with a shared connection; we have a *Training Epoch* if the updating is replaced with one sweep through the entire training sample. Back-propagation is simple with local fit but can be very slow due its minimization involving the computation of its second-order solutions.

We should also note the following issues in implementation:

Starting values for weights should be chosen randomly close to zero, and then the model will become more non-linear as the weights increase.

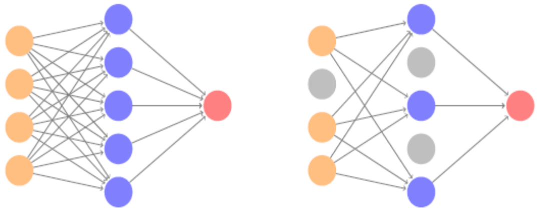
Too many weights overfit the data for the global minimum of R. *Weight decay* is a more explicit regularization similar to the Ridge for the linear method that adds a penalty to the error function $R(\theta)+\lambda J(\theta)$ with tuning parameter $\lambda \geq 0$, where

$$J(\theta) = \sum_{k,m} \beta_{km}^2 + \sum_{m,\ell} \alpha_{m\ell}^2 \tag{20.60}$$

Use cross-validation to choose λ; larger values of λ shrink the weights toward zero as in the Ridge regression. At the outset standardize all features to have mean zero and standard deviation one, and finally, remove the large effects of input scale on the final result. In general, it is better to have too many hidden layers than too few so as to avoid missing important non-linear complexities; we can always reduce weight toward zero if necessary. With multiple minima, it is better to average predictions over the networks instead of using weights averaging since then the model nonlinearity suggests the averaged solution might have a disappointing result.

Dropout Learning Network is a relatively new and efficient regularization similar to Ridge and inspired by the Random Forests model in randomly removing a *fraction* ϕ units in a layer when fitting the model, separately each time a training observation is processed. The remaining units substitute the missing observations and their weights are scaled up by a factor of 1/(1 -ϕ) for compensation as a kind of regularization. Fig.20.18 (right) displays dropout network; gray nodes are randomly selected and disregarded in training set.

Fig. 20.17 Dropout network



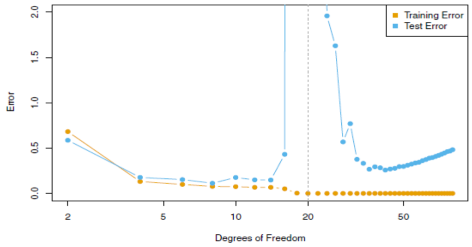
Lea*r*ning methods tend to work best for intermediate levels of the bias-variance trade-off complexity, suggesting interpolation to get low training error results in high test error. A method that interpolates the training data well by making the training model less complex is known as D*ouble Descent*, it simulates the model from

$$Y = \sin(X) + \varepsilon$$

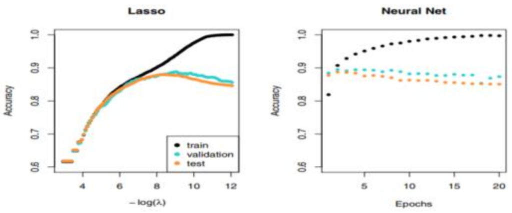
Where $X \sim U|-5,5|$ (uniform distribution), and $\varepsilon \backsim N(0, \sigma^2)$ with $\sigma = 0.3$.

An application of Double Descent for *n*=20 is shown in Fig. 20.14, when the degree of freedom is *d*=*n,* interpolation threshold is zero and remains zero thereafter; the test statistic increases sharply at the threshold but descends to an acceptable level before a final rise.

Plot 20.15 training and test d.f.



20.10 *Document Classification* is an important kind of ML application for predicting attributes of documents, for example investors' attitudes from a large set of text comments. Internet Movie Database or IMDb includes short review movie critiques; the response variable in this case is the *positive* or *negative sentiment* of the reviewers. Each review can be of a different length, including nonwords, spelling errors, slang, etc. Prediction requires finding the features of a document, or *featurizing* it; the *Bag-of-Words* model is simplest for this purpose. The *Bag-of-Words* scores each document for the presence or absence of each of the words in a language dictionary, usually English. We limit the dictionary to *M* words, that means for each document, we generate a binary feature vector of length *M*, and score 1 for every word present, zero otherwise. We use 10,000 most frequently occurring words in the sample of 25,000 reviews, resulting in a training feature matrix of 25,000 × 1000, though only 1.30% of the binary entries are nonzero, most are zero: that is, the matrix is *sparse*. For model tuning, we split a validation set of 2,000 from the training observations of 25,000, and fit two model sequences: a Lasso logistic and a two-class Neural network, with two hidden layers; each layer with 16 ReLU hidden units. The outcome appears in Plot 20.16, showing both have a tendency to overfit and reach roughly the same test accuracy.



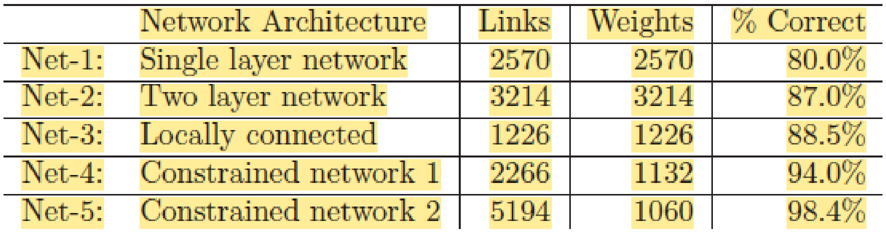
Plot 20.16 IMDb *Bag-of-Words* review

The m *Bag-of-Words* model summarizes a documents by the presence/absence of a word regardless of its context; There are two ways to take context into account; the first treats each document as a sequence of words preceding and following them, the second uses the *Bag-of-Gram* model, for instance *i*th *n*=2, we have a bag of 2-grams with consecutive co-occurrence of every distinct pair~~s~~ of word such as convincingly effective as a positive sentiment and convincingly ineffective as a negative sentiment. We shall return to an alternative ML analysis of this example by RNN below.

20.10 *Convolutional Neutral Net* - CNN

Table 20.5 shows the result of five networks fitted to this data; all networks have Sigmoidal output units fit with the sum-of-squared error function: Net-1 has no hidden layer; it starts overfitting quickly. Net-2 is a single hidden layer vanilla model with 12 hidden units, Net-3, 4 and 5 have each hidden units connected to only a batch of units in the layer below; here locally extracting features from the layer below significantly reduces the number of weights. In this example, Net-5 does the best with errors of 1.6% compared to the vanilla Net-2 of 13%. Sharing the same set of nine weights forces the extracted features in the different parts of the image to be computed by the *same linear functional*, therefore, such networks are named as C*onvolutional Networks*.

Table 20.5 NN with different hidden layers



	Network Architecture	Links	Weights	% Correct
Net-1:	Single layer network	2570	2570	80.0%
Net-2:	Two layer network	3214	3214	87.0%
Net-3:	Locally connected	1226	1226	88.5%
Net-4:	Constrained network 1	2266	1132	94.0%
Net-5:	Constrained network 2	5194	1060	98.4%

The Deep networks models have two specializations, roughly spatial data changing over space, and sequential data changing over time or by relative positions, for example, word positions in a text; the latter is called the C*onvolutional Neural Network* (*CNN)* and the former *Recurrent Neural Network* (*RNN*). The architecture of a CNN (its inputs or neuron, number of layers and weights used in the regression) is best understood through an image processing application; the method is employed in areas such as image recognition and are increasingly used for time-series forecasting. The CNN classification identifies specific features in the image to separate each specific class of objects. CNN works with two types of hidden layers of data. The CNN filter builds up a *hierarchical* process by combining hidden *Convolutional Layers*, searching for small patterns; and CNN *Pooling Layers* down sample the patterns into prominent subsets. A convolution operation boils down to repeatedly multiplying matrix elements and then adding the results; a simple 4 x 3 image example is

$$\text{Original Image} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \\ j & k & l \end{bmatrix}$$

Impose a 2 x 2 filter on the image

$$\text{Convolution Filter} = \begin{bmatrix} \alpha & \beta \\ \gamma & \delta \end{bmatrix}$$

Then apply convolution to the image with the filter to obtain

$$\text{Convolved Image} = \begin{bmatrix} a\alpha + b\beta + d\gamma + e\delta & b\alpha + c\beta + e\gamma + f\delta \\ d\alpha + e\beta + g\gamma + h\delta & e\alpha + f\beta + h\gamma + i\delta \\ g\alpha + h\beta + j\gamma + k\delta & h\alpha + i\beta + k\gamma + l\delta \end{bmatrix}$$

Note that each 2 × 2 image submatrix is multiplied by the 2 × 2 filter matrix to produce each element of the convolved matrix. For example, multiplication of the first top left image matrix by the filter leads to the first right element of the convolved matrix; that of the last 2 × 2 image submatrix (bottom right) with the filter leads to the last element (left) of the convolved matrix. This process condenses the nearby local features. Hence, the convoluted image gives prominence to the small patterns that look like the convoluted filters. Note that the CNN filters are not predefined; CNN *learns* the filters for the classification problem at hand.

CNN pooling layers summarizes a large image into a smaller one by condensing it. One way to do this is by M*ax Pooling Process* that sums up each 2 x 2 block of pixels in an image using maximum value in the block to reduce its size by a factor of two in each direction as explained in the following example,

$$\text{Max pool} \begin{bmatrix} 1 & 2 & 5 & 3 \\ 3 & 0 & 1 & 2 \\ 2 & 1 & 3 & 4 \\ 1 & 1 & 2 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 3 & 5 \\ 2 & 4 \end{bmatrix}$$

The two main blocks of the CNN are the *Feature Extraction* and *Prediction* blocks. The *P*rediction *B*lock is a *Feed F*o*rward* deep NN examined above, and the elements of the Extraction Block are convolutional layers, on-linear transformation of the data pooling layers for dimension reduction, and a fully connected (deep) *Feed F*o*rward* NN; elements are combined together in a sequence of layers:

convolution + nonlinear transformation → pooling → convolution + nonlinear transformation → pooling → . . . → Fully-connected (deep) *NN*

A computer image is a matrix of pixels each element of which represents the intensity of the pixels and its dimension is the resolution of the image with color as the third dimension, hence the image is a three-dimensional matrix; an image kernel is a small matrix employed to filter effects such as blurring and sharpness. Kernels are used for feature extraction which select~~s~~ the most important portions of an image. We refer to this process as *convolution*. We explained different aspects of this process below by a number of plots.

Fig. 20.18 is a (3 × 3) filter ~~is~~ applied to a (6 × 6) producing an output (4 × 4) matrix. Each element of the output matrix is the sum of the product between the input matrix entries and the weight (filter) matrix as explained in the note. Note that the shaded red (3 × 3) input matrix slides to the right and down by one entry.

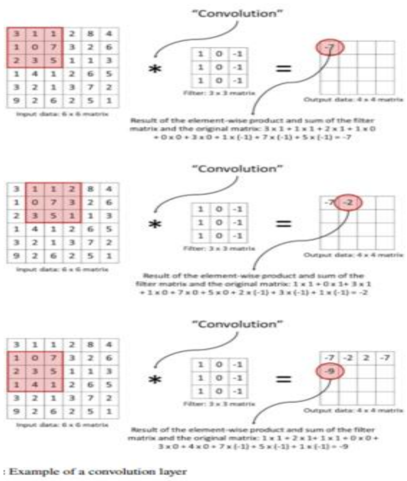

"Convolution"
Filter: 3 x 3 matrix
Input data: 6 x 6 matrix
Output data: 4 x 4 matrix
Result of the element-wise product and sum of the filter matrix and the original matrix: 3 x 1 + 1 x 1 + 2 x 1 + 1 x 0 + 0 x 0 + 3 x 0 + 1 x (-1) + 7 x (-1) + 5 x (-1) = -7
Result of the element-wise product and sum of the filter matrix and the original matrix: 1 x 1 + 0 x 1+ 3 x 1 + 1 x 0 + 7 x 0 + 5 x 0 + 2 x (-1) + 3 x (-1) + 1 x (-1) = -2
Result of the element-wise product and sum of the filter matrix and the original matrix: 1 x 1 + 2 x 1+ 1 x 1 + 0 x 0 + 3 x 0 + 4 x 0 + 7 x (-1) + 5 x (-1) + 1 x (-1) = -9
: Example of a convolution layer

Fig. 20.18 Convolutional Filter

We note also that we can reduce the output matrix dimension by the S*tride* technique, that is, by sliding the shaded red matrix by more than one input matrix entry in order to reduce the output dimension shown in F~~r~~ig. 20.19

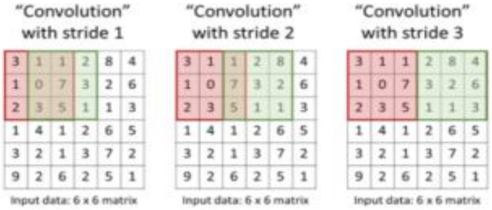

"Convolution" with stride 1
"Convolution" with stride 2
"Convolution" with stride 3
Input data: 6 x 6 matrix
Input data: 6 x 6 matrix
Input data: 6 x 6 matrix

Fig. 20.19 convolutional layer with stride

Due to the border effect, the convolutional matrix has smaller dimensions than the input matrix. We can resolve this problem by filling a border with zeros using the padding technique as shown in Fig. 20.20

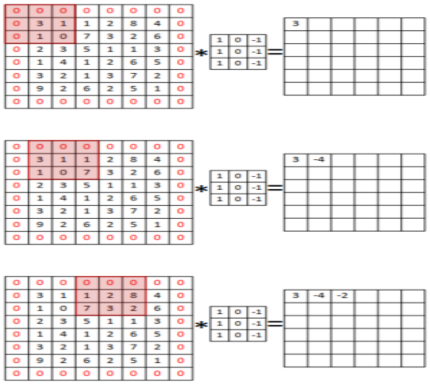


Fig. 20.20 convolutional layer with padding

Each convolutional layer may have more than one convolutional filter; Fig. 20.21 shows three (6 × 6) matrices with two filters, and the output matrix is also a set of two (4 × 4) matrices.

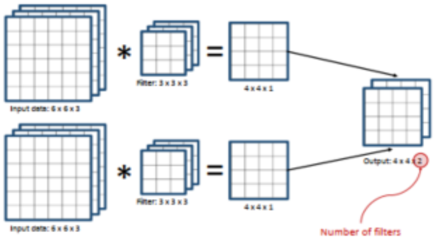


Fig. 20.21 convolutional layer with two convolutional filters.

The outputs of the convolutional layer go through the activation function nonlinear transformation, for example, by ReLU, as in Fig. 20.22

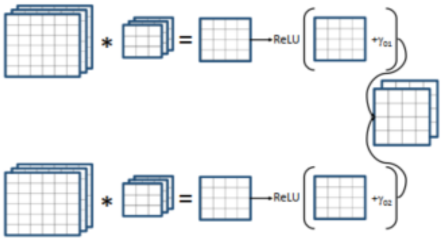


Fig. 20.22 convolutional layers with nonlinear transformation

In the final step, we apply the method for dimension reduction, a common pooling method employed is the *Max Pooling* whereby the final output is the maximum entry in a sub-matrix of the convolutional layer output as explained in Fig. 20.23

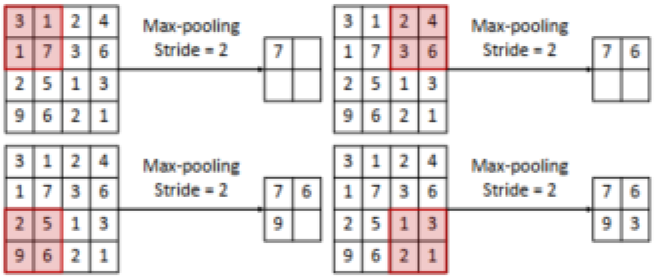


Fig. 20.23 nonlinear transformation with two convolutional filters

The above process is repeated as many times as the number of convolutional layers in the networks. The user has to define the hyperparameters for 1) number of convolutional layers (*C*), 2) number of pooling layers (*p*), 3) number (K_c) and dimension (height Q_c, width R_c; depth S_c), and 4) architect of the deep NN. The parameters to be estimated are 1) Filter weights: $W_{ic} \in \mathbb{R}^{Q_c \times R_c \times S_c}$, i=1, …, K_c, and *c*=1, . . ., *C.* 2) ReLU biases $\gamma_c \in \mathbb{R}^{K_c}$, *c*=1, . . ., *C.* 3) all the parameters of the fully connected deep network.

20.11*Recurrent Neutral Net* (*RNN*)-

RNNs are Networks that allow for feedback among the hidden layers because they use intenal (memory) to process sequences of inputs. A generic RNN is presented by

$$H_t = f(H_{t-1}, X_t) \tag{20.61}$$

$$\hat{Y}_{t+h|t} = g(H_t) \tag{20.62}$$

Where $\hat{Y}_{t+h|t}$ is the prediction of Y_{t+h} given t, f and g are to be defined as H_t and is a K-dimensional hidden state. As a time-series approach, the RNN is a type of state-space econometric model we discussed in chapter 15 here examined in the context of *p* > *N*. RNN can remembers the order by which inputs go through the hidden state and can sequence the data so each sample is dependent on previous ones as is common in time-series analysis. However, the RNN solutions are vulnerable to a small or divergent gradient problem. To overcome this problem, we employ a version of RNN called the *Long-Short Term Memory* (**LSTM**) network. Fig. 20.24 illustrates the architecture of a typical LSTM network where red circles indicate logistic activation, and blue circles hyperbolic tangent activation, and symbols × and + stand for multiplication and summation operations. The RNN consists of the cell state, and the forget, input, and output gates. The cell state introduces some memory to the LSTM to remember the past, LSTM retains only the relevant information for making *predictions*, the forget cell informs LSTM what information to throw away, and the output gate provides the activation for the final output at time t. The input and output gates have the same structure and filter the information from the previous time period as well as the new input. The prediction is a linear combination of hidden states ns

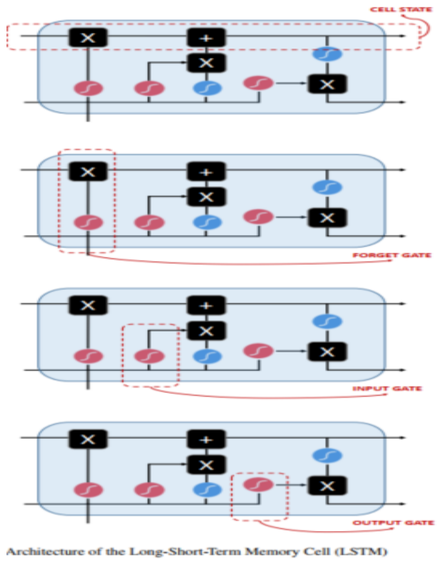
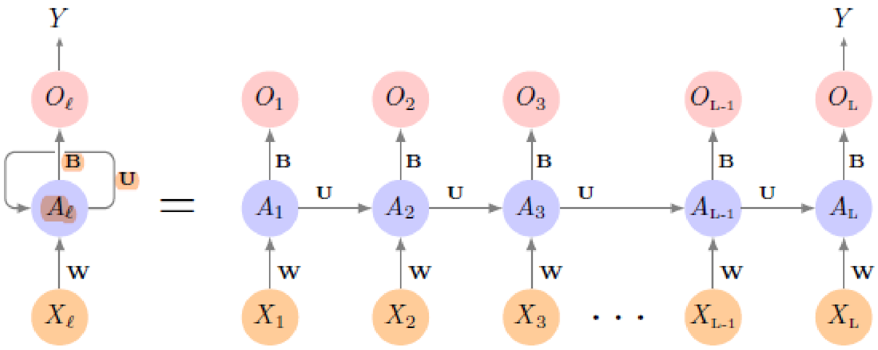


Fig. 20.24 LSTM Architecture

RNN is designed for data that are sequential in nature, using forward and backward information to construct networks, e.g. lag values in a time-series, or relative positions of words in a document, as explained in Fig. 20.25

Fig. 20.25 Recurrent NN


Y
O_ℓ
B
U
A_ℓ
W
X_ℓ
=
O_1 O_2 O_3 O_{L-1} O_L
Y
B
U
A_1 A_2 A_3 A_{L-1} A_L
W
X_1 X_2 X_3 . . . X_{L-1} X_L

The network uses the input sequence of X sequentially, each X_ℓ feeds into the hidden layer with the activation vector $A_{\ell-1}$as input from the previous element in the sequence to obtain the current vector A_ℓ; the same weights **W, U, B** (parameter estimates) are used as each element is processed. The output layer provides a sequence of predictions O_ℓ from A_ℓ though only the last one, O_L, is the target in a single target response.

RNN takes advantage of sequential data to produces the output response while CNN does that using the spatial nature of data. More specifically, RNN can be expressed by a collections of $K \times$

(*p*+1) shared weights w_{kj} for the input layer of **W** matrix, and **U** as a matrix of K × K of shared weights $u_{kj}$for the hidden-to-hidden layers, and **B** as a *K*+1 vector of shared weights β_k for the output layer, hence RNN uses $A_{\ell k}$ inputs

$$A_{\text{ɫ}k} = g(\omega_{k0} + \textstyle\sum_{j=1}^{p} \omega_{kj} X_{\ell j} + \sum_{s=1}^{K} \omega_{kj} A_{\ell-1,S} \qquad (20.63)$$

to produce O_ℓ outputs

$$O_\ell = \beta_0 + \textstyle\sum_{k=1}^{K} \beta_k A_{\ell k} \qquad (20.64)$$

Note that the weights are not a function of ℓ, the same weights are in each sequence; the A_ℓ accumulates what has been learned before to produce prediction*s*. For regression, a loss function with observations on (*X, Y*) is

$$(Y - O_L)^2 \qquad (20.65)$$

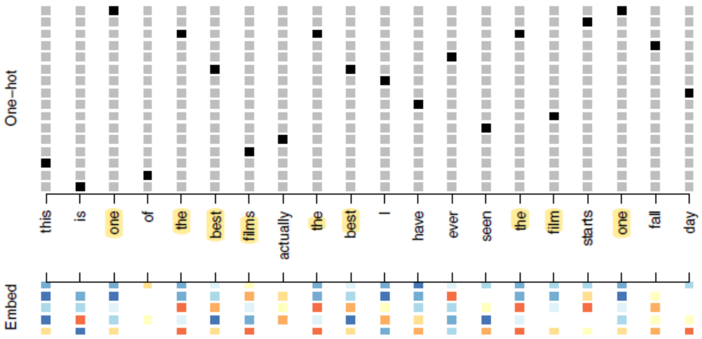
Where the final outcome is produced from $O_L = \beta_0 + \sum_{k=1}^{K} \beta_k A_{LK}$, without using the intermediate values if the target is a single response type, in order to obtain parameter estimates that minimize the sum of squares

$$\textstyle\sum_{i=1}^{n}(y_i - O_{iL})^2 = \sum_{i=1}^{n}((y_i - (\beta_0 + \sum_{k=1}^{K} \beta_k g(\omega_{ko} + \sum_{j=1}^{p} \omega_{kj} x_{iLj} + \sum_{s=1}^{K} u_{ks} \alpha_{i,L-i,s})))^2 \quad (20.66)$$

Example 1: We examined IMDb (Internet Movie Database) with the *Bag-of-Words* model. The other alternative to analyze the document features and response sentiment is by *RNN,* using instead the sequence of words. That is, instead of using a binary vector with 9,999 zeros, we employ a set of *m embedded* real numbers none of which are zeros. Plot 20.17 shows RNN with a dictionary of 16 words rather than 1,000 embedded in *m*=5 dimensions. The idea is to use the information from the positions of embedded words meaning in the text, for example, synonyms are placed near each other. We should also limit each document to L words; those shorter than L are padded with zero upfront so X is presented by L vectors.

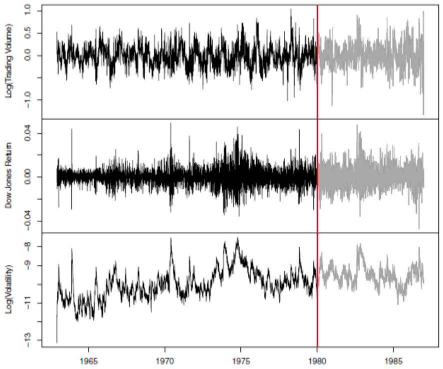
A More detailed *RNN* sequential document process uses the *Long Term and Short Term Memory* (***LSTM***) two track hidden activations so the computation of A_ℓ receives input from hidden units both further back in time, and closer in time. This prevents the early signals being left out by the time they are propagated through the chain to the final A_ℓ. Refitting the model with *LSMT* improves the performance by 87% on the IMDb test data compared to the 88% gained by the *Bag-of-Words* model.

Plot 20.17 RNN for IMDb with LSTM



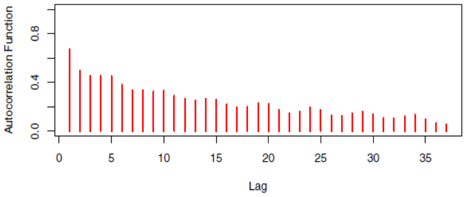
Example 2-Time-series Forecasting. We use the NYSE data set for three daily time series covering Dec. 3, 1962 to Dec. 31, 1986 for the *Log Trading Volume* of shares on that day relative to a 100-day moving average of past turnover, the *Dow Jones Return Difference* on consecutive trading days, and *Log Volatility* for the absolute values of daily price movements; hence we have measurements (v_t, r_t, z_t) on day t and there are T=6,051 such triple observations shown in Plot 20.18 that display~~s~~ significant auto-correlation ; similar values for nearby in time.

Plot 20.18 NRR for NYSE data



This is displayed more clearly when we consider pairs of observations with a lag of ℓ days apart $(v_t,\ v_{t-\ell})$ and compute the auto-correlation coefficient of all lags up to ℓ=37 shown in Plot 20.19 Here the response value is also a predictor; however, the structure of the problem differs from example 1 in that we only have one series, not 25,000 series of short documents even though both examples exploit the sequential nature of the data sets. We can present the time-series RNN process

Plot 20.19 autocorrelation

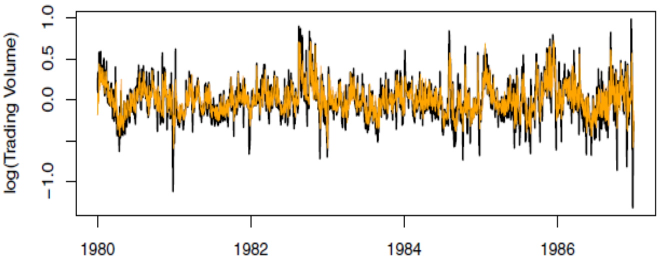


in terms of the sequential Fig. 20.7 model that extracts many short mini-series of predefined length L of input X=($X_1, X_2, \ldots, X_L$) to compute the target Y of the form

$$X_1 = \begin{pmatrix} v_{t-L} \\ r_{t-L} \\ z_{t-L} \end{pmatrix}, \qquad X_2 = \begin{pmatrix} v_{t-L+1} \\ r_{t-L+1} \\ z_{t-L+1} \end{pmatrix}, \ldots, X_L = \begin{pmatrix} v_{t-L} \\ r_{t-L} \\ z_{t-L} \end{pmatrix}, and\ Y = v_t$$

Plot 20.20 uses the NYSE data of the past five trading days to predict the next day trading volume, so L=5, the model fitted with K=12 hidden units using a training sample sequence derived from the data before Jan. 2, 1980, and then forecasted log-volume after that date to obtain R^2=42% on the test data set.

Plot 20.20-*RNN* forecast with NYSE series



This is similar to running a AR(ℓ) regression with L=5 that results R^2=0.41 compared to 0.42 by RNN.

Nonlinear Factor Regression by Autoencoders.

A nonlinear equivalent to the ML linear PCA for dimension reduction is the *Autoencoders* model by which the outputs approximate the input variables. The input variables going through neurons to produce a compressed input which then is decoded or decompressed into the output layer. The method attempts to extract the hidden layer with the smallest number of neurons that represent the latent nonlinear factors. Fig. 20.26 illustrates the procedure with five inputs and three hidden layers with four, one and four neurons respectively. The second hidden layer Q_1^2reprents the latent single factor for extraction; the encoded layer precedes and the decoded layer follows it.

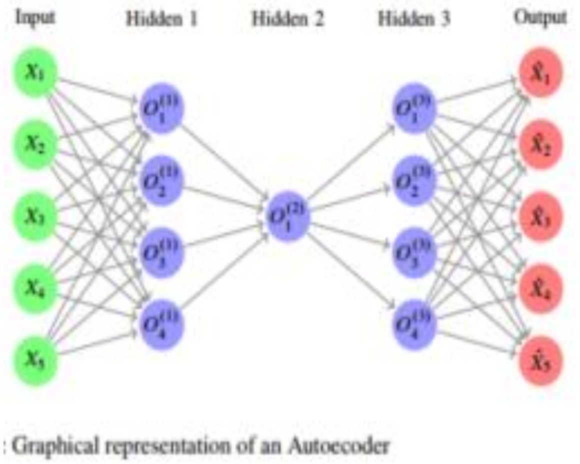


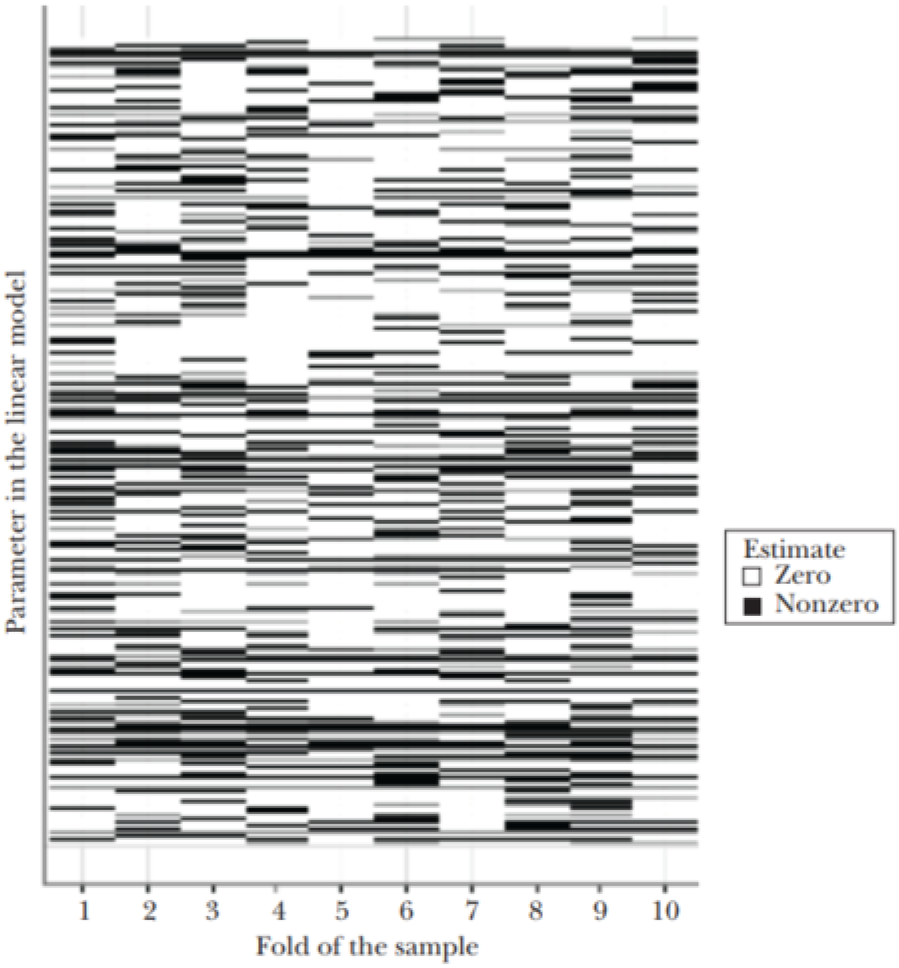
Fig. 20.26 Autoencoder

The estimated nonlinear factors are employed as inputs for ML linear or non-linear forecasting models as in chapter 19 and for tree-based models in this chapter.

20.12 *Overall View of the Relationship between Machine Learning and Econometrics*.

Machine Learning has opened up new, powerful tools for ~~to~~ predictive econometrics tasks that practice of the traditional methods do not permit, especially when the number of features is larger than the number of observations. The traditional methods, particularly ~~the~~ least squares, are impossible to apply while dimension reduction ML tools render the problem a manageable task. The question then is whether a good ML predictive model can be used to identify the underlying model implied by the ML prediction procedure; therefore, provide inference on the coefficient estimation. One problem with ML methods employed for making inference is the absence of standard errors on the coefficients. Even with ML output based linear functions, the standard errors would have to take into account the model selection itself, and it may be impossible to obtain consistent estimates after the model is selected by a data-driven procedure. Moreover, ML methods partition the observations into training and hold-out samples by CV folding, and re-estimate each partition by a sparse ML predictor such as Lasso by regularization that amounts to setting many coefficients equal to zero and hence, not used in the analysis. This results in potential use of different predictors in different ML models used in each partitioning. Plot 20.21 shows ten randomly selected non-zero coefficients across ten Lasso regressions obtained by application to ten group of 50,000 housing units from a 2011 metropolitan sample of the American Housing Survey. The variables used in each partition, namely the black cells, change for each partition; highlighting the fundamental ML inference problem for coefficient estimation. This is not a problem of the quality of prediction- R2 is roughly constant across partitioned subsamples. However, a variable used in one partition is unused in another, resulting in ~~a~~ few stable patterns overall. This problem occurs because the variables are correlated with each other, e.g. the number of rooms and the unit's land area. Then similar predictions can be obtained based on very different model predictors depending on the sample partition used and we have little guidance on the list of variables used. By contrast, in traditional econometrics correlations between observed predictors is indicated by standard errors that may reflect uncertainly about attributing effects to one predictor over the other.

Two very different models can produce similar ML predictions because ML can fit many different functions without having to specify them; hence, the lack of standard errors limits ML from making parameter inference after the selection of the predictive model; in other words, the ML challenge here is parameter consistency estimation itself. Regularization contributes to this problem by selecting a less complex model, thereby encouraging omitted variable bias due to correlations between observed and unobserved variables. While a good predictive model is likely to reveal some underlying structure, and some econometric results suggest convergence- where the structure will be recovered with a high-quality prediction model, we should guide against interpreting parameter estimates as indication of the discovered model structure. In traditional econometrics, assumptions about the data generating process would permit interpretation of the estimated coefficients as model specification, but with ML we have to limit the variable correlation by assumption. Asymptotic Lasso model selection consistency requires assuming the true model has only a few relevant variables; none of the irrelevant variables should be even moderately correlated with the set of sparse relevant variables. It is here in model specification that theory plays an important part for inference in ML application.



Plot 20.21 Missing predictors in nonlinear ML Inference

Still, ML has developed powerful procedures for prediction missing from traditional econometrics. One area is use of satellite images to extract economic outcomes, for instance, for tracking poverty when direct data for a developing country is missing. Another is use of language and text as data for prediction, for example, online financial messages as bullish, bearish or neither to predict market volatility and stock returns. Another type of ML application relevant for econometric application is when estimation requires a prior prediction step. The first stage of the linear instrumental variables model is effectively a predictive step for the endogenous variable by the instruments; the first stage estimates are merely a means to the second stage of consistent parameter estimation. Even with a few instruments, there would still be high-dimensional problems in scaling them, the functions used, and interactions, etc. In the ML approach such problems are resolved by the data picking the effective specification. Similarly, ML can help in policy problem prediction, the impact of an income transfer program on how effectively it is targeted; that would involve predicting group benefits from their features. Finally, ML can test theories directly by treating that task as a model prediction. In this case, there are two approaches to model specification: top-down deductive, or bottom up inductive. The contribution of ML is to employ the inductive tools when the deductive method is hard to apply; it has always been the case that the two approaches co-exist in economics side-by-side.

Selected Reading

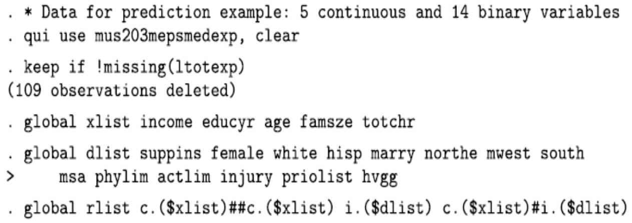
James et. al. (2021) chapter 7, 8 and 10 discuss nonlinear and deep learning M.L. models with many empirical exaples in R; Hastie et. al. (2001) chapters 9, 10 and 11 coverr those models at greater depth and details. Cameron and Trivedi (2022), chapter 28 has several nonlinear M.L. empirical examples in Stata, Muiiainathan et. al. (2017) examines the avantages of M.L. appraoch to econometric prediction while pointing out its drawbacks related to nonlinear inference. Brieman (2001) developed Random Forest.

LAB NONLINEAR SHRINKAGE MODELS

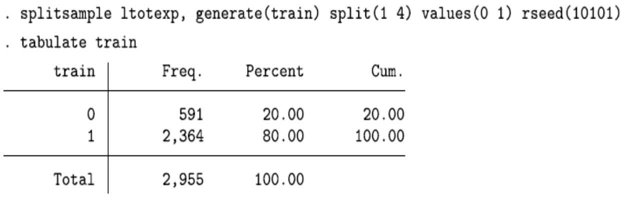
Lab20 1. Single Tree-based model

Open mus203mepsmedexp.dta and use health expenditure with suppl. Insurance as the principal variable of interest. We have 5 continuous and 14 binary variables, creating 188 predictors including interaction terms and start with a split of 80% training sample and 20% error test sample and remove all missing values to have fully observable samples.

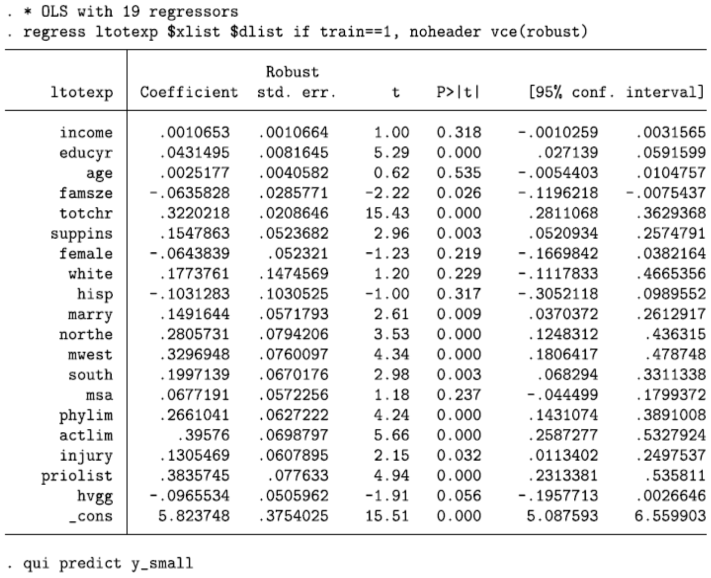
a) Estimate an OLS model of *ltotexp* regressed on the original 19 predictors; an adaptive Lasso with full list of predictors.



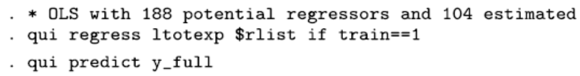
```
. * Data for prediction example: 5 continuous and 14 binary variables
. qui use mus203mepsmedexp, clear

. keep if !missing(ltotexp)
(109 observations deleted)

. global xlist income educyr age famsze totchr

. global dlist suppins female white hisp marry northe mwest south
>     msa phylim actlim injury priolist hvgg

. global rlist c.($xlist)##c.($xlist) i.($dlist) c.($xlist)#i.($dlist)
```



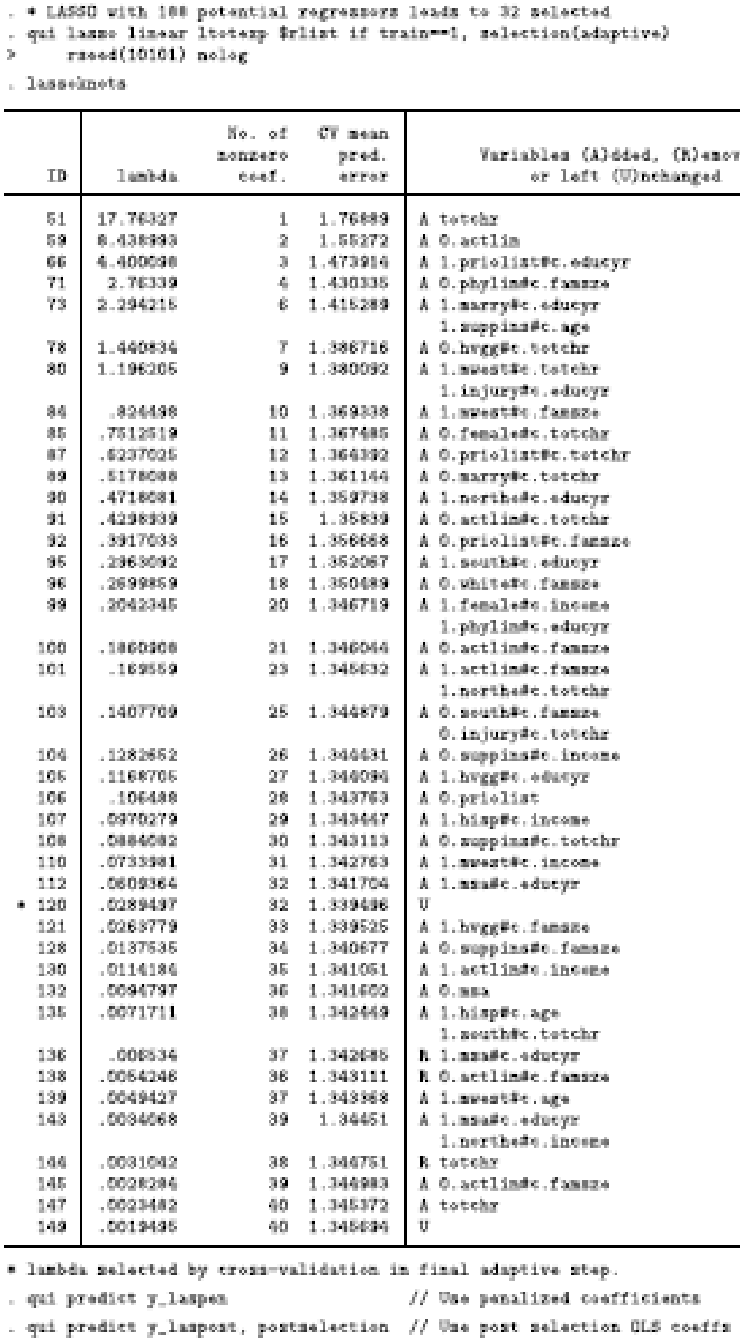
```
. splitsample ltotexp, generate(train) split(1 4) values(0 1) rseed(10101)

. tabulate train

      train |      Freq.     Percent        Cum.
------------+-----------------------------------
          0 |        591       20.00       20.00
          1 |      2,364       80.00      100.00
------------+-----------------------------------
      Total |      2,955      100.00
```



```
. * OLS with 19 regressors
. regress ltotexp $xlist $dlist if train==1, noheader vce(robust)
------------------------------------------------------------------------------
             |               Robust
     ltotexp | Coefficient  std. err.      t    P>|t|     [95% conf. interval]
-------------+----------------------------------------------------------------
      income |   .0010653   .0010664     1.00   0.318    -.0010259    .0031565
      educyr |   .0431495   .0081645     5.29   0.000      .027139    .0591599
         age |   .0025177   .0040582     0.62   0.535    -.0054403    .0104757
      famsze |  -.0635828   .0285771    -2.22   0.026    -.1196218   -.0075437
      totchr |   .3220218   .0208646    15.43   0.000     .2811068    .3629368
     suppins |   .1547863   .0523682     2.96   0.003     .0520934    .2574791
      female |  -.0643839    .052321    -1.23   0.219    -.1669842    .0382164
       white |   .1773761   .1474569     1.20   0.229    -.1117833    .4665356
        hisp |  -.1031283   .1030525    -1.00   0.317    -.3052118    .0989552
       marry |   .1491644   .0571793     2.61   0.009     .0370372    .2612917
      northe |   .2805731   .0794206     3.53   0.000     .1248312     .436315
       mwest |   .3296948   .0760097     4.34   0.000     .1806417     .478748
       south |   .1997139   .0670176     2.98   0.003      .068294    .3311338
         msa |   .0677191   .0572256     1.18   0.237     -.044499    .1799372
      phylim |   .2661041   .0627222     4.24   0.000     .1431074    .3891008
      actlim |     .39576   .0698797     5.66   0.000     .2587277    .5327924
      injury |   .1305469   .0607895     2.15   0.032     .0113402    .2497537
    priolist |   .3835745    .077633     4.94   0.000     .2313381     .535811
        hvgg |  -.0965534   .0505962    -1.91   0.056    -.1957713    .0026646
       _cons |   5.823748   .3754025    15.51   0.000     5.087593    6.559903
------------------------------------------------------------------------------

. qui predict y_small
```



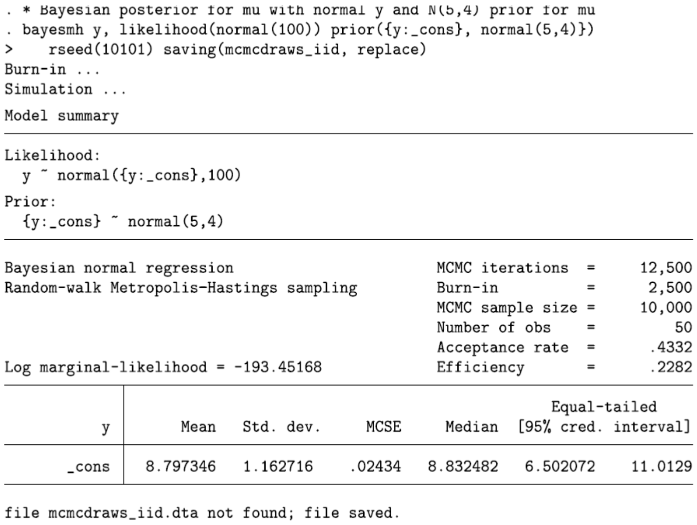
```
. * OLS with 188 potential regressors and 104 estimated
. qui regress ltotexp $rlist if train==1
. qui predict y_full
```



```
. * LASSO with 188 potential regressors leads to 32 selected
. qui lasso linear ltotexp $rlist if train==1, selection(adaptive)
>     rseed(10101) nolog

. lassoknots

─────────┬──────────────────────────────────┬─────────────────────────────────────
         │                 No. of   CV mean │
         │                nonzero     pred. │    Variables (A)dded, (R)emoved,
      ID │     lambda       coef.     error │          or left (U)nchanged
─────────┼──────────────────────────────────┼─────────────────────────────────────
      51 │   17.76327           1   1.76889 │ A totchr
      59 │   6.438993           2   1.55272 │ A 0.actlim
      66 │   4.400698           3  1.473914 │ A 1.prioliist#c.educyr
      71 │    2.76339           4  1.430335 │ A 0.phylim#c.famsze
      73 │   2.294215           6  1.415289 │ A 1.marry#c.educyr
         │                                  │   1.suppins#c.age
      78 │   1.440834           7  1.386716 │ A 0.hvgg#c.totchr
      80 │   1.196205           9  1.380092 │ A 1.mwest#c.totchr
         │                                  │   1.injury#c.educyr
      84 │    .824498          10  1.369338 │ A 1.mwest#c.famsze
      85 │   .7512519          11  1.367485 │ A 0.female#c.totchr
      87 │   .6237025          12  1.364392 │ A 0.prioliist#c.totchr
      89 │   .5178088          13  1.361144 │ A 0.marry#c.totchr
      90 │   .4718081          14  1.359738 │ A 1.northe#c.educyr
      91 │   .4298939          15   1.35839 │ A 0.actlim#c.totchr
      92 │   .3917033          16  1.356668 │ A 0.prioliist#c.famsze
      95 │   .2963092          17  1.352067 │ A 1.south#c.educyr
      96 │   .2699859          18  1.350489 │ A 0.white#c.famsze
      99 │   .2042345          20  1.346719 │ A 1.female#c.income
         │                                  │   1.phylim#c.educyr
     100 │   .1860908          21  1.346044 │ A 0.actlim#c.famsze
     101 │    .169559          23  1.345632 │ A 1.actlim#c.famsze
         │                                  │   1.northe#c.totchr
     103 │   .1407709          25  1.344879 │ A 0.south#c.famsze
         │                                  │   0.injury#c.totchr
     104 │   .1282652          26  1.344431 │ A 0.suppins#c.income
     105 │   .1168705          27  1.344094 │ A 1.hvgg#c.educyr
     106 │    .106488          28  1.343763 │ A 0.prioliist
     107 │   .0970279          29  1.343467 │ A 1.hisp#c.income
     108 │   .0884082          30  1.343113 │ A 0.suppins#c.totchr
     110 │   .0733981          31  1.342763 │ A 1.mwest#c.income
     112 │   .0609364          32  1.341704 │ A 1.msa#c.educyr
   * 120 │   .0289497          32  1.339496 │ U
     121 │   .0263779          33  1.339525 │ A 1.hvgg#c.famsze
     128 │   .0137535          34  1.340677 │ A 0.suppins#c.famsze
     130 │   .0114184          35  1.341051 │ A 1.actlim#c.income
     132 │   .0094797          36  1.341602 │ A 0.msa
     135 │   .0071711          38  1.342449 │ A 1.hisp#c.age
         │                                  │   1.south#c.totchr
     136 │    .006534          37  1.342685 │ R 1.msa#c.educyr
     138 │   .0054246          36  1.343111 │ R 0.actlim#c.famsze
     139 │   .0049427          37  1.343368 │ A 1.mwest#c.age
     143 │   .0034068          39   1.34451 │ A 1.msa#c.educyr
         │                                  │   1.northe#c.income
     144 │   .0031042          38  1.344751 │ R totchr
     145 │   .0028284          39  1.344983 │ A 0.actlim#c.famsze
     147 │   .0023482          40  1.345372 │ A totchr
     149 │   .0019495          40  1.345694 │ U
─────────┴──────────────────────────────────┴─────────────────────────────────────
* lambda selected by cross-validation in final adaptive step.

. qui predict y_laspen                    // Use penalized coefficients

. qui predict y_laspost, postselection   // Use post selection OLS coeffs
```

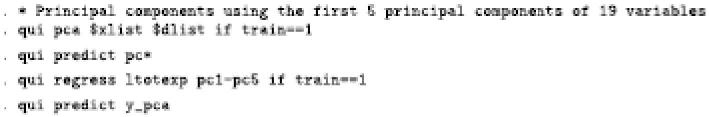
b) Now fit a multiple-tree Random forests model to the data set in a); then again fit a BART model to the same data.

```
. * Random forest with 19 variables
. qui rforest ltotexp $xlist $dlist if train==1,
>     type(reg) iter(200) depth(10) lsize(5)

. qui predict y_ranfor
```



```
. * Bayesian posterior for mu with normal y and N(5,4) prior for mu
. bayesmh y, likelihood(normal(100)) prior({y:_cons}, normal(5,4)})
>    rseed(10101) saving(mcmcdraws_iid, replace)
Burn-in ...
Simulation ...
Model summary
------------------------------------------------------------------------------
Likelihood:
  y ~ normal({y:_cons},100)
Prior:
  {y:_cons} ~ normal(5,4)
------------------------------------------------------------------------------

Bayesian normal regression                       MCMC iterations  =     12,500
Random-walk Metropolis-Hastings sampling         Burn-in          =      2,500
                                                 MCMC sample size =     10,000
                                                 Number of obs    =         50
                                                 Acceptance rate  =      .4332
Log marginal-likelihood = -193.45168             Efficiency       =      .2282
------------------------------------------------------------------------------
             |                                                Equal-tailed
           y |      Mean   Std. dev.     MCSE     Median  [95% cred. interval]
-------------+----------------------------------------------------------------
       _cons |  8.797346   1.162716    .02434   8.832482   6.502072    11.0129
------------------------------------------------------------------------------

file mcmcdraws_iid.dta not found; file saved.
```

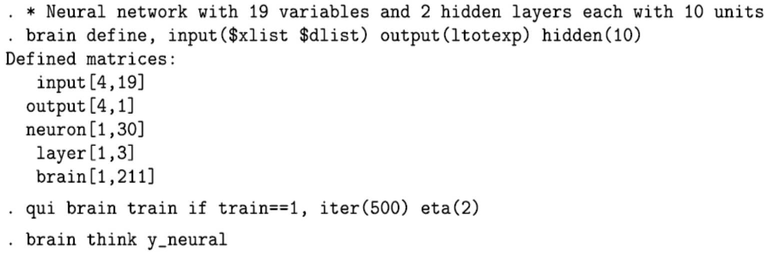
c) Fit a PCA model with 5 PC of the 19 underlying predictors.



```
. * Principal components using the first 5 principal components of 19 variables
. qui pca $xlist $dlist if train==1
. qui predict pc*
. qui regress ltotexp pc1-pc5 if train==1
. qui predict y_pca
```

Lab20 2. Neural networks**.**

a) Open mus203mepsmedexp.dta and fit a non-linear Neural net model with 19 variables and 2 hidden layers each having 10 units



```
. * Neural network with 19 variables and 2 hidden layers each with 10 units
. brain define, input($xlist $dlist) output(ltotexp) hidden(10)
Defined matrices:
   input[4,19]
  output[4,1]
  neuron[1,30]
   layer[1,3]
   brain[1,211]
. qui brain train if train==1, iter(500) eta(2)
. brain think y_neural
```

***Lab20 3*. Comparison of non-linear models.**

b) Compare Training and Test samples MES by various methods.

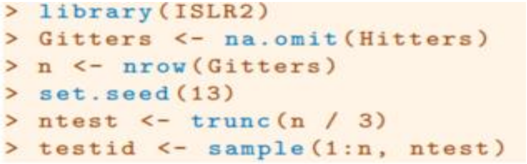
```
. * Training MSE and test MSE for the various methods
. qui regress ltotexp

. qui predict y_noreg

. foreach var of varlist y_noreg y_small y_full y_laspen y_laspost y_pca
> y_neural y_ranfor y_boost {
  2.     qui gen `var´errorsq = (`var´ - ltotexp)^2
  3.     qui sum `var´errorsq if train == 1
  4.     scalar mse`var´train = r(mean)
  5.     qui sum `var´errorsq if train == 0
  6.     qui scalar mse`var´test = r(mean)
  7.     display "Predictor: " "`var´" _col(21)
>        " Train MSE = " %5.3f mse`var´train "  Test MSE = " %5.3f mse`var´test
  8.     }
Predictor: y_noreg   Train MSE = 1.821  Test MSE = 2.063
Predictor: y_small   Train MSE = 1.339  Test MSE = 1.492
Predictor: y_full    Train MSE = 1.262  Test MSE = 1.509
Predictor: y_laspen  Train MSE = 1.298  Test MSE = 1.491
Predictor: y_laspost Train MSE = 1.297  Test MSE = 1.493
Predictor: y_pca     Train MSE = 1.397  Test MSE = 1.545
Predictor: y_neural  Train MSE = 1.211  Test MSE = 1.808
Predictor: y_ranfor  Train MSE = 1.047  Test MSE = 1.574
Predictor: y_boost   Train MSE = 1.459  Test MSE = 1.664
```

* The in-sample MSE is smallest for RF and neural net but the out-of-sample as the lowest MSE for the OLS with regressors and lasso estimators.

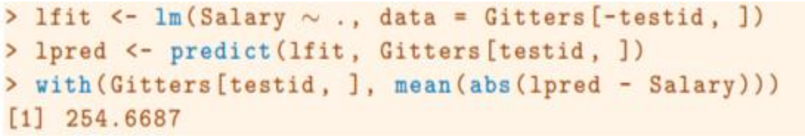
Lab20 4. Single layer neural net.

Note: This section requires *keras* package interface to *python* code via *tensorflow* package to fit neural models in ***R***.

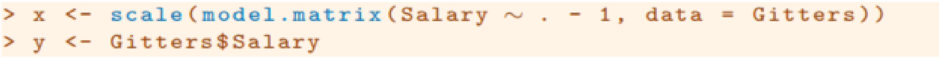
Fit a single layer neural net to hitters data set



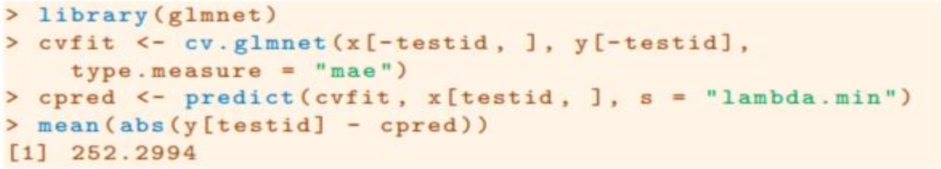
```
> library(ISLR2)
> Gitters <- na.omit(Hitters)
> n <- nrow(Gitters)
> set.seed(13)
> ntest <- trunc(n / 3)
> testid <- sample(1:n, ntest)
```



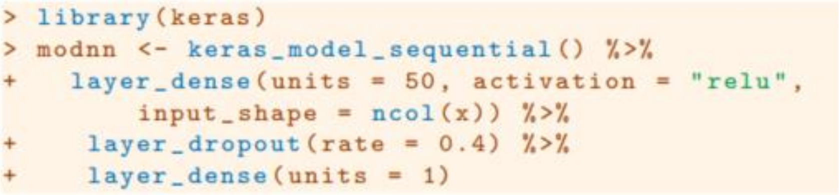
```
> lfit <- lm(Salary ~ ., data = Gitters[-testid, ])
> lpred <- predict(lfit, Gitters[testid, ])
> with(Gitters[testid, ], mean(abs(lpred - Salary)))
[1] 254.6687
```



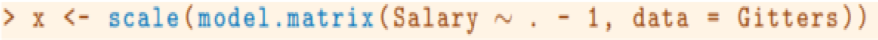
```
> x <- scale(model.matrix(Salary ~ . - 1, data = Gitters))
> y <- Gitters$Salary
```



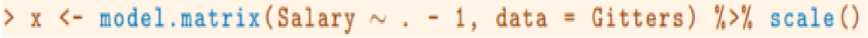
```
> library(glmnet)
> cvfit <- cv.glmnet(x[-testid, ], y[-testid],
    type.measure = "mae")
> cpred <- predict(cvfit, x[testid, ], s = "lambda.min")
> mean(abs(y[testid] - cpred))
[1] 252.2994
```



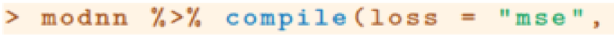
```
> library(keras)
> modnn <- keras_model_sequential() %>%
+   layer_dense(units = 50, activation = "relu",
        input_shape = ncol(x)) %>%
+     layer_dropout(rate = 0.4) %>%
+     layer_dense(units = 1)
```



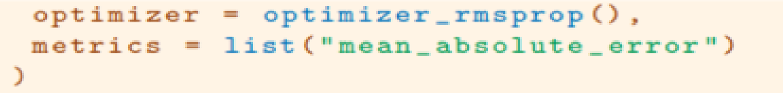
```
> x <- scale(model.matrix(Salary ~ . - 1, data = Gitters))
```



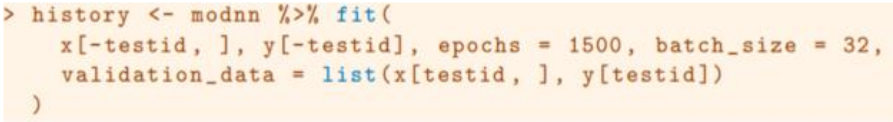
```
> x <- model.matrix(Salary ~ . - 1, data = Gitters) %>% scale()
```



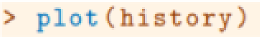
```
> modnn %>% compile(loss = "mse",
```



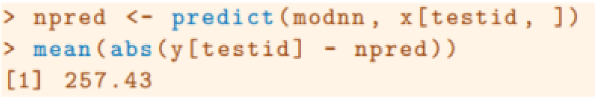
```
  optimizer = optimizer_rmsprop(),
  metrics = list("mean_absolute_error")
)
```



```
> history <- modnn %>% fit(
    x[-testid, ], y[-testid], epochs = 1500, batch_size = 32,
    validation_data = list(x[testid, ], y[testid])
  )
```



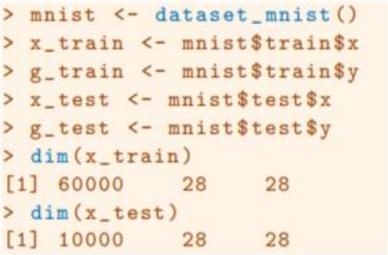
```
> plot(history)
```



```
> npred <- predict(modnn, x[testid, ])
> mean(abs(y[testid] - npred))
[1] 257.43
```

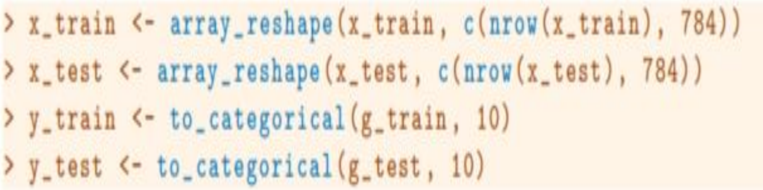
Lab 20 5. Multiple layers Nets.

a) Open MNIST data set that comes with the *keras* package. This involves several steps in R.

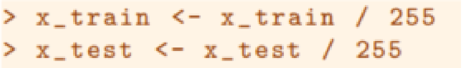


```
> mnist <- dataset_mnist()
> x_train <- mnist$train$x
> g_train <- mnist$train$y
> x_test <- mnist$test$x
> g_test <- mnist$test$y
> dim(x_train)
[1] 60000    28    28
> dim(x_test)
[1] 10000    28    28
```

- 60000 images in the training and 10,000 in the test data set. We scale the unit inputs by grayscale values (0, 255).

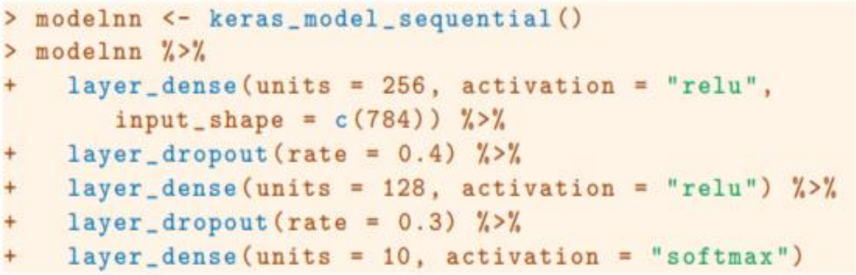


```
> x_train <- array_reshape(x_train, c(nrow(x_train), 784))
> x_test <- array_reshape(x_test, c(nrow(x_test), 784))
> y_train <- to_categorical(g_train, 10)
> y_test <- to_categorical(g_test, 10)
```



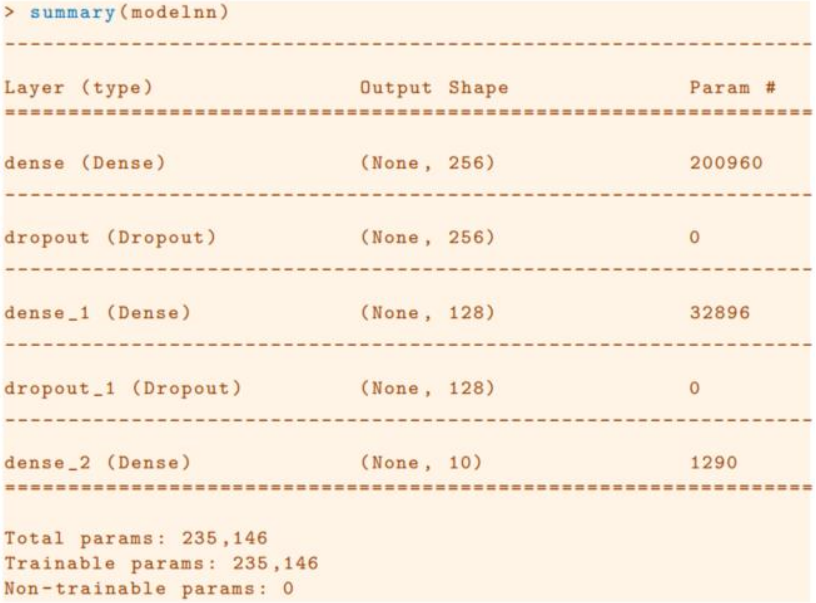
```
> x_train <- x_train / 255
> x_test <- x_test / 255
```

b) Fit a multi-layer neural net model to the data.

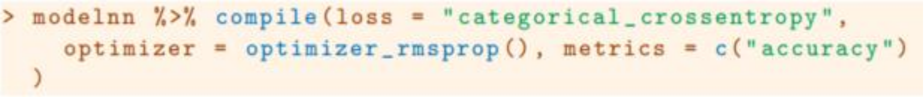


```
> modelnn <- keras_model_sequential()
> modelnn %>%
+   layer_dense(units = 256, activation = "relu",
      input_shape = c(784)) %>%
+   layer_dropout(rate = 0.4) %>%
+   layer_dense(units = 128, activation = "relu") %>%
+   layer_dropout(rate = 0.3) %>%
+   layer_dense(units = 10, activation = "softmax")
```

- The first layer of input units 28 by 28=784 goes into a hidden layer using ReLU activation function, dropout layer follows from the second hidden layer with 128 units, and the final output layer uses activation *softmax* for a 10-class classification- model minimization is by cross-entropy function.

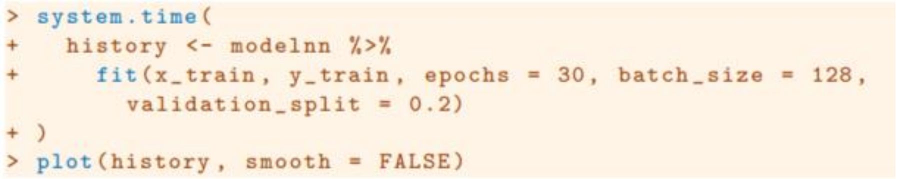


```
> summary(modelnn)
________________________________________________________________
Layer (type)                 Output Shape              Param #
================================================================
dense (Dense)                (None, 256)               200960
________________________________________________________________
dropout (Dropout)            (None, 256)               0
________________________________________________________________
dense_1 (Dense)              (None, 128)               32896
________________________________________________________________
dropout_1 (Dropout)          (None, 128)               0
________________________________________________________________
dense_2 (Dense)              (None, 10)                1290
================================================================
Total params: 235,146
Trainable params: 235,146
Non-trainable params: 0
```

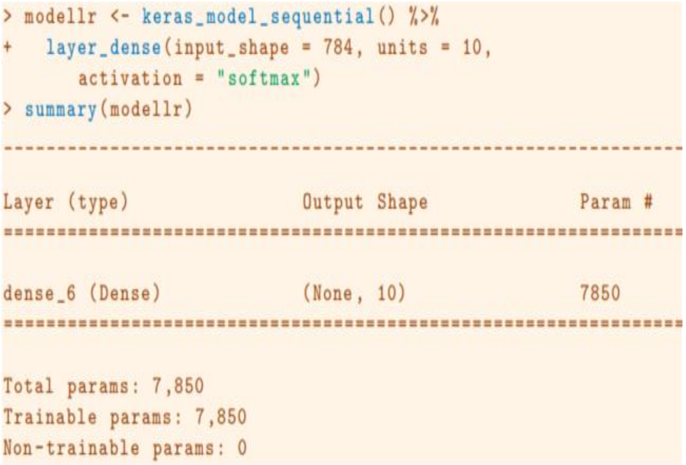


```
> modelnn %>% compile(loss = "categorical_crossentropy",
    optimizer = optimizer_rmsprop(), metrics = c("accuracy")
  )
```

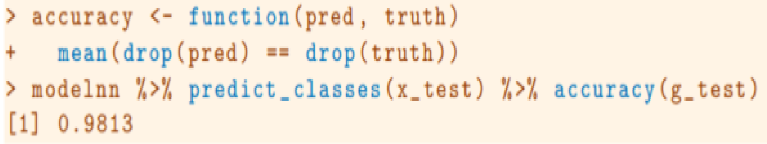
- Now we fit the model to the training data and obtain the test error



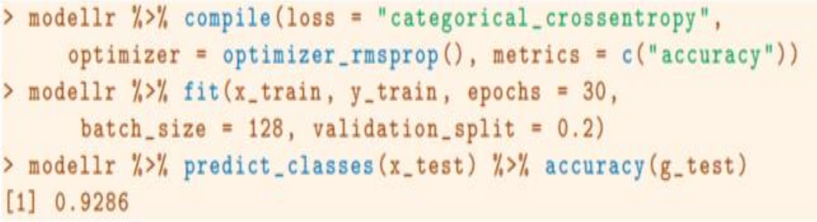
```
> system.time(
+   history <- modelnn %>%
+     fit(x_train, y_train, epochs = 30, batch_size = 128,
        validation_split = 0.2)
+ )
> plot(history, smooth = FALSE)
```



```
> modellr <- keras_model_sequential() %>%
+   layer_dense(input_shape = 784, units = 10,
      activation = "softmax")
> summary(modellr)
----------------------------------------------------------------
Layer (type)                Output Shape              Param #
================================================================
dense_6 (Dense)             (None, 10)                7850
================================================================
Total params: 7,850
Trainable params: 7,850
Non-trainable params: 0
```



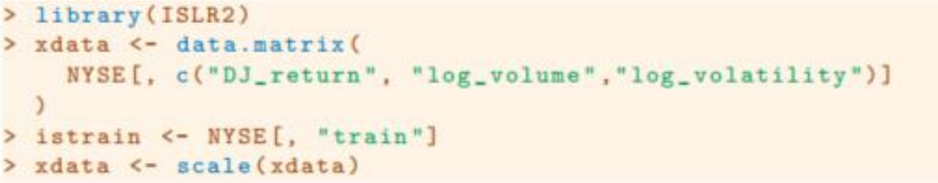
```
> accuracy <- function(pred, truth)
+   mean(drop(pred) == drop(truth))
> modelnn %>% predict_classes(x_test) %>% accuracy(g_test)
[1] 0.9813
```



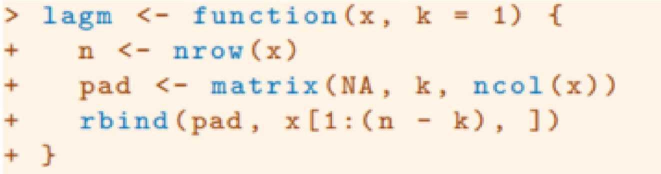
```
> modellr %>% compile(loss = "categorical_crossentropy",
    optimizer = optimizer_rmsprop(), metrics = c("accuracy"))
> modellr %>% fit(x_train, y_train, epochs = 30,
      batch_size = 128, validation_split = 0.2)
> modellr %>% predict_classes(x_test) %>% accuracy(g_test)
[1] 0.9286
```

Lab-x 6. RNN Time-series prediction.

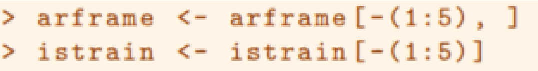
a) Fit a time-series AR predictive model for prediction to the NYSE data set after standardizing, creating a data frame.



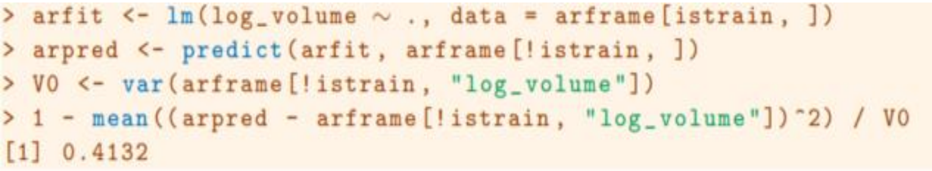
```
> library(ISLR2)
> xdata <- data.matrix(
    NYSE[, c("DJ_return", "log_volume","log_volatility")]
  )
> istrain <- NYSE[, "train"]
> xdata <- scale(xdata)
```



```
> lagm <- function(x, k = 1) {
+   n <- nrow(x)
+   pad <- matrix(NA, k, ncol(x))
+   rbind(pad, x[1:(n - k), ])
+ }
```

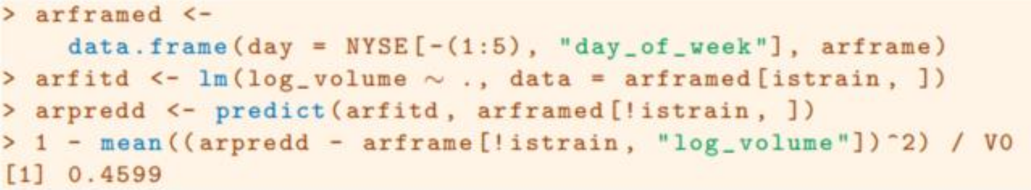


```
> arframe <- arframe[-(1:5), ]
> istrain <- istrain[-(1:5)]
```



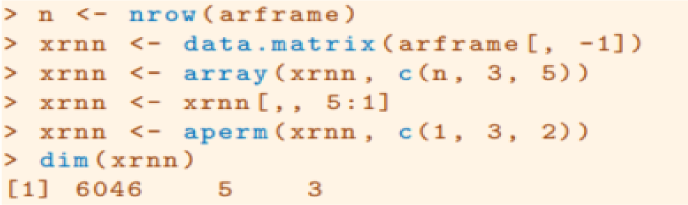
```
> arfit <- lm(log_volume ~ ., data = arframe[istrain, ])
> arpred <- predict(arfit, arframe[!istrain, ])
> V0 <- var(arframe[!istrain, "log_volume"])
> 1 - mean((arpred - arframe[!istrain, "log_volume"])^2) / V0
[1] 0.4132
```

b) Refit the last model adding the factor *day-of-week*.



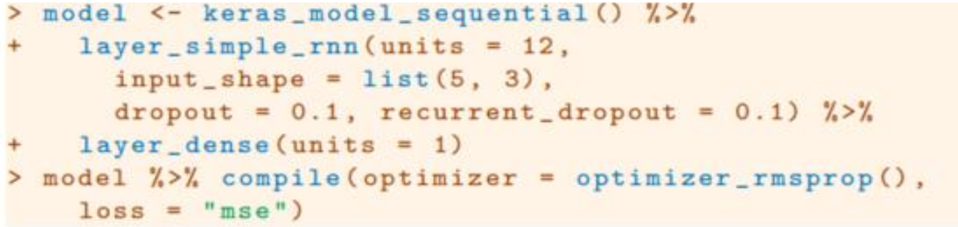
```
> arframed <-
    data.frame(day = NYSE[-(1:5), "day_of_week"], arframe)
> arfitd <- lm(log_volume ~ ., data = arframed[istrain, ])
> arpredd <- predict(arfitd, arframed[!istrain, ])
> 1 - mean((arpredd - arframe[!istrain, "log_volume"])^2) / V0
[1] 0.4599
```

c) Reshape the data as a sequence of *L*=5 feature vectors as the lagged version of the time-series back to *L*=5

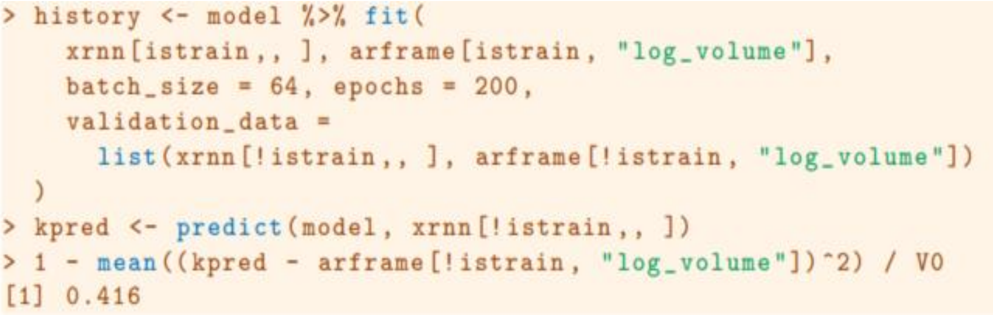


```
> n <- nrow(arframe)
> xrnn <- data.matrix(arframe[, -1])
> xrnn <- array(xrnn, c(n, 3, 5))
> xrnn <- xrnn[,, 5:1]
> xrnn <- aperm(xrnn, c(1, 3, 2))
> dim(xrnn)
[1] 6046    5    3
```

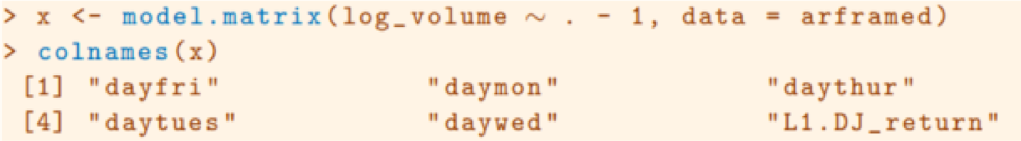
d) Fit the RNN using 12 hidden units with two forms of dropout feeding into the hidden layer by the input sequence and by previous hidden units.



```
> model <- keras_model_sequential() %>%
+   layer_simple_rnn(units = 12,
      input_shape = list(5, 3),
      dropout = 0.1, recurrent_dropout = 0.1) %>%
+   layer_dense(units = 1)
> model %>% compile(optimizer = optimizer_rmsprop(),
    loss = "mse")
```

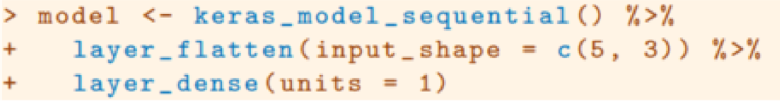


```
> history <- model %>% fit(
    xrnn[istrain,, ], arframe[istrain, "log_volume"],
    batch_size = 64, epochs = 200,
    validation_data =
      list(xrnn[!istrain,, ], arframe[!istrain, "log_volume"])
  )
> kpred <- predict(model, xrnn[!istrain,, ])
> 1 - mean((kpred - arframe[!istrain, "log_volume"])^2) / V0
[1] 0.416
```

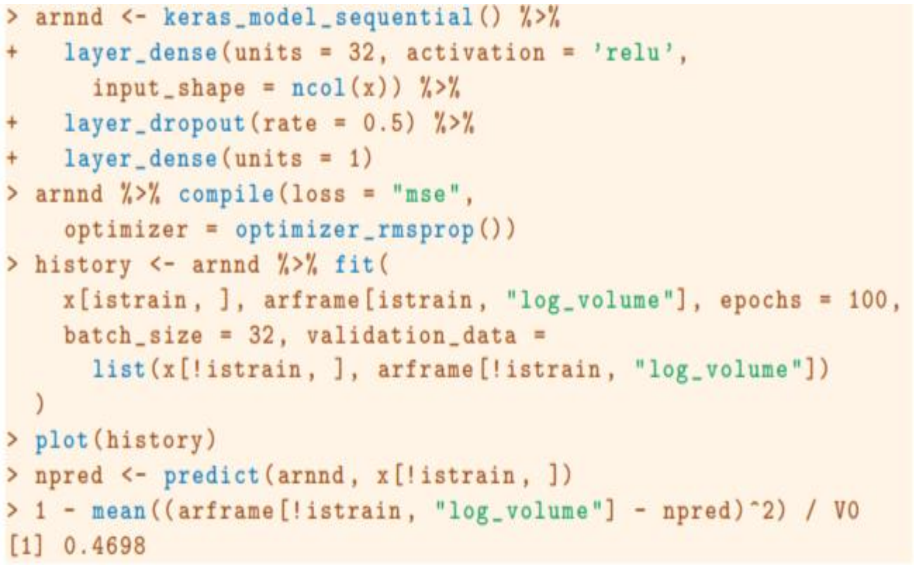


```
> x <- model.matrix(log_volume ~ . - 1, data = arframed)
> colnames(x)
 [1] "dayfri"          "daymon"          "daythur"
 [4] "daytues"         "daywed"          "L1.DJ_return"
```

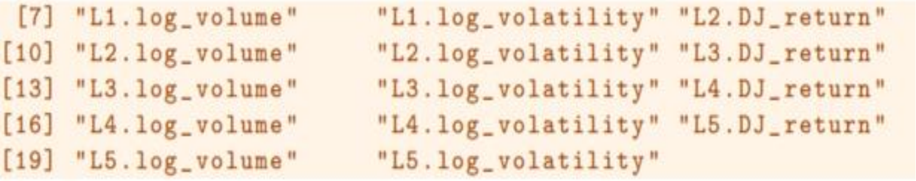
- We could replace the *keras* command with the following



```
> model <- keras_model_sequential() %>%
+   layer_flatten(input_shape = c(5, 3)) %>%
+   layer_dense(units = 1)
```



```
> arnnd <- keras_model_sequential() %>%
+   layer_dense(units = 32, activation = 'relu',
      input_shape = ncol(x)) %>%
+   layer_dropout(rate = 0.5) %>%
+   layer_dense(units = 1)
> arnnd %>% compile(loss = "mse",
    optimizer = optimizer_rmsprop())
> history <- arnnd %>% fit(
    x[istrain, ], arframe[istrain, "log_volume"], epochs = 100,
    batch_size = 32, validation_data =
      list(x[!istrain, ], arframe[!istrain, "log_volume"])
  )
> plot(history)
> npred <- predict(arnnd, x[!istrain, ])
> 1 - mean((arframe[!istrain, "log_volume"] - npred)^2) / V0
[1] 0.4698
```



```
 [7] "L1.log_volume"     "L1.log_volatility" "L2.DJ_return"
[10] "L2.log_volume"     "L2.log_volatility" "L3.DJ_return"
[13] "L3.log_volume"     "L3.log_volatility" "L4.DJ_return"
[16] "L4.log_volume"     "L4.log_volatility" "L5.DJ_return"
[19] "L5.log_volume"     "L5.log_volatility"
```

LAB EXERCISES NONLINEAR SHRINKAKAGE

Lab20_x 1. Single Tree: Classification

Open Carseats data file.

a) Transform sales into a classification variable for high and low sales base on 8 as the threshold, merge the variable into a R data frame, obtain misclassification error from the descriptives, and plot the tree.

b) plot the error rate and obtain the new test error prediction percentage. Does a larger tree produce a better test error?

***Lab20_x* 2.** Single Tree: Regression

Open the data set Boston

a) Fit a tree to the training set and plot the tree.

b) Prune the tree to see if that improves prediction error.

Lab20_x 3. Bagging & Random Forest

a) Use Boston house price data to fit a bagging model (a special case of Random Forest with *m*=*p*).

b) Fit a Random Forest model, obtain the importance of each variable and plot the outcome.

Lab20_x 4. Boosting

Q-Fit a Boosting model to the Boston data, plot the descriptives, and use that to predict *medv* on the test set with different values for the shrinkage parameter λ.

Lab20_x 5. CNN

Open the CIFAR100 image data available in the *keras* package.

a) Standardize the images with one-hot encode he response factors to produce a binary matrix, and provide some of the training images.

b) Specify a moderate-size *CNN* with the same output and input channels,

c) Fit the model by specifying the fitting algorithm

Lab20_x 6. RNN

Open the IMDb movie-review data

Q-Fit a sequential document classification LSTM two-layer model of RNN

Mathematical Appendix

Characteristic Roots

Let square matrix A have (*n.n*) dimension, with a_{ij} elements and x be a *(n.1)* vector such that

$$Ax=\lambda x$$

then the scalar λ is called the *characteristic root* of *A*; rewrite this with the help of an identity matrix *I* of (n.n) dimension as

$$(A - \lambda I)x=0$$

Given all elements of x are zero, that is $|A - \lambda I|_{\text{det}}=0$, we solve the characteristic root equations, also known as *eigenvalues*, that satisfy the above.

Examples below shows how to obtain λ solutions for *A*.

Example 1:

$$|A - \lambda I| = \begin{bmatrix} 0.5 & -0.2 \\ -0.2 & 0.5 \end{bmatrix} - \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix} = \begin{bmatrix} 0.5-\lambda & -0.2 \\ -0.2 & 0.5-\lambda \end{bmatrix}$$

The solving for λ that make $|A - \lambda I|_{\text{det}}=0$ leads to the quadratic equation $\lambda^2 - \lambda +.21=0$. Let a=1, b= -1 ; c=0.21, solving this quadratic equation by $\lambda = \frac{-b\pm\sqrt{b^2-4ac}}{2a}$. The two solutions are λ_1 =0.7 and λ_2 =0.3; in this example both are *real number* characteristic roots or eigenvalues. However, it is also possible to have *complex numbers* as solutions if $(\sqrt{b^2-4ac}) < 0$, see below.

Example 2:

Now change A so each element in column 2 is twice the value in column 1:

$$|A - \lambda I| = \begin{bmatrix} 0.5 & 1 \\ -0.2 & -0.4 \end{bmatrix} - \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix} = \begin{bmatrix} 0.5-\lambda & 1 \\ -0.2 & -0.4-\lambda \end{bmatrix}$$

Solving for the quadratic equation $\lambda^2 - 0.1\,\lambda = 0$ leads to the two characteristic roots (both real number) of λ_1 =0 and λ_2 =0.1.

The polynomial form of the characteristic equation depends on the number of its eigenvalue roots *n;* so, with *A* matrix with (*n.n*) dimension, the equation can be generalized as

$$\lambda^n + b_1\lambda^{n-1} + b_2\lambda^{n-2} + \ldots + b_{n-1}\lambda + b_n = 0.$$

The *n* roots of the characteristic equation can be positive, or negative, and in any case difficult to obtain; it is necessary to rely on numerical methods to solve for the *n* roots. However, often it is enough to know the qualitative features of the solution; sufficiency condition for the solutions to exist is that all eigenvalues of the characteristic equation be less than one in absolute values; we check this condition by ascertaining that the roots all fall within a circle with a unit radius, see discussion below. Note that b_n is the only term in the equation unaffected by λ and defined by

b_n =(- 1)^n|A|$_{\text{det}}$.

It follows that λ^n and b_n will have the same sign if *n* is even and opposite signs if *n* is odd. Let us check this rule with the above examples. For instance, take first example equation λ^2 - λ +.21=0, b_2=0.21, i.e. b1=-1, b2=0.21, and |A|=0.5*0.5 - 0(- 0.2* -0.2=0.21, thus b_2=(-1)^2*(0.21). For λ^2 – 0.1 λ =0, b_2=0 and |A|=0, thus b_2=(-1)^2*(0). Take an example with *n*=3:

$$|\mathrm{A} - \lambda I| = \begin{bmatrix} 0.5-\lambda & 0.2 & 0.2 \\ 0.2 & 0.5-\lambda & 0.2 \\ 0.2 & 0.2 & 0.5-\lambda \end{bmatrix}$$

The characteristic equation is λ^3 – 1.5 λ^2 +0.63 λ +0.081=0, eigenvalues are λ_1=0.9, λ_2=0.3 and λ_3=0.3; |A|det=0.081, resulting in b_3=(-1)^3*(0.081).

The characteristic roots can take any values and fall into three possible outcomes:

1. ***All b$_i$ are real and distinct***. Three subcases can occur. First, 0< b_i< 1, the homogenous equation *converges* since the limit of each b^t_i equal zero as $t \rightarrow \infty$. If b_i< 0 & |b_i<|1, b^t_i will be positive for event values and negative for odd t values, the solution will again display convergence with oscillation. Finally. With |b_i | > 1, the solution will *diverge*.
2. ***All b$_i$ are real but m≤ n of the roots are repeated.*** Let the single common solution be $b_1 = b_2 = \ldots = b_m = \bar{b}$ and the remaining distinct solutions *n-m* roots denoted by b_{m+1} through b_n. In the former case, there will be *m* repeated solutions ($t\bar{b}^t$, $t^2\bar{b}^t$, …, $t^{m-1}\bar{b}^t$) to the homogenous equation.
3. ***Some roots are complex***, see below**.** Complex root solutions to a homogenous equation that come in conjugate pairs and have the form $b_i \pm i\varphi$, b_i and φ are real number and i =

$\sqrt{-1}$. They are usually expressed as polar coordinates of trigonometric relationships, see below.

Determinants and Eigenvalues

The determinants of *A* square matrix (n.n) is equal to the product of its eigenvalue roots:

$$|A|= \prod_{i=1}^{n} \lambda_i$$

Therefore, to solve for the quadratic characteristic equation solutions $\lambda_1$and λ_2 , we must have

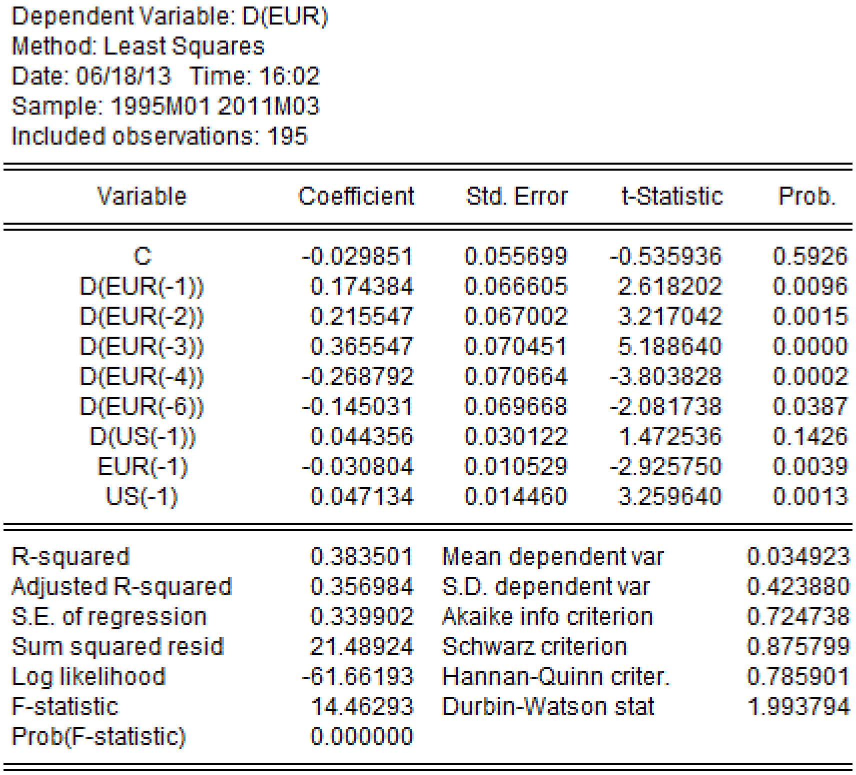
$$(\lambda - \lambda_1)(\lambda - \lambda_2)= \lambda^2 + (\lambda_1 - \lambda_2)\, \lambda + \lambda_1\lambda_2=0$$

It follows that $b_2= \lambda_1\lambda_2$. Checking for λ^2 - λ +.21=0, b_2= 0.21 and $\lambda_1\lambda_2$ =(0.3)(0.7)= 0.21 and also $|A|_{det}=(0.5)^2-(0.2)^2=0.21$. For λ^2 – 0.1 λ =0, b_2=0, $\lambda_1\lambda_2$=(0)(0.1)=0 and also $|A|_{det}$=0. And for λ^3 – 1.5 λ^2 +0.63 λ +0.081=0, b_3= - 0.081 and $\lambda_1\lambda_2\lambda_3$=(0.9)(0.3)(0.3)=0.081 (opposite signs), and $|A|_{det}$=0.081.

Unit Circle Test of Characteristic Roots Stability

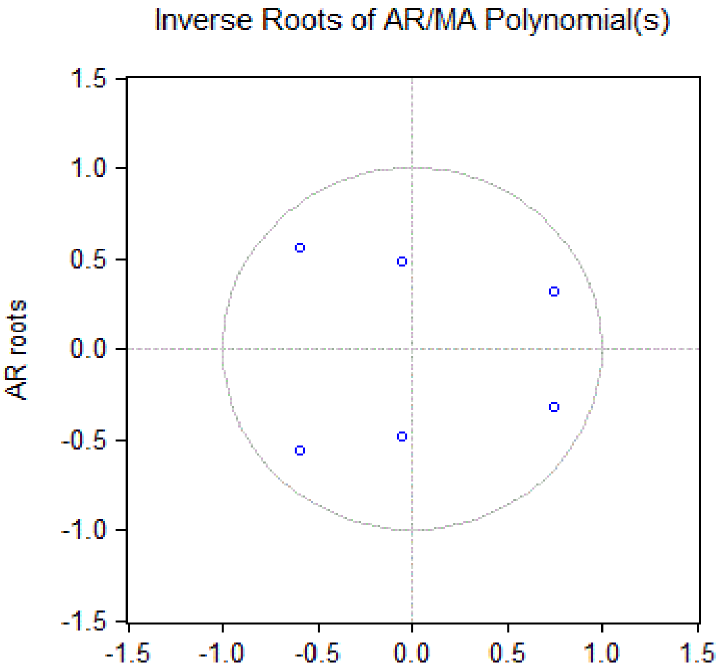
A time-series is stable when its lagged values decline as we move further back in time, that is, the lag variable coefficients become smaller as $T\rightarrow\infty$; otherwise, the series will be explosive. That equivalently depends on the size of the solutions for the eigenvalue roots of the characteristic equation of the series. Since some of the roots may be complex and some real, it is easier to check the size of the eigenvalue solutions by *Argand* diagram with complex unit circle. The same requirement applies to cointegrated time-series. Depending on the form of the characteristic equation, stability is satisfied *if all the roots lie either strictly inside or strictly outside the unit circle*, that is either all λi > 1, or equivalently all λi < 1. On the other hand, any roots falling on the unit circle suggest evidence of instability (non-stationarity). See, Appendix on complex roots and unit circle.

Example: Cointegration test for US and EU natural gas prices with a differenced EU natural gas price series regressed on the lagged-level and lagged-differenced UG and EU results in the following



```
Dependent Variable: D(EUR)
Method: Least Squares
Date: 06/18/13   Time: 16:02
Sample: 1995M01 2011M03
Included observations: 195
==========================================================================
        Variable          Coefficient   Std. Error   t-Statistic   Prob.
==========================================================================
           C              -0.029851     0.055699     -0.535936    0.5926
       D(EUR(-1))          0.174384     0.066605      2.618202    0.0096
       D(EUR(-2))          0.215547     0.067002      3.217042    0.0015
       D(EUR(-3))          0.365547     0.070451      5.188640    0.0000
       D(EUR(-4))         -0.268792     0.070664     -3.803828    0.0002
       D(EUR(-6))         -0.145031     0.069668     -2.081738    0.0387
       D(US(-1))           0.044356     0.030122      1.472536    0.1426
        EUR(-1)           -0.030804     0.010529     -2.925750    0.0039
        US(-1)             0.047134     0.014460      3.259640    0.0013
==========================================================================
R-squared                  0.383501   Mean dependent var      0.034923
Adjusted R-squared         0.356984   S.D. dependent var      0.423880
S.E. of regression         0.339902   Akaike info criterion   0.724738
Sum squared resid          21.48924   Schwarz criterion       0.875799
Log likelihood            -61.66193   Hannan-Quinn criter.    0.785901
F-statistic                14.46293   Durbin-Watson stat      1.993794
Prob(F-statistic)          0.000000
==========================================================================
```

The characteristic equation of this model, demonstrated in the Argand diagram below produced by software, suggests a stable (stationary) model because all of the roots fall within the unit circle.



Eigenvalues and Rank

The rank r of a square ($n.\ n$) matrix A is equal to the number of independent rows (or columns):

Rank (A)=r if $r<n$; *A* is a full rank matrix if $rank(A)=n$. Since the determinant is not equal to zero if all rows of A are independent, it follows that the *rank of a matrix is equal to its nonzero eigenvalues*, that is if $|A|=\boldsymbol{n}$, the none of the eigenvalues can be zero. On the other extreme, if |A|=0, then all λ_i must be zero. The intermediate case is when some of the roots are zero and some none zero, $0<rank(A)=r<n$. Using the property of matrix determinant that interchanging its rows (or columns) does not affect its value, we can re-arrange $|A - \lambda I| = 0$ so that the first r rows are of linearly independent, and the remaining (n-r) rows are zero roots.

Applications to Cointegration and Rank in Johansen Procedure

The Johansen procedure rests on the relationship between the rank of matrix and its eigenvalues.

$$\Delta x_t=A_1x_{t-1}\ -x_{t-1}\ +\varepsilon_t=\ (A_1-I)x_{t-1}+\varepsilon t=\pi x_{t-1}+\varepsilon_t$$

The rank of $\pi=(A_1-I)$ is equal to the number of cointegrated vectors; as a result, if $\pi=(A_1-I)=0$, all the $\{x_{it}\}$ processes are unit root and thus not cointegrated, while if $\pi=(A_1-I)=$ n, then all the variables are stationary if we exclude characteristic roots greater than 1 to ensure a convergent system of difference equations. If the rank π=1, then all rows of π can be written as a scalar multiple of the first:

$$\Delta x_{it}=\ (\pi_{11}\ x_{1t-1}+\ \pi_{12}x_{2t-1}+\ldots+\ \pi_{1n}x_{nt-1})+\varepsilon_{it}\ \text{ or}$$

where s_1=1 and $s_i=\ \pi_{ij/}\ \pi_{11}$. Therefore, $(\pi_{11}\ x_{1t-1}+\ \pi_{12}x_{2t-1}+\ldots+\ \pi_{1n}x_{nt-1})\ =(\Delta x_{it}\ \ -\ \varepsilon_{it})/s_i$ because Δx_{it} is I(1) and ε_{it} is a standard normal random error; if rank(π)=r, there are r linearly independent stationary combinations; if rank(π)=n, all variables are stationary. The Johansen test determines the number of the roots significantly different from zero; if all eigenvalues of $A_1$are within the unit circle, then π is of full rank.

Johansen method of calculating eigenvalues

The Johansen method first selects the appropriate number of lags for the VAR model

$$x_t= A_1x_{t-1}+ \mathrm{A}_2x_{t-2}+\ldots+ \mathrm{A}_px_{t-p})+\varepsilon_t$$

First, estimate the VAR in first differences: $\Delta x_t= B_1\Delta x_{t-1}+ B_2\Delta x_{t-2}+\ldots+ B_{p-1}\Delta x_{t-p+1}+e_{1t}$

Second, regress x_{t-1} on a VAR of the form: $x_{t-1}= C_1\Delta x_{t-1}+ C_2\Delta x_{t-2}+\ldots+ C_{p-1}\Delta x_{t-p+1}+e_{1t}$

Third, compute the squares of the n (canonical) correlations between e_{1t} and e_{2t} obtained from the solutions to$|\lambda S_{22} – S_{12}S^{-1}{}_{11}S'_{12}|=0$ where $S_{ij}=T^{-1}\sum_{i=1}^{T} e_{it}(e_{it})'$ and $S_{12}=T^{-1}\sum_{i=1}^{T} e_{2t}(e_{1t})'$; e_{1t} and e_{2t} are column *vectors of residuals in first and second steps.*

Fourth, Obtain the maximum likelihood of the cointegrating vectors from the solutions to

$$\lambda S_{22}\,\pi_{\mathrm{i}}= = S_{12}S^{-1}{}_{11}S'_{12}\pi_i$$

VI imaginary and complex numbers

The quadratic equation x^2=1 has two **real number** solutions, x=1 and x=-1. By contrast no real number satisfies x^2= - 1. However, consider an **imaginary number** i^2= - 1; i can be multiplied using standard rules, e.g. (2i).(3i)=(6)i^2= - 6. This suggests x= -1 is a solution to x^2= - 1:

$(- i)^2=(- 1)^2.(i)^2= - 1$. Therefore, the firs equation has two real number roots (+1; - 1), while the second equation has two imaginary roots (i ; - i). Given any real numbers a & b, then (a+bi) represents a real number if b=0, and an imaginary number if a=0 and b is nonzero. In general, a ***complex number*** is expressed by (a+bi), i.e. such a number has two components: a real number a and an imaginary number bi.

Complex numbers are added and multiplied using the standard algebraic rules:

$$\text{I- } (a_1+b_1i)+(a_2+b_2i)= (a_1+a_2)+(b_1+b_2)i\ ;$$

$$\text{II- } (a_1+b_1i).(a_2+b_2i)= a_1a_2+a_1b_2i + b_1a_2i+b_1b_2\, i^2= (a_1a_2 - b_1b_2)+ (a_1b_2 - b_1a_2)i$$

We usually simplify a complex number by separating its real component (a_1a_2 - b_1b_2), from its imaginary component (a_1b_2 - b_1a_2)i.

Trigonometric representation of complex numbers.

Figure 1 presents a circle with a radius equal to one at the origin of (x, y); (x_0, y_0) and Θ is the angel at this point with the x-axis. The ***sine*** of Θ is defined as the y-coordinate of the point; the ***cosine*** its x-coordinate.

$$sin(\Theta)= y_0 \ \& \ cos(\Theta)= x_0$$

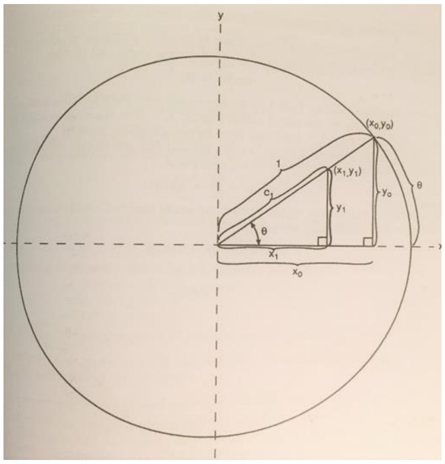


Figure 1 *Graphical Presentation of complex numbers*

Θ is measured in **radians**, that is the counterclockwise distance along the unit circle from the x-axis to the point (x_0, y_0); the circumference of a unit radius circle is 2π; one-quarter rotation around the unit *circle is $\Theta=1/4(2\pi)= \pi/2$ radian, therefore, the 90^0 right angel triangle. As a result, a 45^0 angel is $\pi/4$ radian; a 180^0* has an angle of π radian, etc.

Consider the smaller triangle with vertex (x_1, y_1) that shares the same angle with the original triangle. Then

$$y_1/c_1=y_0/1 \quad \& \quad x_1/c_1=x_0/1 \quad \text{or}$$

$$y_1= c_1.\ sin(\Theta) \ \& \quad x_1= c_1.\ cos(\Theta)$$

Furthermore, c_1 is the Euclidian distance from the origin to (x_1, y_1) point, and given by $c_1=\sqrt{x_1^2 + y_1^2}$. Describing the point in terms of $c_1.\ sin(\Theta)$ & $c_1.\ con(\Theta)$ called its **polar coordinates** of c and Θ.

Properties of Sine and Cosine Functions

The functions *sin(Θ) & con(Θ)*are known as *sinusoidal functions*. Figure 2 illustrates that as a function of *Θ; at zero Sin(0)=0, the function rises to 1 as Θ* rises to *π/2,* then falls to zero as *Θ* rises further to *π* , its minimum is -1 at *Θ= 3π/2* and then starts moving up. Once a distance of 2*π* radians around the unit circle is covered, the function repeats itself. $sin(2\pi + \Theta)= sin(\Theta)$; more generally with j full revolutions

$$sin(2\pi j + \Theta)= sin(\Theta)$$

The sine function is thus periodic; employed in time-series analysis to describe a cycle that repeats itself in a specific cycle. The cosine function starts at 1 and falls to zero as Θ rises to $\pi/2;$ it is a *horizontal* shift of the sine function:

$$cos(\Theta)=sin(\pi/2 + \Theta)$$

With the negative values of Θ *(clockwise rotation), we have*

$$sin(- \Theta)= - sin(\Theta) \qquad \& \qquad cos(- \Theta)= cos(\Theta)$$

A point (x_0, y_0) on the unit circle has $1=\sqrt{x_0^2 + y_0^2}$, and $1= [cos(\Theta)]^2+ [sin(\Theta)]^2$

Trigonometric representation of cycles

Define a function $g(\Theta)=sin(2\Theta)$, i.e. as Θ goes from zero to π, 2Θ goes from zero to 2π, *so* $g(\Theta)$ is back at its initial value. More generally, $sin(k\Theta)$ goes through k cycles in the time it takes for $sin(\Theta)$ to complete a single cycle, see Figure 2.

We can define the values of y at time t as a function of sine and cosine such as

$$y_t=R.\cos(\omega t + \alpha)$$

R is the *amplitude* of the equation, y_t achieves a maximum $+R$ and a minimum of $- R$. α called the *phase*, determines where the cycle y_t would be at t=0. The parameter ω determines how quickly the variable cycles, and it is presented by two measures: the *period* is the duration of time it takes for the process to repeat one full cycle. For example, if $\omega.$ =1, y repeats itself every 2π period; if ω =2, then y repeats itself every π period. The frequency measures how frequently y_t cycles compared to the simple cos(t), that is a measure of the number of cycles completed in 2π periods, for instance, with ω =2, the cycles complete twice as *quickly as those for cos(t). There is a relationship between these two measures of speed of cyclical rotation, namely the period is equal to 2π divided by frequency.*

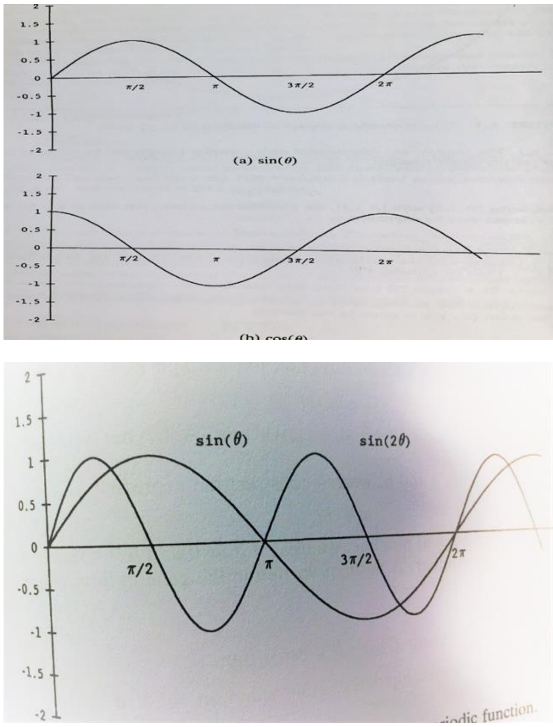


Figure 2 *Sine & Cosine perodic functions and effect of changing their frequency.*

We can preset the complex number (a + bi) in an Argand diagram by Figure 3, with real component (a) on the horizontal axis and the imaginary component (b) on the vertical axis. The size, or the modulus, of a complex number is measured by the distance |a + bi| =$\sqrt{a^2+b^2}$The complex unit circle is the set of all complex numbers whose modulus is unity. In Figure 3, the real number +1 is presented by point A, the imaginary number – *i* by point B, and the complex number (- 0.6 – 0.8*i*) by point C. A key aspect of interest of a complex number is whether its modulus is less than 1, and hence *inside the unit circle*. For example, (- 0.3 + 0.4*i*)= $\sqrt{0.09+0.16}=\sqrt{0.25}$=0.5 but (3 + 4*i*)= $\sqrt{9+16}=\sqrt{25}$=5 is not inside the unit circle. A complex number can be presented by its modulus R=$\sqrt{a^2+b^2}$ and the angel of *Θ it makes with the real axis as measured by*

$$\cos(\Theta)=a/R \qquad \& \qquad \sin(\Theta)=b/R$$

Hence, a complex number in **polar coordinates form** is written as

$$[R.\cos(\Theta)+i.R.\ \sin(\Theta)]=R[\cos(\Theta)+i.\sin(\Theta)]$$

The complex conjugate of (a + bi) is (a - bi); the two numbers are *conjugate pairs*. adding a conjugate pairs results in a real number (equal to 2a), while their product is also a real number (a^2 +b^2). Finally, the modulus of a complex number is equal to the square root of its conjugate pairs:

$|a + bi| = \sqrt{(a+ib).(a-ib)}.$

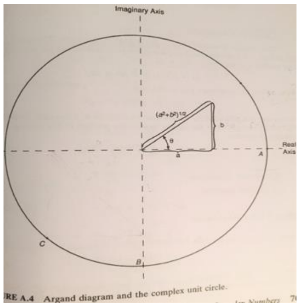


Figure 3-*Unit circle Presentation of Complex root eigenvalues*

Stability of a time-series, namely, convergence to its long-run path, requires testing for the size of the characteristic solutions some of which may be complex, some real. The size of the complex well as real characteristic roots can easily be checked with Argand diagram. Suppose an *AR*(*p*) model

$$Y_t = \lambda_1 y_{t-1} + \lambda_2 y_{t-2} + \ldots + \lambda_p y_{t-p} + \varepsilon_t \text{ or } (y_t - \lambda_1 y_{t-1} - \lambda_2 y_{t-2} - \ldots - \lambda_p y_{t-p}) = \varepsilon_t$$

This model is dynamically stable if all of the z roots of its characteristic equation obtained from solving

$$(1 - \lambda_1 z - \lambda_2 z^2 - \ldots - \lambda p z^p) = 0$$

lie strictly ***outside*** *the unit circle*. That is, for any eigenvalue solution z substituted in the above condition, the sum of the terms to the right of 1 must be less than 1. It must be pointed out that this condition can also be stated in terms of the roots of the *inverse characteristic equation*, i.e. expressed in terms of the lag operator coefficients of the series, see section 5.3. That is, the condition can be equivalently stated that all eigenvalues of

$$(z^p - \lambda_1 z^{t-1} - \lambda_2 z^{t-2} - \ldots - \lambda_p) = 0$$

lie strictly ***inside*** *the unit circle.* For example, for an *AR*(2) model, dynamic stability (stationarity) requires that either all roots of the quadratic equation $(1 - \lambda_1 z - \lambda_2 z^2)=0$ lie strictly outside the unit circle; or all roots of the quadratic equation $(z^2 - \lambda_1 z - \lambda_2)=0$ lie strictly inside the unit circle, see

Q 6.1 exercise.

Geometric series approximations for Rational functions

A *rational function* expresses a *y* variable as a ratio of two polynomials in an *x* variable; for example, $y=\frac{x-1}{x^2+2x+4}$. This definition suggests that any polynomial can always be a rational function expressed as a ratio to 1, e.g. as $y=\frac{1}{x^2+2x+4}$. In time-series analysis, when the polynomials represent to infinite series of lags over time, we can obtain good approximations for such series by expressing them as rational function that provide the basis for the *rational distributed lag model*

$$y_t = \alpha\ x\frac{\beta(L)}{\lambda(L)}x_t + \varepsilon_t$$

where $\beta(L)$ & $\lambda(L)$ typically stand for lags distributed over *t* of y_t and ε_t , written in the lag operator; such a model is parsimoniously effective when the lag series are expressed in terms of their approximate functions.

Let us examine the relationship between an infinite series and its approximation. Consider f(x)= $\frac{1}{1-x}$. If x ≠ o, then $\frac{1}{1-x} = 1 + \frac{1}{1-x}$, hence the approximation error is 1. Multiplying the above by x results in $\frac{1}{1-x} = x + \frac{x^2}{1-x} = 1 + x + \frac{x^2}{1-x}$, hence $\frac{1}{1-x} = 1 + x$, and the approximation error is $\frac{x^2}{1-x}$**.** Multiplying the above by x results in $\frac{1}{1-x} = 1 + x + x^2 + \frac{x^3}{1-x} = 1 + x + x^2 + \frac{x^3}{1-x}$, so $\frac{1}{1-x} = 1 + x + x^2$, and the approximation error is $\frac{x^3}{1-x}$; etc. Therefore, we can approximate the sum of an infinite series $(1+x+x^2+x^3+\ldots)$ by $\frac{1}{1-x}$, that is as a rational function of the series itself, subject to the sum of the degrees of approximation errors at each step. Since the series is defined by a common ratio of *x*, it is a geometric series, and the key requirement for approximating a geometric series by the series rational function itself is for $|x|< 1$, otherwise the approximation method does not work, i.e. the series diverge and explode. However, provided $|x|< 1$, the approximation will be small, for example if $x=0.1$, then approximating for $(1+x+x^2+x^3)$ by $\frac{1}{1-x}$ results in

$[(0.1)^4/0.9]*100\%=0.01$ percent; if on the other hand $x > 1$ the approximation will no longer be useful.

Selected References

Arellano, M and S. Bond (1991), “Some tests of specification for panel data: Mon Carlo evidence and an application to employment equations”, *Review of Economic Studies*, **58**:277-297.

Arellano, M and O. Bover (1995), “Another look at the instrumental variable estimation of error-component models”, *Journal of Econometrics*, **68**: 29-51.

Blundell, R and S. Bond (1998), “Initial conditions and moment restrictions in dynamic panel data models”, *Journal of Econometrics*, **87**: 115-143.

Breiman L (2001). "Random Forests". *Machine Learning*. **45** (1): 5–32

Chan, S.W., W.K. Li, and H. Tong (eds.), *Statistics and Finance: An Interface*. London: Imperial College Press.

Chan F. and L. Ma’ta’s (2022) “ Linear Econometric Models with Machine Learning”, in

Chan F. and L. Ma’ta’s (ed.s), *Econometrics with Machine Learning*, New York: Springer

Colin Cameron, A. and P. K. Trivedi (2022), *Microeconometrics Using Stata*, 2ed ed. Volume I: Cross-sectional and Panel Regression Methods; Volume II*: Nonliner Models and Causal Inference Methods*, Txas: Stata Press

Cameron A, and P Trivedi (2005), *Microeconometrics*, Cambridge, Cambridge.U.P.

Cameron, A, P. Trivedi, Milne, and J. Piggott, “A Micoeconometric Model of the Demand for Health Care and Health Insurance in Australia”, *Review of Economic Studies* **55**: 85-106.

Casella, G and E. George (1992), “Explaining the Gibbs Sampler”, *the American Statistician*, **46**: 167-174.

Chatfield, C. and Xing, H. (2019), *The Analysis of Time Series*, 7th edition, Boca Raton, FL: CRC Press.

Commandeur, J. and Koopman, S.J. (2007), *An Introduction to State Space Time Series Analysis*, Oxford: Oxford U.P.

Diebold, F X (2006), *Elements of Forecasting*, 4th edition, Mason, OH: South-Western,

Efron, B (1979), “Bootstrapping Methods: Another Look at the Jackknife”, *Annals of Statistics* **7**:1-26.

Efron, B and J. Tibsharani (1993), *An Introduction to the Bootstrap*, London, Chapman and Hall.

Enders, W. Applied (2015), *Econometric Time Series*, 4th edition, John Wiley and Son, NJ.

Engle, R. (1982), “Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation”, *Econometrica* **50**: 987-10078.

Engle, R and C. Granger (1987), “Co-integration and error correction: representation, estimation and testing”, *Econometrica* **55**: 251-276.

Garthwait, p, J Kadane, and A. O’Hagan (2005), “Statistical Methods for Eliciting Probability Distributions”, *Journal of the American Statistical Association* **100**: 680-700.

Geweke, J (1989), “Bayesian Inference in Econometric Models Using Monte Carlo Integration”, *Econometrica* **38:** 73-89.

Geweke J and S Poter-Hudak (1983), “The Estimation and Application of Long Memory Series Models”, *Journal of Time Series Analysis* **4**: 2211-237

Gong, X, A. von Soest, and P. Zhang (2005), “The Effects of Gender of Children on Expenditure Patterns in Rural China: A Semiparametric Analysis”, *Journal of Applied Econometrics*, **20**: 509-527.

Gonzalez-Rivera, G. (2013), *Forecasting for Economics and Business*, NJ: Pearson.

Granger, C. (1969), “Investigating causal relationships by econometric models and cross-spectral methods”, *Econometrica* **37**: 424-438.

Granger C. and M.H. Pesaran (2000), “A decision-based approach to forecasting evaluation”, in

Granger, C. and J Joyeux (1980), “An Introduction to Long Memory Time Series Models and Fractional Differencing”, *Journal of Time Series Analysis*, **1**: 15-29.

Greenberg, E (2014), *Introduction to Bayesian Econometrics*, 2th Edition, NY: Cambridge U.P.

Hamilton, J. (1994), *Time Series Analysis,* NJ: Princeton U P.

Hastie, T, R. Tibshirani, J. Friedman (2001), *The Elements of Staristical Learning*, New York: Springer

Heckman, J (1976), “The Common Structure of Statistical Models of Truncation, Sample Selection and Limited Dependent Variables and a Simple Estimator for Such Models, *Annals of Economic and Social Measurement* 5: 475-492.

Im, K., M H Pesaran and Y. Shin (2003), “Testing for unit roots in heterogenous panels”, *Journal of Econometrics* **115**: 53-74.

James, G., D. Witten, T. Tibshirani, (2021), *An Introduction to Ststistical Learnig, with Applications in R,* 2ed ed., New York: Springer.

Johansen S (1988), “Statistical analysis of cointegration vectors”, *Journal of Economic Dynamics and Control* **12**:231-254.

Kalman, R (1960), “A new approach to linear filtering and prediction problems”, *Journal of Basic Engineering Transactions ASMA, Series D* **82**:35-45

Kenan, J. (1985), “The Duration of Contract Strikes in U.S. Manufacturing”, *Journal of Econometrics* **28**: 5-28.

Koohi-Kamali, F (2019), “The Rothbarth Internal Allocation Model Re-Examined: Semiparametric and Parametric Tests of Child Gender Discrimination”, https://4a7f2b3d-4863-4989-a89c-2965592c1873.filesusr.com/ugd/95f6d9_12e684c8035c4c5e8be5724491b04df9.pdf

McCall, B (1996), “Unemployment Insurance Rules, Joblessness, and Part-Time Work”, *Econometrica* **64**: 647-682.

Mullainathan, S. and J. Spies, “Machine Learning: A, applied Economtric Approoach”, *Journal of Economic Perspectives*, **31**(2): 87-106.

Pesaran, M H (2015), *time series and panel data econometrics*, Oxford: Oxford U. P.

Pesaran, M H, Y. Shin, and R. Smith (2001), “Bounds testing approaches to the analysis of level relationships”, *Journal of Applied Econometrics* **16**: 289-326.

Pesaran, M H, Y. Shin, and R. Smith (1999), “Pooled mean group estimation of dynamic heterogenous panels”, *Journal of the American Statistical Association* **94**: 621-634.

Phillips, A. (1954), “Stabilisation policy in a closed economy”, *Economic Journal*, **64**: 290-232.

Robinson, P (1995), “Log-Periodogram Regression of Time Series with Long Range Dependence”, *The Annals of Statistics* **23**: 1048-1072.

Robinson, P (1988), “Root-N-Consistent Semiparametric Regression”, *Economtetica* **56**: 931-954.

Tabshirani, R. (1996),“ Regression Shrinkage and Selection via the Lasso”, *Journal of the Royal Statistical Society: Series B (Methodological)*, **58**(1): 267–288, https://doi.org/10.1111/j.2517-6161.1996.tb02080.x

Tobin, J (1958), “Estimation of Relationship for Limited Dependent Variables”, *Econometrica* **26**: 24-36.

Sowell, F (1992), “Maximum likelihood estimation of stationary univariate fractionally integrated time series models”, *Journal of Econometrics* **53:** 165-188.

Silverman, B. (1986), *Density Estimation for Statistics and Data Analysis*, London: Chapman and Hall.

Wooldridge, J. (2010), *Econometric Analysis of Cross Section and Panel Data*, 2th edition, Cambridge: MIT Press.