MAXIMUM LIKELIHOOD FOR POLITICAL ANALYSIS

Theory and Applications

POLI 787 Lecture Notes

Marco R. Steenbergen

Department of Political Science
University of North Carolina at Chapel Hill

Contents

List of Figures

List of Tables

List of Abbreviations

CDF	Cumulative density function
LPM	Linear probability model
LR	Likelihood ratio test statistic
LM	Lagrange multiplier test statistic
ML	Maximum likelihood
MLE	Maximum likelihood estimation
N	Normal distribution
PDF	Probability density function
VCE	Variance-covariance matrix of the estimator(s)
W	Wald test statistic

List of Symbols

$\boldsymbol{\nabla}$	Gradient
ϵ_i	Error or disturbance term
λ	Standard logistic probability density function
Λ	Standard logistic distribution function
ϕ	Standard normal probability density function
Φ	Standard normal distribution function
θ	Parameter
$\boldsymbol{\theta}$	Parameter vector
Θ	Uni-dimensional parameter space
$\boldsymbol{\Theta}$	Multi-dimensional parameter space
$\mathcal{E}$	Expected value
$\boldsymbol{\mathcal{H}}$	Hessian
H_0	Null hypothesis
$\boldsymbol{\mathcal{I}}_e$	Expected Fisher information matrix
$\boldsymbol{\mathcal{I}}_o$	Observed Fisher information matrix
f	Probability density or mass function
ℓ	Log-likelihood function
ℓ'	First derivative of the log-likelihood (single parameter)
ℓ''	Second derivative of the log-likelihood (single parameter)
$\mathcal{L}$	Likelihood function
$\mathbf{V}$	Variance-covariance matrix of the estimator
$\boldsymbol{x}_i$	Vector of covariates/predictors
y_i	Response variable

List of Stata Commands

The following is a list of Stata commands that will be used throughout this report. Commands marked with an asterisk are not part of the standard Stata installation and will have to be downloaded from the Internet.[1] Please consult the Stata manuals and help files for detailed descriptions of each command.

`bootstrap`	Bootstrap confidence intervals	Chapter 7.4
`cloglog`	Binary response variables	Ch. 9
`denormix`*	Mixture models	Chapter 5.10
`estat bootstrap`	Bootstrap confidence interval	Chapter 7.4
`estat vce`	General ML programming	Chapter 5.9
`fitstat`	Model fit	Chapter 8.3
`gologit2`*	Ordinal response variables	Ch. 10
`hetprob`	Binary response variables	Ch. 9
`logit`	Binary response variables	Ch. 9
`lrtest`	Hypothesis testing	Chapter 6.1
`mfx`	Categorical response variables	Chapters 9-11
`ml check`	General ML programming	Chapter 5.9
`ml graph`	General ML programming	Chapter 5.9
`ml init`	General ML programming	Chapter 5.9
`ml max`	General ML programming	Chapter 5.9
`ml model`	General ML programming	Chapter 5.9
`ml plot`	General ML programming	Chapter 5.9
`ml search`	General ML programming	Chapter 5.9
`oglm`*	Ordinal response variables	Ch. 10
`ologit`	Ordinal response variables	Ch. 10
`oprobit`	Ordinal response variables	Ch. 10
`probit`	Binary response variables	Ch. 9
`program`	General ML programming	Chapter 5.9
`scobit`	Binary response variables	Ch. 9
`switchr`*	Switching regression models	Ch. 5.10

[1] These programs can be downloaded from within Stata by typing into the command line `net search` followed by the program name. This will show a series of URLs. By clicking on the appropriate link and following the instructions, the program will be installed onto your computer.

`test`	Hypothesis testing	Chapter 6.1

List of Stata Programs

The following Stata programs are available on the course website and can be used to reconstruct the examples in this report.

`bivnorm.do`	Example 5.4
`example 5.9.do`	Example 5.9
`example 5.12.do`	Example 5.12
`Example 5.13.do`	Example 5.13
`Example 6.6.do`	Example 6.6
`Example 6.7.do`	Example 6.7
`Example 7.5.do`	Examples 7.5-7.6
`Example 7.7.do`	Example 7.7
`myratio.do`	Examples 7.5-7.7
`normal.do`	Example 5.3
`poisson.do`	Example 5.2
`weibull2.do`	Example 5.10

List of Data Sets

The following data sets are available on the course web site and can be downloaded to reconstruct the examples.[2]

`Example 5.8.dta`	Randomly generated
`Example 5.10.dta`	Randomly generated
`Example 5.12.dta`	2004 NES (pre-election)
`Example 5.13.dta`	2000 NES
`Example 7.5.dta`	2004 NES (pre-election)

[2]Legend: NES = (American) National Election Studies.

Part I

The Theory of Maximum Likelihood

Chapter 1

Introduction

1.1 History and Significance of MLE

Maximum likelihood estimation (MLE) provides a versatile estimation procedure with many desirable properties.[1] In terms of versatility, it is safe to say that many of the statistical methods that are now commonplace in political research would not be possible without MLE. These methods often rely on complex functions that are not linear in the parameters and hence cannot be estimated using ordinary (OLS) or generalized least squares (GLS). Without the benefit of MLE, it would not be possible to perform logit and probit analysis, sample selection modeling, event count modeling, or event duration modeling, to name just a few applications of MLE that are now commonly found in the political science literature.

In terms of desirable properties, one of the greatest benefits of MLE is that it provides a unified framework for estimation and hypothesis testing, the two central components of statistical inference (see King 1989). That is to say, when we estimate a model using MLE we also obtain all of the ingredients that are required to perform a proper hypothesis test, including test statistics and their (asymptotic) distributions. Moreover, it can be demonstrated that any ML estimator is (asymptotically) consistent, efficient, and normally distributed. All of these are highly desirable properties, although I should stress the proviso "asymptotically" because the small sample properties of MLE are often unknown or less desirable.

[1] Throughout these notes, I shall refer to maximum likelihood estimation as MLE. The term "maximum likelihood" will be abbreviated as ML.

Figure 1.1: Sir Ronald A. Fisher (1890-1962)

Applications of MLE in political science are relatively young, dating back to the 1980s with most of the growth occurring since the 1990s. The theory of MLE, however, is actually quite old. It dates back to a series of papers published by the British geneticist and statistician **Sir Ronald A. Fisher** between 1912 and 1922 (Fisher 1912, 1921, 1922).[2] Herein, Fisher gave three different rationales for MLE, at the same time introducing a number of other important statistical concepts such as efficiency, information, and sufficiency. The term "maximum likelihood" appeared only in the 1922 article, although Fisher had used the term "likelihood" already in the 1921 paper.

Part of why there was such a long delay between the theoretical development of MLE and its practical application is that, for all but the simplest problems, ML estimates cannot be found analytically. As we shall see in the next chapter, MLE requires computation of the first derivative of the log-likelihood function; by setting this derivative equal to zero, ML estimators can be derived. This is quite simple to do when the log-likelihood function is linear in the parameters but it quickly becomes an untractable problem when the log-likelihood function is more complex. In that case, estimates will have to be found computationally, through the use of numerical optimization methods. The computer power and algorithms required for numerical optimization have become available only relatively recently. This happened first for relatively simple log-likelihood functions such as those encountered in logit and probit analysis. For other log-likelihoods, especially those involving multiple integrals (e.g. multinomial probit analysis), numerical optimization has become feasible only in the last decade or so.

[2]The first paper was published while Fisher was a third year undergraduate student at Cambridge University (see Aldrich 1997; Box 1978).

The availability of ever more efficient algorithms and ever more powerful computers—PCs that is, instead of the mainframes that were once required to perform computations—mean that MLE has become readily available to political scientists. Statistical software packages such as Stata include a large number of "canned" routines for performing MLE, not to mention that they also provide relatively simple procedures for programming one's own likelihood functions. This means that a wide range of methods has become available that are now considered standard when ten or fifteen years ago they would have been deemed esoteric. The current expectation in the discipline is that scholars should take advantage of these methods, especially since they allow political scientists to build more complex models and models that reflect the data generating mechanisms better than classical linear regression analysis could. Gone, then, are the days that knowing OLS was sufficient to get by as a political scientist. To engage in competent political analysis today, it is essential that one knows and understands methodological tools that are firmly rooted in MLE.

This report is an effort to teach you the theory and application of MLE. The first part focuses on theory. While it is the more abstract and purely statistical portion of this report, it provides an essential foundation for understanding the applications discussed in Part II. Without a solid understanding of how MLE works in the abstract, the nuances of the applications cannot be understood.

The selection of applications discussed in the second part does not pretend to be exhaustive. Rather, it is a purposefully chosen subset of those applications that (1) appear frequently in the political science literature and (2) are not covered in other courses taught at UNC. The second criterion means, for example, that no time series applications of MLE are discussed, even though MLE figures prominently in time series econometrics. It also means that the current report foregoes a discussion of covariance structure models (also known as LISREL models), although again much of the statistical development of these models rests on MLE. The applications that will be discussed include: models for binary dependent variables, models for ordinal dependent variables, probabilistic choice models, models for censored and truncated variables, sample selection models, event count models, event duration models, and multilevel models.

In describing the different applications, four elements are highlighted: (1) the underlying behavioral model; (2) the translation of this behavioral model into a statistical model; (3) estimation of this statistical model; and (4) in-

terpretation of the results. The focus on the first two elements is important because it shows the important role that assumptions play in MLE and its applications. The emphasis on estimation reveals how one can obtain estimates of model parameters from data. In discussing model estimation, I shall focus exclusively on Stata as a programming platform.[3] Finally, interpretation is crucial because many of the applications that we shall discuss are distinctively more complex than the familiar linear regression model, which has a straightforward meaning, and because interpretation is key to our jobs as political scientists. Again, Stata will figure prominently in this part of the discussion.

It is my hope that the materials covered in this report will provide useful background information to other course materials and will help you to understand MLE. MLE has revolutionized political analysis, its versatility making possible what was considered untractable only a few decades ago. I hope these notes will help you to appreciate the impact that MLE has had and continues to have on political analysis. For, even though political analysis has taken a distinctly Bayesian turn in recent years, any good Bayesian will tell you that likelihoods still play a critical role in their model of inference. Thus MLE truly is the cornerstone of modern political analysis and, for this reason alone, understanding it is an essential part of your graduate training as a political scientist.

1.2 How to Use These Notes

This report contains several sections that are marked with an asterisk. These are optional sections. We shall not cover them in class but they provide useful extensions to the material that is covered. If your objective is to obtain a comprehensive understanding about MLE and its applications, then it is recommended that you read the optional sections. If your goal is just to understand the basics, then the optional sections may be skipped.

Accompanying this report are several electronic resources that will be made available through the course web site. First all of the Stata syntax files

[3]This focus is motivated by two rationales. First, Stata contains a wide range of MLE applications and also provides an easy framework for programming one's own ML estimators. Second, most students already have considerable familiarity with Stata, thus making the learning curve less steep. It should be noted, however, that many alternatives to Stata exist, including Gauss, Limdep, R, SAS, and S-Plus.

referenced in these notes are available to you. Second, the data sets used in the examples are available to you for purposes of replication. A list of syntax and data files can be found at the beginning of this syllabus.

Chapter 2

Estimation of Single Parameters

MLE is one of a number of different estimation methods that are used in statistics and applied data analysis. The general character of estimation problems is that we have one or more parameters that describe the distribution of one or more variables in the population; these parameters cannot be directly observed. Using data from a sample drawn from this population, our goal is to obtain estimates of the parameters. Estimation methods outline principles that allow us to generate formulae for assembling data into estimates.

There are several approaches to the estimation problem. In POLI 784, you have already encountered ordinary (OLS) and generalized least squares (GLS). Econometricians rely heavily on the (generalized) method of moments to solve estimation problems. In recent years, political scientists have begun to estimate parameters using methods of Bayesian inference. Compared to the familiar OLS and GLS methods, and to a lesser extent method of moments estimation, MLE offers a number of distinct advantages. First, MLE is theoretically grounded in that it requires an explicit model of the data generation process. Second, under general conditions, maximum likelihood estimators have desirable asymptotic properties such as consistency and efficiency. Third, MLE is extremely versatile and can be used to estimate a much wider range of models than can be handled by OLS or GLS. Finally, MLE provides a unified approach to estimation and hypothesis testing.

But what exactly is MLE? To understand its logic it is useful to sketch out some intuitions about the relationships between parameters and samples,

in particular how changing the value of a parameter is likely to change the nature of the sample that one draws.

2.1 The Intuition behind Maximum Likelihood

Imagine a probability density function (PDF—continuous case) or probability mass function (discrete case) with a parameter θ. The function describes the distribution of a random variable in the population; θ is some theoretically interesting attribute of that distribution. Now imagine that θ is given; we know it with certainty. Then we have some ability to predict the data that we would observe in a sample of size n. Put differently, given a particular value of θ, certain values are more likely to be sampled from the distribution than others. Example 2.1 illustrates this intuition.

> **Example 2.1.** Consider a binomial distribution with parameter $\pi = .5$. We draw a sample of $n = 2$ observations. The probability of observing $y = 0$ successes in this sample is .25, while the probability of observing $y = 1$ success is .50, and the probability of observing $y = 2$ successes is .25. Clearly, it is more likely to obtain a sample with a single success than one where none or all of the sample units succeeded. Thus, $y = 1$ is the most likely outcome given $\pi = .5$.

The logic so far is a straightforward application of the basic laws of probability. Here I have treated the parameter as given and the data as the thing we seek to predict and about which there is uncertainty. Of course, the statistical inference problem is precisely the opposite: we know our data but we do not know the parameter. But the underlying intuition can be easily reversed. That is, taking our data as given, certain values of the parameter are more likely to have given rise to this data than other values.The parameter value that is most likely to have produced the data is accepted as the estimate of the parameter. Example 2.2 illustrates how this logic works.

> **Example 2.2.** Consider a sample of size $n = 2$. We know that this sample has been drawn from a binomial distribution. We also know that it includes one success ($y = 1$). Two researchers debate the value of π that gave rise to this data. According to researcher A, $\pi = .5$ is the parameter value that gave rise to the sample. Researcher B

believes that $\pi = 1$ is the right estimate of the parameter value. Who would you believe?

The answer is that A is more likely to be correct. Under the binomial distribution, the expected number of successes is $n\pi$. Under A's estimate of π this produces one success in a sample of two. Under B's estimate of π we should expect to see two successes. The reality is that we observe one success, which is consistent with A's estimate of π but less so with B's estimate. Thus $\pi = .5$ is the parameter value that is most likely to have given rise to the data.

The language used here is instructive. We are not saying that the binomial parameter is .5 with certainty. Even when π is different from .5, it might be that by chance we have drawn a sample where $y = 1$. But of all the possible guesses/estimates of π, .5 is *most likely* to have given rise to the data at hand. And an estimate of .5 is certainly a great deal more likely to have produced one success in a sample of two observations than an estimate of 1.

This process of working backwards from the data to an estimator is called MLE. More formally:

> ***Definition 2.1.*** A ML estimator, $\hat{\theta}$, of a parameter θ produces an estimate such that the probability density function or probability mass function corresponding to this estimate is most likely to have generated the data.

Note the "hat" on top of the parameter. It is conventional to denote estimators and estimates in this manner, so as to distinguish them clearly from the parameter. So in the previous example, we could have denoted the ML estimate as $\hat{\pi} = .5$.

With the definition in place, the question is now how one finds ML estimates other than by way of trial and error, as was done in Example 2.2. It is to this question that we turn next. In the process of answering this question, things will get a lot more mathematical. Underlying this mathematical complexity, however, the same simple intuition remains.

2.2 The Calculus of Maximum Likelihood

The intuitions of MLE should be formalized if we want the estimation procedure to be of practical use. To do so, we start by making two *assumptions*:

1. The population is characterized by a probability mass or density function that is known, up to the parameter(s) that has/have to be estimated.

2. The sample consists of n independent draws from the probability mass or density function. In other words, the data are generated through *random sampling* so that the observations in the sample are independent. More precisely, any dependencies between the observations are captured through one or more parameters. Thus, the observations are conditionally independent—conditional, that is, on the parameters of the probability mass or density function.

Taken together, these two assumptions imply that the observations in the sample are i.i.d.—identically and independently distributed.

2.2.1 The Likelihood and Log-Likelihood Functions

Based on these assumptions, we can now describe the likelihood of obtaining the data. We can then work out the estimator that is most likely to have produced this data. Let $f(y|\theta)$ denote the probability density or mass function, which contains a single parameter θ. Further, let D denote the data, consisting of n independent observations. Then the following holds.

Definition 2.2. The **likelihood function** is given by:

$$\begin{aligned} \Pr(D) &= f(y_1, y_2 \cdots y_n|\theta) \\ &= f(y_1|\theta)f(y_2|\theta)\cdots f(y_n|\theta) \\ &= \prod_{i=1}^{n} f(y_i|\theta) \qquad (2.1) \\ &= \mathcal{L}(\theta|D) \end{aligned}$$

(The second equation in Definition 2.2 follows from the independence of the observations.) Here $\mathcal{L}(\theta|D)$, or $\mathcal{L}$ for short, stands for the likelihood function and is sometimes also abbreviated as L.[1]

[1]Throughout these notes I shall assume that the data are i.i.d. However, a more general statement of the likelihood function is: $\mathcal{L}(\theta|D) = c(D)f(D|\theta)$, where $c(D)$ is a constant that depends on the data but not on θ. This constant allows $\mathcal{L}$ to remain invariant under one-to-one transformations of the data (see Chapter 4). This rendition of the likelihood function applies to any sampling design, including those in which the observations are not independent.

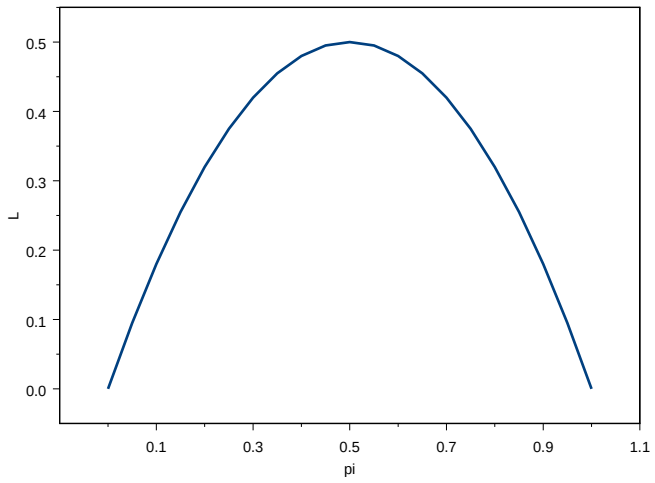

Figure 2.1: Binomial Likelihood Function

Equation (2.1) express the so-called **likelihood principle**, which states that the information in any sample can be found, if at all, from the likelihood function. This principle captures the intuition that I started out with and provides the central foundation of MLE.

> **Example 2.3.** Consider the binomial problem from Example 2.2. Imagine that we observe $y = 1$, $n = 2$. For the binomial distribution, the likelihood function is the same as the distribution, i.e.
>
> $$\mathcal{L} = \binom{n}{y}\pi^y(1-\pi)^{n-y}$$
>
> Figure 2.1 plots this likelihood for different values of π (and setting $n = 2$ and $y = 1$). We see that the likelihood function reaches its apex at $\hat{\pi} = .5$, which means this is the MLE. This supports the intuition outlined in Example 2.2.

In practice, the likelihood function is transformed into a log-likelihood function, which is equal to the natural logarithm of $\mathcal{L}$.

Definition 2.3. The **log-likelihood function** is given by

$$\begin{aligned} \ln \mathcal{L}(\theta|D) &= \ln \left[\prod_{i=1}^{n} f(y_i|\theta) \right] \\ &= \sum_{i=1}^{n} \ln f(y_i|\theta) \\ &= \ell(\theta|D) \end{aligned} \tag{2.2}$$

Here $\ell(\theta|D)$, or ℓ for short, is the log-likelihood, which is sometimes also abbreviated as LL or lnL. This function is used in lieu of $\mathcal{L}$ because working with sums is easier than working with products (e.g. computers handle summation much better than multiplication). Moreover, since the likelihood function is essentially a product of probabilities in the discrete case, its numeric value can be extremely low, which can give problems with numeric precision in computers. The log-likelihood function bypasses this problem since logarithms of fractions are larger numbers.

2.2.2 Optimizing the Log-Likelihood

The likelihood function gives the likelihood of the data. To find the MLE of θ, we should maximize this function.

Definition 2.4. Let $\hat{\theta}$ be the **maximum likelihood estimator** then

$$\mathcal{L}(\hat{\theta}) = \sup_{\theta \in \Theta} \mathcal{L}(\theta) \tag{2.3}$$

where sup stands for supremum (the maximal element in a set of real numbers) and Θ is the *parameter space*, i.e. a non-empty set of feasible values of the parameter.

For the reasons described above, in practice we optimize the log-likelihood function rather than the likelihood function; both will produce identical estimates.

Finding the maximum of the log-likelihood function involves three steps common to all optimization procedures:

1. Take the first derivative of ℓ with respect to θ:

$$\ell' = \frac{d\ell}{d\theta}$$

2. Set this derivative equal to 0:

$$\ell' = 0$$

   This is called the **log-likelihood equation**. It captures the **first-order condition** for finding a maximum. When we solve for θ we obtain the ML estimator, $\hat{\theta}$.

3. Verify that $\hat{\theta}$ is a maximum by evaluating the second derivative of ℓ at $\hat{\theta}$:

$$\ell" = \frac{d^2\ell}{d\theta d\theta}$$

   If $\ell" < 0$ at $\hat{\theta}$, then $\hat{\theta}$ is the ML estimator. This second derivative test is sometimes known as the **second-order condition** for finding a maximum.

It is important to dwell a little on the first and second derivatives. The first derivative captures the **gradient** of the log-likelihood function. As stated by the log-likelihood equation, a (local) maximum occurs where this gradient is zero (assuming the second-order condition is also met). For simple problems, such as those considered in this and the next chapter, the log-likelihood equation can be solved algebraically. In reality, most log-likelihood equations are too complex to be solved in this way. In those cases, numerical optimization methods are used to obtain solutions, a topic that will be discussed in greater detail in Chapter 5.

The second derivative influences the **curvature** of the log-likelihood function. More specifically, $-\ell''$ captures the curvature. The larger ℓ'' is the stronger the peak of the distribution is. In general this is desirable, as it allows for more precise estimation. The quantity $-\ell''$ is known as the observed Fisher information.

> ***Definition 2.5.*** The **observed Fisher information** is equal to negative one times the second derivative of the log-likelihood, i.e. $-\ell''$.

We now turn to a number of examples of log-likelihood functions and show the mechanics of their optimization. It should be stressed that these are relatively simple examples where the estimator can be found by algebraic means. These examples illustrate well the logic of MLE but the practice more typically relies on numerical optimization, as discussed previously.

Example 2.4. Consider the Poisson distribution

$$f(y|\mu) = \frac{\mu^y \exp(-\mu)}{y!}$$

where $\mu > 0$ is the parameter that we want to estimate. Imagine that we have drawn n independent observations from this distribution, then the likelihood function is given by

$$\begin{aligned}\mathcal{L} &= \frac{\mu^{y_1} \exp(-\mu)}{y_1!} \times \frac{\mu^{y_2} \exp(-\mu)}{y_2!} \times \cdots \times \frac{\mu^{y_n} \exp(-\mu)}{y_n!} \\ &= \frac{\mu^{\sum_i y_i} \exp(-n\mu)}{\prod_i y_i!}\end{aligned}$$

The log-likelihood function is

$$\begin{aligned}\ell &= \sum_i y_i \ln(\mu) - n\mu - \ln\left(\prod_i y_i!\right) \\ &= \sum_i y_i \ln(\mu) - n\mu - \sum_i \ln(y_i!)\end{aligned}$$

The optimization of ℓ proceeds as follows. First, we take the first derivative

$$\begin{aligned}\ell' &= \frac{d\ell}{d\mu} \\ &= \frac{\sum_i y_i}{\mu} - n\end{aligned}$$

Next we set this derivative equal to 0 and solve for μ:

$$\begin{aligned}\frac{\sum_i y_i}{\mu} - n = 0 &\Leftrightarrow \\ \frac{\sum_i y_i}{\mu} = n &\Leftrightarrow \\ \hat{\mu} = \frac{\sum_i y_i}{n} &\Leftrightarrow \\ \hat{\mu} = \bar{y}\end{aligned}$$

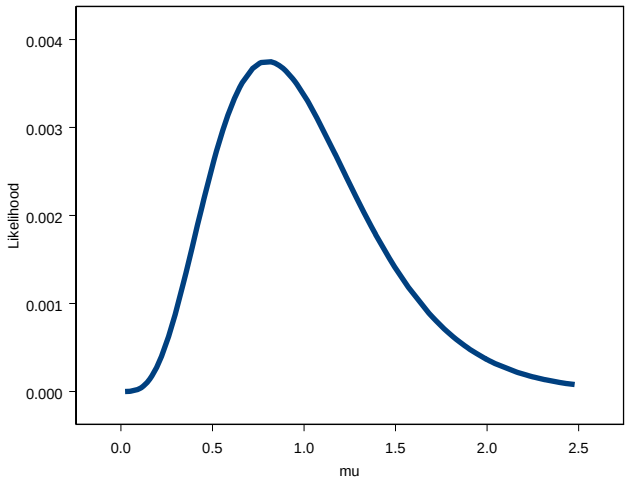

Figure 2.2: Poisson Likelihood Function

This means that the sample mean is the MLE of μ. Before we accept this, however, we need to make sure that the sample mean produces a maximum for ℓ. Taking the second derivative gives

$$\ell'' = -\frac{\sum_i y_i}{\mu^2}$$

We know that μ^2 is positive for any feasible estimate of μ. Since y can take on only non-negative values in the Poisson distribution, it follows that the numerator of the expression is positive as long as there is one non-zero value of y, which is almost always true. Thus, ℓ'' itself is negative, which means that the necessary condition for a maximum is satisfied.

An example can illustrate that $\bar{y}$ is the MLE. Consider a sample of the following 5 observations: 0, 0, 1, 1, 2. The likelihood function is given by

$$\mathcal{L} = \frac{\mu^4 \exp(-5\mu)}{0!0!1!1!2!}$$

Further, the log-likelihood function is given by

$$\ell = 4 \ln \mu - 5\mu - \ln(0!0!1!1!2!)$$

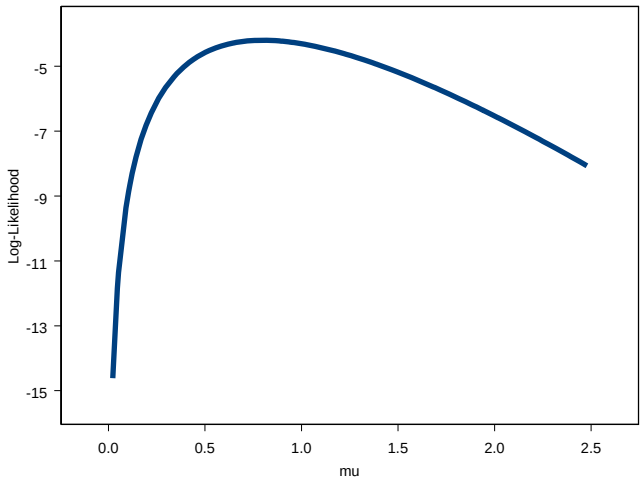

Figure 2.3: Poisson Log-Likelihood Function

We can plot $\mathcal{L}$ and ℓ for different values of μ, as is done in Figures 2.2 and 2.3. As we can see, both figures have a global maximum at $\mu = .8$, which makes this the MLE. Of course, this is also the value that we obtain by computing $\bar{y}$. Hence, the figures illustrate that $\bar{y}$ is the MLE.

Example 2.5. For another example, consider the exponential distribution

$$f(y|\beta) = \frac{1}{\beta}\exp\left(-\frac{y}{\beta}\right)$$

for $y \geq 0$ and $\beta > 0$. Given a sample of n independent observations, the likelihood function is given by

$$\begin{aligned}\mathcal{L} &= \frac{1}{\beta}\exp\left(-\frac{y_1}{\beta}\right) \times \frac{1}{\beta}\exp\left(-\frac{y_2}{\beta}\right) \times \cdots \times \frac{1}{\beta}\exp\left(-\frac{y_n}{\beta}\right) \\ &= \left(\frac{1}{\beta}\right)^n \exp\left(-\frac{\sum_i y_i}{\beta}\right)\end{aligned}$$

The log-likelihood function is

$$\ell = -n\ln\beta - \frac{\sum_i y_i}{\beta}$$

We optimize ℓ by first taking the first derivative with respect to β:

$$\frac{dl}{d\beta} = \frac{\sum_i y_i}{\beta^2} - \frac{n}{\beta}$$

Setting this derivative to zero yields:

$$\begin{aligned}
\frac{\sum_i y_i}{\beta^2} - \frac{n}{\beta} = 0 &\Leftrightarrow \\
\frac{1}{\beta}\left(\frac{\sum_i y_i}{\beta} - n\right) = 0 &\Leftrightarrow \\
\frac{\sum_i y_i}{\beta} = n &\Leftrightarrow \\
\hat{\beta} = \frac{\sum_i y_i}{n} &\Leftrightarrow \\
\hat{\beta} = \bar{y}
\end{aligned}$$

Hence, it follows that the sample mean is the ML estimator of β. To verify that $\bar{y}$ produces a maximum, we evaluate the second order condition:

$$\frac{d^2 l}{d\beta d\beta} = -\frac{2\sum_i y_i}{\beta^3} + \frac{n}{\beta^2}$$

Substituting $\bar{y}$ for β and solving yields

$$-\frac{n}{\bar{y}^2} < 0$$

This means that $\bar{y}$ is indeed the ML estimator.

2.3 More about Log-Likelihood Functions

Let us consider again the Poisson log-likelihood function that was derived in Example 2.4:

$$\ell = \sum_i y_i \ln(\mu) - n\mu - \sum_i \ln(y_i!)$$

You will notice that only the first two terms involve the parameter of interest. The third term is invariant or constant with respect to μ and does not yield information about this parameter. From an estimation perspective, then, we

could just as easily formulate the log-likelihood function as $\sum_i y_i \ln(\mu) - n\mu$. This is called the **kernel of the log-likelihood function**. It is that part of the log-likelihood function that contains the parameter(s) of interest or, conversely, that part of the log-likelihood function that omits constant terms. Log-likelihood functions can often be simplified by stating only their kernel, so it is useful to consider what part of the log-likelihood function provides information about a parameter and what part does not. In the remainder of this report, we shall frequently consider only the kernel of the log-likelihood.

Example 2.6. In Example 2.3, we described the binomial likelihood function:

$$\mathcal{L} = \binom{n}{y}\pi^y(1-\pi)^{n-y}$$

The corresponding log-likelihood function is

$$\ell = y\ln\pi + (n-y)\ln(1-\pi) + \ln\binom{n}{y}$$

The last term of ℓ is not a function of the parameters and may thus be omitted for optimization purposes. This means that the kernel of the log-likelihood function is given by

$$\ell = y\ln\pi + (n-y)\ln(1-\pi)$$

Optimization of this kernel will yield $\hat{\pi} = y/n$. (You should verify this result.)

The ideal log-likelihood function resembles a quadratic function. There are two important reasons for this. First, quadratic functions have a unique maximum that can be retrieved relatively easily. Second, for quadratic functions, it is guaranteed that the resulting estimators are asymptotically normally distributed (LeCam 1970; see also Chapter 4). Where log-likelihood functions depart from quadratic functions, estimation tends to become more complex and issues of local maxima can arise (see Chapter 5).

2.4 Standard Errors

For purposes of statistical inference, including the computation of confidence intervals (see Chapter 7), it is necessary to compute the standard error of

the ML estimator of a parameter. This task requires that we compute the expected Fisher information, invert it, and then take the square root. The result of these operations is a formula for the standard error of the estimator.

In Chapter 2.2, we saw that the observed Fisher information is simply the negative of the second derivative of the log-likelihood function, i.e. $-\ell''$ (see Definition 2.2). The expected Fisher information is defined as follows:

> ***Definition 2.6.*** The **expected Fisher information** is equal to negative one times the expected value of the second derivative of the log-likelihood function, i.e. $-\mathcal{E}[\ell'']$.

The following examples illustrate the derivation of the expected Fisher information.

> **Example 2.7.** For the Poisson log-likelihood, Example 2.4 demonstrated that $\ell" = -(1/\mu^2)\sum_i y_i$. To obtain the expected Fisher information we begin by taking the expectation of ℓ'':
>
> $$\begin{aligned}
> \mathcal{E}[\ell''] &= \mathcal{E}\left[-\frac{\sum_i y_i}{\mu^2}\right] \\
> &= -\frac{1}{\mu^2}\,\mathcal{E}\left[\sum_i y_i\right] \\
> &= -\frac{1}{\mu^2}\sum_i \mathcal{E}[y_i] \\
> &= -\frac{1}{\mu^2}\sum_i \mu \\
> &= -\frac{n\mu}{\mu^2} \\
> &= -\frac{n}{\mu}
> \end{aligned}$$
>
> The second line reflects the property that the expectation of a constant times a variable is equal to the constant times the expectation of the variable. In this case, the constant is $-(1/\mu^2)$ and the variable is y_i. The third line reflects the property that the expectation of a sum is equal to the sum of the expectations. The fourth line shows that the expected value of a Poisson-distributed variable is μ. The fifth line carries out the summation over this expected value, and the last line simplifies the result. This result is not yet the expected Fisher

information. To obtain this we need to take negative one times the expected value, i.e.

$$-\mathcal{E}[\ell''] = \frac{n}{\mu}$$

Example 2.8. For the exponential log-likelihood, Example 2.5 demonstrated that $\ell'' = -(2/\beta^3)\sum_i y_i + (n/\beta^2)$. Taking the expected value (and applying familiar rules of the calculus of expectations) we get:

$$\begin{aligned}\mathcal{E}[\ell''] &= -\frac{2}{\beta^3}\sum_i \mathcal{E}[y_i] + \frac{n}{\beta^2} \\ &= -\frac{2n\beta}{\beta^3} + \frac{n}{\beta^2} \\ &= -\frac{n}{\beta^2}\end{aligned}$$

Here we have taken advantage of the result that the expectation of an exponentially-distributed variable is equal to β. Taking negative one times the expectation yields n/β^2 as the expected Fisher information.

The **variance** of a ML estimator is obtained by inverting the expected Fisher information:

Definition 2.7. The variance of a ML estimator is equal to the reciprocal of the expected Fisher information for that estimator, i.e.

$$V[\hat{\theta}] = \frac{1}{-\mathcal{E}[\ell'']} \tag{2.4}$$

The **standard error** is equal to the square root of (2.4). Thus, once the expected Fisher information has been obtained, computing standard errors for ML estimators is quite easy.

Example 2.9. In example 2.7, we obtained the expected Fisher information for the Poisson distribution. Using this result we obtain

$$\begin{aligned}V[\hat{\mu}] &= \left(\frac{n}{\mu}\right)^{-1} \\ &= \frac{\mu}{n}\end{aligned}$$

The standard error is then $\sqrt{\mu}/\sqrt{n}$.

Example 2.10. In example 2.8, we showed that the expected Fisher information for the exponential distribution is n/β^2. Inverting this result yields:

$$V[\hat{\beta}] = \frac{\beta^2}{n}$$

This means that the standard error of $\hat{\beta}$ is equal to $\beta/\sqrt{n}$.

If one takes a close look at the formulas derived in examples 2.9 and 2.10, one observes that the right-hand side is a function of the very parameters that we seek to estimate. We can overcome this problem by substituting the ML estimators of the parameters. Doing this, there are no more unknown quantities on the right-hand side of the variance formulas. The resulting variances are so-called **estimated variances** and the corresponding standard errors are **estimated standard errors**. For instance, the estimated variance for the Poisson ML estimator is $\hat{V}[\hat{\mu}] = \hat{\mu}/n$; the estimated variance for the exponential ML estimator is $\hat{V}[\hat{\beta}] = \hat{\beta}^2/n$. The "hat" over the V-symbol highlights that we are dealing with an estimated quantity. In practice, almost all of the standard errors that we report will be *estimated* standard errors since it is rare to find that the right-hand side of a variance formula contains known quantities only.

2.5 The Role of Distributional Assumptions

Throughout the discussion so far we have made distributional assumptions to derive ML estimators. While nonparametric and semi-parametric MLE methods exist (e.g. Gallant and Nychka 1987), they are beyond the scope of this report. Some people view the need to make distributional assumptions a real weakness of (parametric) MLE because such assumptions might be arbitrary and, if violated, could undermine the statistical properties of the estimators.

It is possible to formulate several responses to this criticism. First, it is useful to remember Clyde Coombs' (1976) adage that "all knowledge is the result of theory—we buy information with assumptions" (p. 5). The versatility and desirable statistical properties that MLE offers are possible in large part because we make the kinds of distributional assumptions that OLS and GLS, for instance, avoid. In this sense, distributional assumptions

are the price that one pays for the ability to estimate complex models and have some degree of confidence in the resulting estimates.

Second, the notion that distributional assumptions are of necessity arbitrary is exaggerated. In many instances, political scientists know a great deal about the phenomena they study, including knowledge of (plausible) distributions of those phenomena. This allows for the specification of a distributional assumption that (at least approximately) captures the real distribution and/or that seems reasonable in terms of the theoretical properties of the phenomenon of interest.

King (1989: 45-48) provides a great example of a theoretically derived distributional assumption that is worthwhile repeating. Imagine that our goal is to model the number of U.S. Senators who vote for a particular bill. There are $n = 100$ Senators and the number of "yeah" votes, Y, can range between 0 and 100. A reasonable starting point is to treat this as a binomial problem. Each Senator can be characterized through a Bernoulli variable, X, which takes on the value of 1 if the Senator voted for the bill and 0 if he/she did not. The probability of a positive vote is π. Treating this probability as constant and assuming that the Senators act independently, then the probability distribution over $y = \sum_i x_i$ (were i denotes a particular Senator) is

$$f(y|\pi) = \binom{n}{y}\pi^y(1-\pi)^{n-y}$$

The problem with this distributional assumption is that it treats π as fixed across Senators. In effect, we assume that every Senator is equally likely to vote for the bill. This is unreasonable, given that bills are often ideological and will thus be more attractive to one side of the political aisle than the other.[2] We thus need to amend the binomial distribution by treating π as variable. This means that we have to make a secondary distributional assumption concerning the probability of a "yeah" vote. While one can think of a number of distributions that could do the job, in practice statisticians usually opt for the **beta distribution** since it is both flexible and tractable. This distribution has two parameters and is given by:

$$f(\pi|\rho,\gamma) = \frac{\Gamma\left(\rho\gamma^{-1}+[1-\rho]\gamma^{-1}\right)}{\Gamma\left(\rho\gamma^{-1}\right)\Gamma\left([1-\rho]\gamma^{-1}\right)}\pi^{\rho\gamma^{-1}-1}(1-\pi)^{(1-\rho)\gamma^{-1}-1}$$

[2]In addition, we assume that the Senators operate independently. This assumption, too, is unreasonable due to partisan divisions in the Senate. It turns out that the solution for dealing with the problem of a fixed π also resolves the faulty independence assumption.

Here $\Gamma(z)$ is the gamma function, $0 < \pi < 1$, and $0 < \rho < 1$. Statistically, $\mathcal{E}[\rho] = \pi$ and γ captures variation in π across the binary outcomes (i.e. Senators' vote choices). The beta distribution is flexible in that it includes both uni-modal and bi-modal cases, as well as symmetrical and skewed distributions. As such, it provides a useful means of specifying heterogeneity in π.

We now need to bring the beta and binomial distributions together. That is, we need to modify the binomial distribution in such a manner that π varies according to a beta distribution. The process of joining these distributions is more generally known as **compounding** and has two steps. First, we characterize the joint distribution of y and π. Next, we collapse over π so that we are left with a distribution for y only. Characterization of the joint distribution depends on the multiplicative theorem of probability theory. For two events, A and B, the multiplicative theorem states that $\Pr(AB) = \Pr(A|B)\Pr(B)$. Accordingly,

$$f(y, \pi|\rho, \gamma) \quad = \quad f_b(y|\pi) f_\beta(\pi|\rho, \gamma)$$

where f_b denotes the binomial distribution and f_β denotes the beta distribution. We then collapse over π by taking

$$\begin{aligned} f(y|\rho, \gamma) &= \int_0^1 f(y, \pi|\rho, \gamma) d\pi \\ &= \frac{n!\Gamma\left(\rho\gamma^{-1} + (1-\rho)\gamma^{-1}\right)\Gamma\left(\rho\gamma^{-1} + y\right)\Gamma\left((1-\rho)\gamma^{-1} + n - y\right)}{y!(n-y)!\Gamma\left(\rho\gamma^{-1}\right)\Gamma\left((1-\rho)\gamma^{-1}\right)\left(\rho\gamma^{-1}(1-\rho)\gamma^{-1} + n\right)} \\ &= \frac{n!\Gamma(\alpha+\beta)\Gamma(\alpha+y)\Gamma(\beta+n-y)}{y!(n-y)!\Gamma(\alpha)\Gamma(\beta)\Gamma(\alpha+\beta+n)} \end{aligned}$$

with $\alpha = \rho\gamma^{-1}$ and $\beta = (1-\rho)\gamma^{-1}$. This distribution is known as the **extended beta binomial distribution**.[3]

[3]King (1989) presents a slightly different equation, which is patterned after Prentice (1986). That equation removes the gamma function:

$$f(y|pi, \gamma) \quad = \quad \frac{n!}{y!(n-y)!} \frac{\prod_{j=0}^{y-1}(\pi + \gamma j) \prod_{j=0}^{n-y-1}(1 - \pi + \gamma j)}{\prod_{j=0}^{n-1}(1 + \gamma j)}$$

(Here ρ was replaced by π and advantage was taken of several properties of the gamma function.) This form is advantageous because it is easy to demonstrate that it collapses to the binomial distribution as $\gamma \to 0$. Positive values of γ imply positive correlation between units, while negative values imply negative correlation.

This example demonstrates that a distributional assumption for a political phenomenon can be derived from theoretical notions about how the phenomenon behaves. As such, the assumption is far from arbitrary, although some subjective choices entered the derivation (e.g. the choice of the beta distribution for π). Indeed, one could argue that the need to think through distributional implications makes MLE more theoretically rich than, for example, OLS and GLS, which are ultimately methods for data fitting.

The example also illustrates another point. Applied researchers often rely on "canned" MLE routines in statistical packages. This is fine as long as one realizes that these routines have embedded distributional assumptions and as long as one is comfortable with those assumptions. If a routine relies on a distributional assumption that seems empirically or theoretically implausible, then there is little point in using an ML estimator based on that assumption for, in this case, one truly runs the risk of drawing invalid inferences. However, this is not an indictment against MLE persé, just against the particular application. The researcher should have picked a different distribution and worked out the estimator rather than relying on the "canned routine." As a general rule, one should put considerable thought into the distributional assumptions that one makes and derive ML estimators according to the distribution that makes the most empirical and/or theoretical sense.

Chapter 3

Estimation of Multiple Parameters

In many cases, the distributions in which we are interested contain multiple parameters. This is true, for example, of the normal distribution, which is characterized by both a mean and a variance. Even if the distribution contains only one parameter, we may wish to model it as a function of covariates, which effectively means that we will have to consider a set of parameters. We shall encounter an example of this in Chapter 13 when we discuss the Poisson regression model.

The theory of MLE extends quite easily to the multi-parameter case. Indeed, the logic of estimating multiple parameters is similar to the single-parameter case. However, the mechanics of MLE become a bit more complicated in that we will have to rely on multivariate calculus. It will also prove useful to redefine a number of constructs in terms of vectors and matrices. As in Chapter 2, we shall consider only instances where ML estimators can be found via algebraic means. In Chapter 5, we shall consider more complex (and realistic) cases where MLE requires numerical optimization.

3.1 Likelihoods, Gradients, and Hessians

Consider a probability density or mass function $f(y|\boldsymbol{\theta})$, where $\boldsymbol{\theta}$ is a vector of p parameters. Drawing a sample from this distribution we can define the likelihood function in the following way:

Definition 3.1. For a sample of n (conditionally) independent

observations, the likelihood function is given by

$$\mathcal{L} = \prod_{i=1}^{n} f(y_i|\boldsymbol{\theta}) \tag{3.1}$$

Further,

Definition 3.2. The log-likelihood function is given by

$$\ell = \sum_{i=1}^{n} \ln f(y_i|\boldsymbol{\theta}) \tag{3.2}$$

The maximum likelihood estimator can be defined in the following manner.

Definition 3.3. Let $\hat{\boldsymbol{\theta}}$ be the maximum likelihood estimator then

$$\ell(\hat{\boldsymbol{\theta}}) = \sup_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} \ell(\boldsymbol{\theta}) \tag{3.3}$$

where $\boldsymbol{\Theta}$ is the p-dimensional parameter space.

The ML estimator can be obtained by taking the first partial derivatives of ℓ, setting them to zero, and solving for the elements of $\boldsymbol{\theta}$. An example can illustrate how this process works.

Example 3.1. Consider the normal PDF:

$$f(y|\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{1}{2\sigma^2}(y-\mu)^2\right\}$$

where μ and σ^2 are the parameters of interest, corresponding to the population mean and variance, respectively. Thus, $\boldsymbol{\theta'} = (\mu\ \sigma^2)$. Assume that a sample of n independent observations has been drawn

from this distribution, then the likelihood function is given by

$$\begin{aligned}\mathcal{L} &= \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left\{-\frac{1}{2\sigma^2}(y_1-\mu)^2\right\}\times \\ &\quad \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left\{-\frac{1}{2\sigma^2}(y_2-\mu)^2\right\}\times\cdots\times \\ &\quad \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left\{-\frac{1}{2\sigma^2}(y_n-\mu)^2\right\} \\ &= (2\pi\sigma^2)^{-.5n}\exp\left\{-\frac{1}{2\sigma^2}\sum_i (y_i-\mu)^2\right\}\end{aligned}$$

The log-likelihood function is

$$\ell = -.5n\ln(2\pi) - .5n\ln(\sigma^2) - \frac{1}{2\sigma^2}\sum_i (y_i-\mu)^2$$

To find the ML estimator of μ we take the first partial derivative of ℓ with respect to this parameter:

$$\frac{\partial\ell}{\partial\mu} = \frac{1}{\sigma^2}\sum_i (y_i-\mu)$$

By setting this derivative to 0 we can derive $\hat{\mu}$:

$$\begin{aligned}\frac{1}{\sigma^2}\sum_i (y_i-\mu) = 0 &\Leftrightarrow \\ \sum_i (y_i-\mu) = 0 &\Leftrightarrow \\ \sum_i y_i - n\mu = 0 &\Leftrightarrow \\ n\mu = \sum_i y_i &\Leftrightarrow \\ \hat{\mu} = \frac{\sum_i y_i}{n} &\Leftrightarrow \\ \hat{\mu} = \bar{y}&\end{aligned}$$

Thus the sample mean is the ML estimator of the mean of a normal distribution.

To find the MLE of σ^2 we take the first partial derivative of ℓ with respect to this parameter:

$$\frac{\partial\ell}{\partial\sigma^2} = -\frac{n}{2\sigma^2} + \frac{1}{2\sigma^4}\sum_i (y_i-\mu)^2$$

To obtain $\hat{\sigma}^2$ we set this derivative to 0 and solve:

$$\begin{aligned} -\frac{n}{2\sigma^2} + \frac{1}{2\sigma^4}\sum_i (y_i - \mu)^2 = 0 \quad &\Leftrightarrow \\ \frac{1}{2\sigma^2}\left\{-n + \frac{1}{\sigma^2}\sum_i (y_i - \mu)^2\right\} = 0 \quad &\Leftrightarrow \end{aligned} \tag{3.4}$$

$$\begin{aligned} -n + \frac{1}{\sigma^2}\sum_i (y_i - \mu)^2 = 0 \quad &\Leftrightarrow \\ \frac{1}{\sigma^2}\sum_i (y_i - \mu)^2 = n \quad &\Leftrightarrow \\ \hat{\sigma}^2 = \frac{\sum_i (y_i - \hat{\mu})^2}{n} \end{aligned}$$

Here we have substituted $\hat{\mu}$ for μ in order to ensure that the right-hand side consists of known quantities only. Notice that the ML estimator of σ^2 is not the same as the sample variance, s^2, which divides by $n-1$ instead of n.[1]

An example can illustrate these results. Imagine that we have drawn a sample consisting of the following 3 independent observations: 0, 3, and 6. In this case, $\hat{\mu} = 3$, $\hat{\sigma}^2 = 6$, and $\hat{\sigma} = 2.45$. It should be the case that the log-likelihood function reaches its apex for these values of the parameters. The likelihood function in this case is

$$\mathcal{L} \;=\; (2\pi\sigma^2)^{-1.5} \exp\left\{-\frac{1}{2\sigma^2}\left(3\mu^2 - 18\mu + 45\right)\right\}$$

We can plot $\mathcal{L}$ for different values of μ and σ, as is done in the surface plot in Figure 3.1 and the contour plot in Figure 3.2.[2]. The maximum in these figures is at $\mu = 3$ and $\sigma = 2.45$, illustrating that these are the ML estimates.

If you paid close attention to Example 3.1, you would have noticed that one important step was omitted, namely the derivation of the second partial derivative of ℓ. This step is actually a bit complicated in a multi-parameter

[1]The ML estimator of σ^2 is biased in small samples, although not asymptotically. The bias can be corrected by dividing by $n-1$ instead of n.

[2]In these figures $\mathcal{L}$ has been re-scaled so that its apex is 1. We call such a likelihood the **normalized likelihood**.

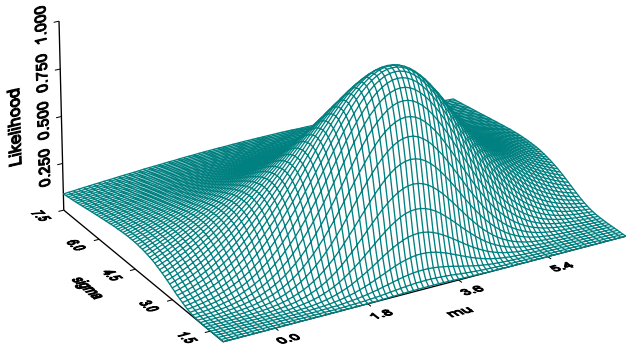

Figure 3.1: Normal Likelihood Function

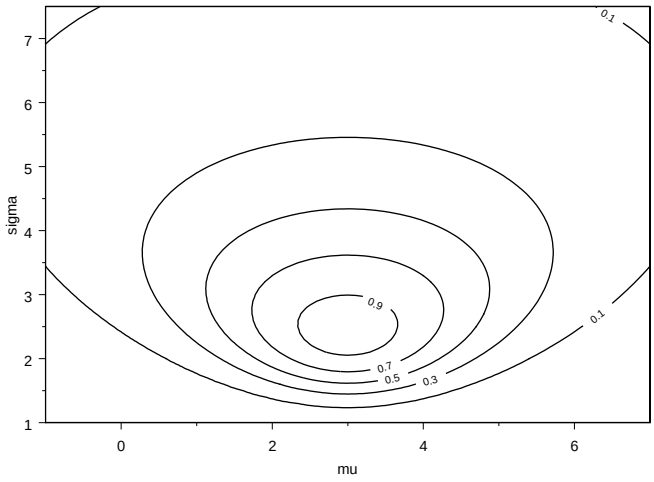

Figure 3.2: Normal Likelihood Contour

estimation problem because we can define multiple second partial derivatives. Specifically, in Example 3.1 there are four second partial derivatives:

$$\begin{aligned}
\frac{\partial^2 \ell}{\partial \mu \partial \mu} &= -\frac{n}{\sigma^2} \\
\frac{\partial^2 \ell}{\partial \mu \partial \sigma^2} &= -\frac{\sum_i (y_i - \mu)}{\sigma^4} \\
\frac{\partial^2 \ell}{\partial \sigma^2 \partial \mu} &= -\frac{\sum_i (y_i - \mu)}{\sigma^4} \\
\frac{\partial^2 \ell}{\partial \sigma^2 \partial \sigma^2} &= \frac{n}{2\sigma^4} - \frac{\sum_i (y_i - \mu)^2}{\sigma^6}
\end{aligned}$$

The first equation means that ℓ was first differentiated with respect to μ; the resulting first partial derivative was then again differentiated with respect to μ. The second equation means that ℓ was first differentiated with respect to μ; the resulting first partial derivative was then differentiated with respect to σ^2. The third equation means that ℓ was first differentiated with respect to σ^2; the resulting first partial derivative was then differentiated with respect to μ. Finally, the fourth equation means that ℓ was first differentiated with respect to σ^2; the resulting first partial derivative was then again differentiated with respect to σ^2.

It is conventional to group these second partial derivatives in a matrix, called the Hessian and denoted by the symbol $\boldsymbol{\mathcal{H}}$ (or $\mathbf{H}$). For the normal PDF, the Hessian is:

$$\begin{aligned}
\boldsymbol{\mathcal{H}} &= \begin{bmatrix} \frac{\partial^2 \ell}{\partial \mu \partial \mu} & \frac{\partial^2 \ell}{\partial \mu \partial \sigma^2} \\ \frac{\partial^2 \ell}{\partial \sigma^2 \partial \mu} & \frac{\partial^2 \ell}{\partial \sigma^2 \partial \sigma^2} \end{bmatrix} \\
&= \begin{bmatrix} -\frac{n}{\sigma^2} & -\frac{\sum_i (y_i - \mu)}{\sigma^4} \\ -\frac{\sum_i (y_i - \mu)}{\sigma^4} & \frac{n}{2\sigma^4} - \frac{\sum_i (y_i - \mu)^2}{\sigma^6} \end{bmatrix}
\end{aligned}$$

In general, the Hessian is defined in the following manner.

> ***Definition 3.4.*** The **Hessian** is the $p \times p$ matrix of second partial derivatives of the log-likelihood function with respect to

the parameters:

$$\begin{aligned}\boldsymbol{\mathcal{H}} &= \frac{\partial^2 \ell}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}'} \\ &= \begin{bmatrix} \frac{\partial^2 \ell}{\partial \theta_1^2} & \frac{\partial^2 \ell}{\partial \theta_1 \partial \theta_2} & \cdots & \frac{\partial^2 \ell}{\partial \theta_1 \partial \theta_p} \\ \frac{\partial^2 \ell}{\partial \theta_2 \partial \theta_1} & \frac{\partial^2 \ell}{\partial \theta_2^2} & \cdots & \frac{\partial^2 \ell}{\partial \theta_2 \partial \theta_p} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 \ell}{\partial \theta_p \partial \theta_1} & \frac{\partial^2 \ell}{\partial \theta_p \partial \theta_2} & \cdots & \frac{\partial^2 \ell}{\partial \theta_p^2} \end{bmatrix}\end{aligned} \tag{3.5}$$

This may also be written as

$$\begin{aligned}\boldsymbol{\mathcal{H}} &= \sum_{i=1}^{n} \frac{\partial^2 \ell_i}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}'} \\ &= \sum_{i=1}^{n} \boldsymbol{\mathcal{H}}_i\end{aligned} \tag{3.6}$$

where $\boldsymbol{\mathcal{H}}_i$ is the Hessian matrix evaluated for a particular observation in the sample.

Just as the second partial derivatives of the log-likelihood function can be assembled into a matrix, the first partial derivatives can be assembled into a vector, which is known as the gradient and denoted by $\boldsymbol{\nabla}$ (or sometimes $\mathbf{g}$). For the normal PDF of example 3.1, the gradient is given by:

$$\begin{aligned}\boldsymbol{\nabla} &= \begin{bmatrix} \frac{\partial \ell}{\partial \mu} \\ \frac{\partial \ell}{\partial \sigma^2} \end{bmatrix} \\ &= \begin{bmatrix} \frac{1}{\sigma^2} \sum_i (y_i - \mu) \\ -\frac{n}{2\sigma^2} + \frac{1}{2\sigma^4} \sum_i (y_i - \mu)^2 \end{bmatrix}\end{aligned}$$

In general, the gradient is defined in the following manner.

Definition 3.5. The **gradient** is the p-vector of first derivatives of the log-likelihood function with respect to the parameters:

$$\begin{aligned}\boldsymbol{\nabla} &= \frac{\partial \ell}{\partial \boldsymbol{\theta}} \\ &= \begin{bmatrix} \frac{\partial \ell}{\partial \theta_1} \\ \frac{\partial \ell}{\partial \theta_2} \\ \vdots \\ \frac{\partial \ell}{\partial \theta_p} \end{bmatrix}\end{aligned} \tag{3.7}$$

This may also be written as

$$\begin{aligned} \boldsymbol{\nabla} &= \sum_{i=1}^{n} \frac{\partial \ell_i}{\partial \boldsymbol{\theta}} \\ &= \sum_{i=1}^{n} \boldsymbol{\nabla}_i \end{aligned} \tag{3.8}$$

Here, $\boldsymbol{\nabla}_i$ is known as the **score**, which is simply a sample unit's values on the first derivative of the log-likelihood function. The quantity $\sum_i \boldsymbol{\nabla}_i$ is known as the **score function** (or score vector), which is just another name for the gradient.

With the definitions of gradient and Hessian in place, it is now possible to define the first and second order conditions for ML estimators.

Definition 3.6. The **first-order condition** for finding an ML estimator is

$$\boldsymbol{\nabla} = \mathbf{0} \tag{3.9}$$

This is the **likelihood equation**.

Definition 3.7. The **second-order condition** for finding an ML estimator is that $\boldsymbol{\mathcal{H}}$ is negative definite.[3]

The combination of the first and second-order conditions allows one to identify ML estimators.

Example 3.2. For the data and estimates presented in Example 3.1, the Hessian is

$$\boldsymbol{\mathcal{H}} = \begin{bmatrix} -.500 & .000 \\ .000 & -.042 \end{bmatrix}$$

It can be easily ascertained that this matrix has only negative eigenvalues (namely -.5 and -.042), which means that $\boldsymbol{\mathcal{H}}$ is negative definite

[3] A matrix is negative definite if all of its eigenvalues are negative. Alternatively, $\boldsymbol{\mathcal{H}}$ is negative definite if $\mathbf{x}'\boldsymbol{\mathcal{H}}\mathbf{x} < 0$, for any conformable column vector $\mathbf{x}$.

and that the estimates reported earlier are ML estimates.

More generally, it is easily demonstrated that the Hessian for the normal log-likelihood is negative definite. Substituting the estimators into the formula derived earlier, we have

$$\boldsymbol{\mathcal{H}} = \begin{bmatrix} -\frac{n}{\hat{\sigma}^2} & -\frac{\sum_i (y_i - \hat{\mu})}{\hat{\sigma}^4} \\ -\frac{\sum_i (y_i - \hat{\mu})}{\hat{\sigma}^4} & \frac{n}{2\hat{\sigma}^4} - \frac{\sum_i (y_i - \hat{\mu})^2}{\hat{\sigma}^6} \end{bmatrix}$$

In this expression, $\sum_i (y_i - \hat{\mu}) = 0$, so that the diagonal elements are 0.[4] We also know that $\sum_i (y_i - \hat{\mu})^2 = n\hat{\sigma}^2$. Hence,

$$\begin{aligned} \frac{n}{2\hat{\sigma}^4} - \frac{\sum_i (y_i - \hat{\mu})^2}{\hat{\sigma}^6} &= \frac{n}{2\hat{\sigma}^4} - \frac{n\hat{\sigma}^2}{\hat{\sigma}^6} \\ &= \frac{n}{2\hat{\sigma}^4} - \frac{n}{\hat{\sigma}^4} \\ &= -\frac{n}{2\hat{\sigma}^4} \end{aligned}$$

The Hessian may thus be written as

$$\boldsymbol{\mathcal{H}} = \begin{bmatrix} -\frac{n}{\hat{\sigma}^2} & 0 \\ 0 & -\frac{n}{2\hat{\sigma}^4} \end{bmatrix}$$

For a matrix where the diagonal elements are zero, the eigenvalues are simply the diagonal values. Both of the diagonal elements in the Hessian are negative, which means that all of the eigenvalues are negative and that $\boldsymbol{\mathcal{H}}$ is negative definite.

As in the univariate case, curvature of the (log-) likelihood function is given by $-\boldsymbol{\mathcal{H}}$. This is known as the observed Fisher information (matrix).

Definition 3.8. The **observed Fisher information** matrix is given by

$$\boldsymbol{\mathcal{I}}_o = -\boldsymbol{\mathcal{H}} \tag{3.10}$$

We prefer curvature to be as large as possible in order to obtain maximum precision in our estimates.

[4] *Proof:* $\sum_i (y_i - \hat{\mu}) = \sum_i (y_i - \bar{y}) = \sum_i y_i - \sum_i \bar{y} = n\bar{y} - n\bar{y} = 0$.

3.2 Standard Errors

In the multi-parameter case, standard errors of ML estimators are calculated by inverting the expected Fisher information matrix and by taking the square root of the diagonal elements of the resulting matrix. Let us begin by defining the expected Fisher information (matrix).

Definition 3.9. The **expected Fisher information** matrix is given by negative one times the expectation of the Hessian:

$$\boldsymbol{\mathcal{I}}_e = -\mathcal{E}\left[\boldsymbol{\mathcal{H}}\right] \tag{3.11}$$

We now illustrate computation of the expected Fisher information for the normal PDF.

Example 3.3. For the normal PDF, we have already seen that

$$\boldsymbol{\mathcal{H}} = \begin{bmatrix} -\frac{n}{\sigma^2} & -\frac{\sum_i (y_i-\mu)}{\sigma^4} \\ -\frac{\sum_i (y_i-\mu)}{\sigma^4} & \frac{n}{2\sigma^4} - \frac{\sum_i (y_i-\mu)^2}{\sigma^6} \end{bmatrix}$$

To obtain the expected Fisher information matrix we start by taking the expected values of the elements in the Hessian. It is easy to demonstrate that $\mathcal{E}\left[-\frac{n}{\sigma^2}\right] = -\frac{n}{\sigma^2}$ and that $\mathcal{E}\left[\frac{-\sum_i (y_i-\mu)}{\sigma^4}\right] = 0$.[5] Obtaining the expectation of the last element of the Hessian is only slightly more complex:

$$\begin{aligned}
\mathcal{E}\left[\frac{n}{2\sigma^4} - \frac{\sum_i (y_i-\mu)^2}{\sigma^6}\right] &= \mathcal{E}\left[\frac{n}{2\sigma^4}\right] - \mathcal{E}\left[\frac{\sum_i (y_i-\mu)^2}{\sigma^6}\right] \\
&= \frac{n}{2\sigma^4} - \frac{1}{\sigma^6}\sum_i \mathcal{E}\left[(y_i-\mu)^2\right] \\
&= \frac{n}{2\sigma^4} - \frac{1}{\sigma^6}\sum_i \sigma^2 \\
&= \frac{n}{2\sigma^4} - \frac{n\sigma^2}{\sigma^6} \\
&= \frac{n}{2\sigma^4} - \frac{n}{\sigma^4} \\
&= -\frac{n}{2\sigma^4}
\end{aligned}$$

[5] The second result should be familiar. The proof centers about demonstrating that $\mathcal{E}[\sum_i (y_i - \mu)] = 0$. We start by rewriting this expression: $\mathcal{E}[\sum_i y_i] - \mathcal{E}[\sum_i \mu]$. Expansion gives $\sum_i \mathcal{E}[y_i] - n\mu$. Since $\mathcal{E}[y_i] = \mu$ this may also be written as $\sum_i \mu - n\mu$. Since $\sum_i \mu = n\mu$, it follows immediately that $\sum_i \mu - n\mu = 0$.

Here, the third line reflects the fact that $\mathcal{E}\left[(y_i - \mu)^2\right]$ is the mathematical definition of the population variance (i.e. it is the second moment about the mean, which equals σ^2). Thus, the expected Fisher information is given by

$$\begin{aligned} -\mathcal{E}\left[\boldsymbol{\mathcal{H}}\right] &= -\begin{bmatrix} -\frac{n}{\sigma^2} & 0 \\ 0 & -\frac{n}{2\sigma^4} \end{bmatrix} \\ &= \begin{bmatrix} \frac{n}{\sigma^2} & 0 \\ 0 & \frac{n}{2\sigma^4} \end{bmatrix} \end{aligned}$$

The variance-covariance matrix of the estimator (VCE) is defined as the inverse of the expected Fisher information matrix.

Definition 3.10. The **variance-covariance matrix** of $\hat{\boldsymbol{\theta}}$ is given by

$$\mathbf{V}\left[\hat{\boldsymbol{\theta}}\right] = \boldsymbol{\mathcal{I}}_e^{-1} \tag{3.12}$$

The **standard errors** are equal to the square roots of the diagonal elements of this matrix.

Example 3.4. For the normal PDF, the variance-covariance matrix is given by

$$\begin{aligned} \boldsymbol{\mathcal{I}}_e^{-1} &= \begin{bmatrix} \frac{n}{\sigma^2} & 0 \\ 0 & \frac{n}{2\sigma^4} \end{bmatrix}^{-1} \\ &= \begin{bmatrix} \frac{\sigma^2}{n} & 0 \\ 0 & \frac{2\sigma^4}{n} \end{bmatrix} \end{aligned}$$

Thus, the standard error of $\hat{\mu}$ is equal to $\sigma/\sqrt{n}$, while the standard error of $\hat{\sigma}^2$ is equal to $(\sqrt{2}\sigma^2)/\sqrt{n}$.

Notice that the formulas for the standard errors of $\hat{\mu}$ and $\hat{\sigma}^2$ contain the unknown parameter σ. In practice, then, we obtain **estimated standard errors** by substituting estimates for the parameters that appear in the formulas of the standard errors.

3.3 The Linear Regression Model

The linear regression model is one of the most important models encountered in political analysis. As such, it is worthwhile dwelling on the estimation of this model via MLE. Consider a continuous response variable y_i that is a linear function of a set of fixed (i.e. non-stochastic) covariates (including a constant), $\boldsymbol{x}_i$, and a stochastic error term, ϵ_i:

$$y_i = \boldsymbol{x}_i\boldsymbol{\beta} + \epsilon_i$$

Under the classical normal linear regression model (CNLRM), the errors are normally distributed with a mean of zero and constant variance:

$$\epsilon_i \sim N\left(0, \sigma^2\right)$$

Moreover, $\mathcal{E}[\epsilon_i, \epsilon_j] = 0$ for $i \neq j$, i.e. there is no autocorrelation. The parameter vector consists of the regression coefficients in $\boldsymbol{\beta}$, as well as σ^2: $\boldsymbol{\theta'} = (\boldsymbol{\beta}\ \sigma^2)$. Thus, $\boldsymbol{\theta'}$ consists of $p = K + 2$ parameters: K partial regression coefficients, $\beta_1 \cdots \beta_p$; a constant, β_0; and an error variance, σ^2.

Based on the model and the assumptions about the errors, the conditional distribution of the response variable is given by

$$f(y_i|\boldsymbol{x}_i, \boldsymbol{\beta}, \sigma^2) \sim N\left(\boldsymbol{x}_i\boldsymbol{\beta}, \sigma^2\right)$$

If we draw n independent observations from this distribution, then the likelihood function is given by[6]

$$\mathcal{L} = (2\pi\sigma^2)^{-.5n} \exp\left\{-\frac{1}{2\sigma^2}\sum_i (y_i - \boldsymbol{x}_i\boldsymbol{\beta})^2\right\}$$

The log-likelihood function is given by

$$\begin{aligned}
\ell &= -.5n\ln(2\pi) - .5n\ln(\sigma^2) - \frac{1}{2\sigma^2}\sum_i (y_i - \boldsymbol{x}_i\boldsymbol{\beta})^2 \\
&= -.5n\ln(2\pi) - .5n\ln(\sigma^2) - \frac{1}{2\sigma^2}\{(\boldsymbol{y} - \boldsymbol{X\beta})'(\boldsymbol{y} - \boldsymbol{X\beta})\} \\
&= -.5n\ln(2\pi) - .5n\ln(\sigma^2) - \frac{1}{2\sigma^2}\{\boldsymbol{y'y} - 2\boldsymbol{\beta'X'y} + \boldsymbol{\beta'X'X\beta}\}
\end{aligned}$$

[6]To obtain this result, simply substitute $\mathbf{x}_i\boldsymbol{\beta}$ for μ in the likelihood function derived in Example 3.1.

Here $\boldsymbol{y}$ is a $n \times 1$ vector stacking all of the observations on the response variable and $\boldsymbol{X}$ is a $n \times (K+1)$ matrix stacking all of the observations on the covariates.

Estimation of the constant and partial regression coefficients requires that we take the first partial derivative with respect to $\boldsymbol{\beta}$:[7]

$$\frac{\partial \ell}{\partial \boldsymbol{\beta}} = -\frac{1}{2\sigma^2}\left\{-2\boldsymbol{X'y} + 2\boldsymbol{X'X\beta}\right\}$$

Setting this derivative to zero yields:

$$\begin{aligned} -\frac{1}{2\sigma^2}\left\{-2\boldsymbol{X'y} + 2\boldsymbol{X'X\beta}\right\} = \mathbf{0} &\Leftrightarrow \\ -2\boldsymbol{X'y} + 2\boldsymbol{X'X\beta} = \mathbf{0} &\Leftrightarrow \\ \boldsymbol{X'X\beta} = \boldsymbol{X'y} & \end{aligned}$$

This expression is known as the *normal equations*. In the absence of perfect multicollinearity, $\boldsymbol{X'X}$ can be inverted and the ML estimator for $\boldsymbol{\beta}$ can be found by pre-multiplying both sides of the normal equations by this inverse:

$$\begin{aligned} \left(\boldsymbol{X'X}\right)^{-1}\boldsymbol{X'X\beta} = \left(\boldsymbol{X'X}\right)^{-1}\boldsymbol{X'y} &\Leftrightarrow \\ \hat{\boldsymbol{\beta}} = \left(\boldsymbol{X'X}\right)^{-1}\boldsymbol{X'y} & \end{aligned}$$

This is the same estimator that one would get using OLS.

Estimation of the error variance requires that we take the first partial derivative of the log-likelihood function with respect to σ^2:

$$\frac{\partial \ell}{\partial \sigma^2} = -\frac{n}{2\sigma^2} + \frac{1}{2\sigma^4}(\boldsymbol{y} - \boldsymbol{X\beta})'(\boldsymbol{y} - \boldsymbol{X\beta})$$

Setting this derivative to zero yields:

$$\begin{aligned} -\frac{n}{2\sigma^2} + \frac{1}{2\sigma^4}(\boldsymbol{y} - \boldsymbol{X\beta})'(\boldsymbol{y} - \boldsymbol{X\beta}) = \mathbf{0} &\Leftrightarrow \\ \frac{1}{2\sigma^2}\left\{-n + \frac{1}{\sigma^2}(\boldsymbol{y} - \boldsymbol{X\beta})'(\boldsymbol{y} - \boldsymbol{X\beta})\right\} = 0 &\Leftrightarrow \\ \frac{1}{\sigma^2}(\boldsymbol{y} - \boldsymbol{X\beta})'(\boldsymbol{y} - \boldsymbol{X\beta}) = n &\Leftrightarrow \\ \hat{\sigma}^2 = \frac{(\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}})'(\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}})}{n} & \end{aligned}$$

[7] For the derivation of this derivative, please see my lecture notes on linear algebra, Chapter 5.4.

Recognizing that $\boldsymbol{e} = \boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}}$ is the residual, this estimator can be written more compactly as

$$\hat{\sigma}^2 = \frac{\boldsymbol{e}'\boldsymbol{e}}{n}$$

This is *not* the conventional estimator of the error variance, which divides $\boldsymbol{e}'\boldsymbol{e}$ by $n - K - 1$ instead of n.[8]

Based on the computations above, the gradient for the classical linear regression model is given by

$$\boldsymbol{\nabla} = \begin{bmatrix} -\frac{1}{2\sigma^2}\left\{-2\boldsymbol{X}'\boldsymbol{y} + 2\boldsymbol{X}'\boldsymbol{X}\boldsymbol{\beta}\right\} \\ -\frac{1}{2\sigma^2}\left\{n - \frac{1}{\sigma^2}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})'(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})\right\} \end{bmatrix}$$

The Hessian requires that we compute the second partial derivatives of the log-likelihood function. Computing these produces

$$\begin{aligned} \boldsymbol{\mathcal{H}} &= \begin{bmatrix} \frac{\partial^2 \ell}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}'} & \frac{\partial^2 \ell}{\partial \boldsymbol{\beta} \partial \sigma^2} \\ \frac{\partial^2 \ell}{\partial \sigma^2 \partial \boldsymbol{\beta}'} & \frac{\partial^2 \ell}{\partial \sigma^2 \partial \sigma^2} \end{bmatrix} \\ &= \begin{bmatrix} -\frac{1}{\sigma^2}\boldsymbol{X}'\boldsymbol{X} & -\frac{1}{\sigma^4}\left\{\boldsymbol{X}'\boldsymbol{y} - \boldsymbol{X}'\boldsymbol{X}\boldsymbol{\beta}\right\} \\ -\frac{1}{\sigma^4}\left\{\boldsymbol{y}'\boldsymbol{X} - \boldsymbol{\beta}'\boldsymbol{X}'\boldsymbol{X}\right\} & -\frac{1}{\sigma^4}\left\{-.5n + \frac{1}{\sigma^2}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})'(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})\right\} \end{bmatrix} \end{aligned}$$

This expression can be simplified if we recognize that $\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$ or, equivalently, $\boldsymbol{\epsilon} = \boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}$. Using this information, $\boldsymbol{X}'\boldsymbol{y} - \boldsymbol{X}'\boldsymbol{X}\boldsymbol{\beta} = \boldsymbol{X}'\boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{X}'\boldsymbol{\epsilon} - \boldsymbol{X}'\boldsymbol{X}\boldsymbol{\beta} = \boldsymbol{X}'\boldsymbol{\epsilon}$ and $(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})'(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}) = \boldsymbol{\epsilon}'\boldsymbol{\epsilon}$. Thus, the Hessian may also be written as:

$$\boldsymbol{\mathcal{H}} = \begin{bmatrix} -\frac{1}{\sigma^2}\boldsymbol{X}'\boldsymbol{X} & -\frac{1}{\sigma^4}\boldsymbol{X}'\boldsymbol{\epsilon} \\ -\frac{1}{\sigma^4}\boldsymbol{\epsilon}'\boldsymbol{X} & -\frac{1}{\sigma^4}\left\{-.5n + \frac{1}{\sigma^2}\boldsymbol{\epsilon}'\boldsymbol{\epsilon}\right\} \end{bmatrix}$$

This means that the observed Fisher information is

$$\boldsymbol{\mathcal{I}}_o = \begin{bmatrix} \frac{1}{\sigma^2}\boldsymbol{X}'\boldsymbol{X} & \frac{1}{\sigma^4}\boldsymbol{X}'\boldsymbol{\epsilon} \\ \frac{1}{\sigma^4}\boldsymbol{\epsilon}'\boldsymbol{X} & \frac{1}{\sigma^4}\left\{-.5n + \frac{1}{\sigma^2}\boldsymbol{\epsilon}'\boldsymbol{\epsilon}\right\} \end{bmatrix}$$

We obtain the variance-covariance matrix of the estimators by inverting the expected Fisher information. Under standard regression assumptions,

[8] The ML estimator of σ^2 is biased in small samples, although the bias disappears when K is finite and n goes to infinity. The division by $n - K - 1$ eliminates the small-sample bias of the ML estimator.

$\mathcal{E}[\boldsymbol{X'\epsilon}] = \mathbf{0}$, $\mathcal{E}[\boldsymbol{\epsilon' X}] = \mathbf{0'}$, and $\mathcal{E}[\boldsymbol{\epsilon'\epsilon}] = n\sigma^2$.[9] Thus, the expected Fisher information is

$$\begin{aligned}\boldsymbol{\mathcal{I}}_e &= \begin{bmatrix} \frac{1}{\sigma^2}\boldsymbol{X'X} & \mathbf{0} \\ \mathbf{0'} & \frac{1}{\sigma^4}\left\{-.5n + \frac{1}{\sigma^2}n\sigma^2\right\} \end{bmatrix} \\ &= \begin{bmatrix} \frac{1}{\sigma^2}\boldsymbol{X'X} & \mathbf{0} \\ \mathbf{0'} & \frac{n}{2\sigma^4} \end{bmatrix}\end{aligned}$$

Inverting this matrix yields

$$\mathbf{V} = \begin{bmatrix} \sigma^2(\boldsymbol{X'X})^{-1} & \mathbf{0} \\ \mathbf{0'} & \frac{2\sigma^4}{n} \end{bmatrix}$$

Here $\sigma^2(\boldsymbol{X'X})^{-1}$ is the variance-covariance matrix of $\hat{\boldsymbol{\beta}}$ and $(2\sigma^4)/n$ is the variance of $\hat{\sigma}^2$. In practice, we compute estimated variances and covariances by substituting $\hat{\sigma}^2$ in the formula for $\mathbf{V}$.

3.4 Nuisance Parameters*

In the multi-parameter case we often find that some of the parameters are of little theoretical interest, even though they still have to be taken into consideration in the estimation. We call such parameters nuisance parameters.

> ***Definition 3.11.*** A **nuisance parameter** is any parameter that is of no immediate interest to the researcher but has to be taken into consideration in the estimation of other parameters.

The need to deal with nuisance parameters, in particular the uncertainty that they add to estimation, has led to the development of variations on the likelihood function. Here we consider four such variations: (1) profile likelihoods, (2) estimated likelihoods, (3) marginal and conditional likelihoods, and (4) integrated likelihoods.[10]

[9] Notice that $\boldsymbol{\epsilon'\epsilon} = \epsilon_1^2 + \epsilon_2^2 + \cdots + \epsilon_n^2 = \sum_i \epsilon_i^2$. Taking expectations we have $\mathcal{E}[\boldsymbol{\epsilon'\epsilon}] = \sum_i \mathcal{E}[\epsilon_i^2]$. Under homoskedasticity, $\mathcal{E}[\epsilon_i^2] = \sigma_i^2 = \sigma^2$. Hence, $\sum_i \mathcal{E}[\epsilon_i^2] = \sum_i \sigma^2 = n\sigma^2$.

[10] For a more extensive discussion of these methods see Berger, Liseo and Wolpert (1999) and Pawitan (2001).

3.4.1 Profile Likelihood

Let θ be a parameter that is of theoretical interest and let η be a nuisance parameter. A likelihood function that depends on both of these parameters is called the **joint likelihood**, $L(\theta, \eta)$. We can concentrate this likelihood so that it is a function only of θ. This produces the following result.

Definition 3.12. The **profile likelihood** function is

$$\mathcal{L}_p(\theta) = \max_{\eta} L(\theta, \eta) \tag{3.13}$$

where θ is fixed.

(Other names for this likelihood function include *concentrated* and *peak* likelihood.) This function represents the likelihood of θ without reference to η. Thus, the nuisance parameter has been removed from the likelihood function, allowing us to describe the likelihood of the parameter of interest.

Example 3.5. Consider the normal PDF. Frequently, researchers are only interested in the mean of this distribution. As such, the variance may be treated as a nuisance parameter. In Example 3.1, we saw that the ML estimator for σ^2 for a given μ is equal to

$$\hat{\sigma}^2_\mu = \frac{1}{n}\sum_i (y_i - \mu)^2$$

The joint likelihood of μ and σ^2 is

$$\mathcal{L}(\mu, \sigma^2) = (2\pi)^{-.5n}(\sigma^2)^{-.5n} \exp\left(-\frac{1}{2\sigma^2}\sum_i (y_i - \mu)^2\right)$$

Substituting the formula for $\hat{\sigma}^2_\mu$ produces a profile likelihood for μ that is equal to

$$\mathcal{L}_p(\mu) = (2\pi)^{-.5n} \exp(-.5n)\left(\frac{1}{n}\sum_i (y_i - \mu)^2\right)^{-.5n}$$

Figure 3.3 shows the graph of the profile likelihood function for μ given the data that were presented in Example 3.1. We see that the apex of this graph is at 3, which is the ML estimator of μ. This suggests nice performance of the profile likelihood in this case. As we shall see below, however, profile likelihoods do not always behave so well.

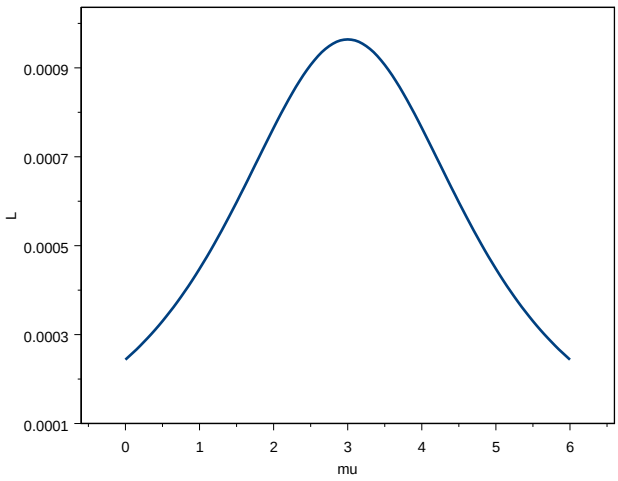

Figure 3.3: Normal Profile Likelihood for μ

Example 3.6. What would happen if we were to treat μ as the nuisance parameter of the normal PDF? While it is unusual to do this, it is possible to derive the profile likelihood for σ^2. We do so by taking the ML estimator for μ (at fixed σ^2) and substituting this into the joint likelihood function. We have seen that $\hat{\mu} = \bar{y}$. Substitution yields:

$$\mathcal{L}_p(\sigma^2) \quad = \quad (2\pi)^{-.5n}(\sigma^2)^{-.5n} \exp\left(-\frac{1}{2\sigma^2}\sum_i (y_i - \bar{y})^2\right)$$

Noting that $\sum_i (y_i - \bar{y})^2 = n\hat{\sigma}^2$, this may be rewritten as

$$\mathcal{L}_p(\sigma^2) \quad = \quad (2\pi)^{-.5n}(\sigma^2)^{-.5n} \exp\left(-\frac{n\hat{\sigma}^2}{2\sigma^2}\right)$$

The advantage of profile likelihoods is that they can always be constructed, which is not true of marginal and conditional likelihoods. The drawback is that profile likelihoods can produce severely biased and inconsistent estimates (Neyman and Scott 1948). This is particularly true if the number of nuisance parameters is large. An example can illustrate this.

Table 3.1: Simulated Data for an ANOVA Profile Likelihood

i	μ_i	y_{i1}	y_{i2}	$\bar{y}_i$
1	.845	1.264	.879	1.072
2	-.355	-.807	-.404	-.605
3	.179	-.057	.485	.214
4	-.178	-.101	-.239	-.170
5	-3.102	-2.924	-3.793	-3.358
6	-1.347	-.881	.465	-.208
7	2.243	4.259	2.288	3.274
8	.437	-.295	1.312	.508
9	1.676	1.151	2.970	2.060
10	-.993	-2.678	-1.611	-2.144
11	-2.540	-1.266	-3.455	-2.360
12	3.183	2.173	3.897	3.035
13	.948	1.770	-.238	.766
14	-.455	-2.539	-.044	-1.292
15	1.014	1.506	1.284	1.395
16	-3.061	-3.451	-1.714	-2.582
17	2.896	2.431	4.381	3.406
18	-3.802	-2.984	-3.215	-3.099
19	-.850	-2.456	-1.120	-1.788
20	6.855	6.375	6.914	6.645

Example 3.7.[11] Imagine an experiment with 20 different conditions ($N = 20$). There are two participants in each condition so that $n = 2N = 40$. We assume $y_{ij} \sim N(\mu_i, \sigma^2)$, where $i = 1 \cdots 20$ denotes a particular condition and $j = 1, 2$ denotes a participant. This assumption means that the conditions differ only in their means, not in their variance. Somewhat atypical of experimental research, our main interest is in σ^2; thus $\mu_1 \cdots \mu_{20}$ are nuisance parameters.

For purposes of demonstrating bias we simulate a data set in which μ_i is known and in which it is also known that $\sigma^2 = 1$. The simulated data are shown in Table 3.1. The second column contains the population mean, the third and fourth columns the observed data, and the last column the sample mean in each condition.

[11] This example is patterned after Pawitan (2001: 274-276).

Let $\theta = (\mu_1, \cdots, \mu_N, \sigma^2)$ be the complete set of parameters, then the full log-likelihood function is given by

$$\ell = -N\ln(2\pi) - N\ln(\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{N}\sum_{j=1}^{2}(y_{ij} - \mu_i)^2$$

Assuming that μ_i is known for all i, this is also the true log-likelihood function for σ^2. The corresponding normalized likelihood function is depicted through the blue line in Figure 3.4.

Derivation of the profile likelihood requires substitution of the ML estimators for μ_i. Since $\hat{\mu}_i = \bar{y}_i$, the profile log-likelihood function is given by

$$\begin{aligned}\ell_p(\sigma^2) &= -N\ln(2\pi) - N\ln(\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{N}\sum_{j=1}^{2}(y_{ij} - \bar{y}_i)^2 \\ &= -N\ln(2\pi) - N\ln(\sigma^2) - \frac{RSS}{2\sigma^2}\end{aligned}$$

where $RSS = \sum_i \sum_j (y_{ij} - \bar{y}_i)^2$ is the residual sum-of-squares. When we optimize this log-likelihood we obtain $\hat{\sigma}^2 = RSS/2N$. The red line in Figure 3.4 depicts the normalized profile likelihood.

As is clear from this figure, the maximum of the profile likelihood is not anywhere near the true population value of σ^2, which is 1. Indeed the probability of the profile likelihood function yielding an estimate near the true value is extremely small. By contrast, the true likelihood provides better recovery of the true value. This illustrates the serious bias that can be associated with the profile likelihood. Indeed, it can be demonstrated that $plim(\hat{\sigma}^2) = \sigma^2/2$, which demonstrates that the profile likelihood estimator is inconsistent, i.e. the bias does not disappear asymptotically (see Pawitan 2001).

3.4.2 Estimated Likelihood

The idea behind the profile likelihood is to replace the nuisance parameter with its ML estimator. But why should we restrict ourselves to the class of maximum likelihood estimators? What if we used any reasonable estimate of the nuisance parameter, regardless of how it was obtained? If we follow this approach, then we arrive at an estimated likelihood function.[12]

[12]The function is sometimes also labeled "pseudo" likelihood, although this is somewhat confusing since that term is used also for other kinds of approximations of the likelihood.

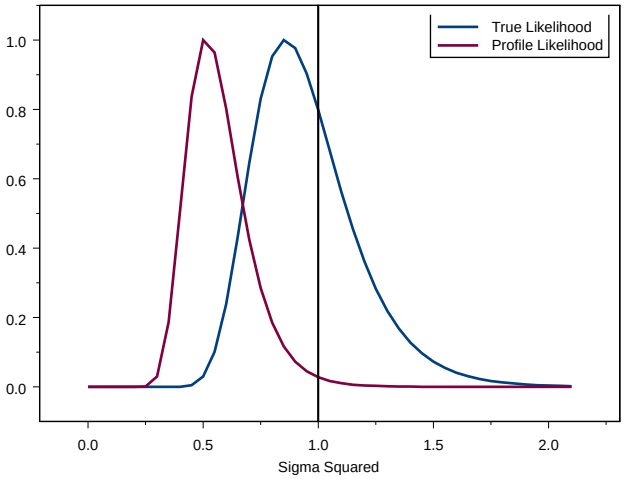

Figure 3.4: The Potential for Bias in Profile Likelihoods

Definition 3.13. Let $\hat{\eta}$ be an estimator of η—obtained through MLE or any other procedure—then the **estimated likelihood** of θ is

$$\mathcal{L}_e(\theta) \quad = \quad L(\theta, \hat{\eta}) \tag{3.14}$$

The estimated likelihood function can be maximized with respect to θ to obtain an estimate of that parameter.

Example 3.8. Consider once more the normal PDF, treating σ^2 as the nuisance parameter. The sample variance, s^2, is a widely used (and unbiased) estimator of σ^2. As you recall, its formula is

$$s^2 \quad = \quad \frac{1}{n-1}\sum_i (y_i - \bar{y})^2$$

Substituting this estimator into the joint likelihood function of μ and σ^2 yields the estimated likelihood for μ:

$$\mathcal{L}_e(\mu) \quad = \quad (2\pi)^{-.5n}\left(\frac{\sum_i (y_i - \bar{y})^2}{n-1}\right)^{-.5n} \exp\left(\frac{n-1}{2}\frac{\sum_i (y_i - \mu)^2}{\sum_i (y_i - \bar{y})^2}\right)$$

This function can be optimized to yield an ML estimator for μ.

Estimated likelihoods provide a flexible approach to dealing with nuisance parameters but they do carry a cost: they do not account for the extra uncertainty that is introduced due to the nuisance parameter. Making this adjustment can be extremely difficult, unless the estimator $\hat{\eta}$ is asymptotically equivalent to the ML estimator (see Gong and Samaniego 1981).

3.4.3 Marginal and Conditional Likelihood

Under some circumstances it is possible to obtain a transformation of the data that will allow the construction of so-called marginal and conditional likelihoods. Those likelihood functions frequently have much better statistical properties than the profile likelihood (i.e. they are consistent). The idea is to transform the data y to (v, w) in such a manner that (1) the marginal distribution of v or (2) the conditional distribution of v given w depends only on θ, the parameter of interest, and not on any nuisance parameters η.

To formalize this idea, let the total parameter be (θ, η). The joint likelihood of these two parameters is partitioned into a conditional and a marginal likelihood, taking advantage of the well-known statistical result that $f(v, w) = f(v|w)f(w) = f(w|v)f(v)$. The goal is to find a transformation so that either $f(v)$ or $f(v|w)$ is a function only of θ. In the case of marginal maximum likelihood estimation, $f(v)$ depends only on θ. To emphasize this fact, we write $f_\theta(v)$. The likelihood is then partitioned in the following way.

> ***Definition 3.14.*** The **marginal likelihood** is a function $\mathcal{L}_1(\theta)$ such that
>
> $$\begin{aligned} \mathcal{L}(\theta, \eta) &= f_{\theta,\eta}(v, w) \\ &= f_\theta(v) f_{\theta,\eta}(w|v) \\ &= \mathcal{L}_1(\theta)\mathcal{L}_2(\theta, \eta) \end{aligned}$$

Optimization of $\mathcal{L}_1(\theta)$ yields an ML estimate of θ.

In the case of conditional maximum likelihood estimation, $f(v|w)$ depends only on θ so that we write $f_\theta(v|w)$. Here partitioning of the likelihood proceeds as follows.

> ***Definition 3.15.*** The **conditional likelihood** is a function

$\mathcal{L}_1(\theta)$ such that

$$\begin{aligned}\mathcal{L}(\theta,\eta) &= f_{\theta,\eta}(v,w)\\ &= f_\theta(v|w)f_{\theta,\eta}(w)\\ &= \mathcal{L}_1(\theta)\mathcal{L}_2(\theta,\eta)\end{aligned}$$

Again, optimization of $\mathcal{L}_1(\theta)$ yields an ML estimate of θ. Note that if v and w are independent, then the marginal and conditional likelihood functions are the same (since $f(v|w) = f(v)$).

It should be emphasized that marginal and conditional likelihood estimation involve an approximation, since $\mathcal{L}_2(\theta,\eta)$ is ignored. One should opt for this approximation only under specific circumstances (see Pawitan 2001). Specifically, marginal and conditional MLE are useful if

1. Full likelihood or profile likelihood estimation produces inconsistent estimators.
2. Not much information is lost by ignoring $\mathcal{L}_2(\theta,\eta)$.
3. $f_\theta(v)$ or $f_\theta(v|w)$ is simpler than $f_{\theta,\eta}(y)$.

The first condition is absolutely essential, since there is no need to go to conditional and marginal MLE if consistent full MLE is available. The second condition is also of considerable importance, although Pawitan (2001) notes that this is typically approached in an informal manner as opposed to rigid proof.

> **Example 3.9.**[13] Consider again the experimental data from Example 3.7 and the problem of estimating σ^2. The question is whether we can transform the data in such a way that conditional or marginal MLE is feasible. One useful transformation is[14]
>
> $$\begin{aligned}v_i &= (y_{i1}-y_{i2})/\sqrt{2}\\ w_i &= (y_{i1}+y_{i2})/\sqrt{2}\end{aligned}$$

[13] This example is based on Pawitan (2001: 279) but includes additional proofs.

[14] This is an appropriate transformation because it is possible to recover the original data. For example, $(v_i+w_i)/\sqrt{2} = y_{i1}$ and $(w_i-v_i)/\sqrt{2} = y_{i2}$.

To see why these transformations are useful consider the statistical properties of v_i. First, since v_i consists of the difference of two normally distributed variables it, too, follows a normal distribution. Second, $\mathcal{E}[v_i] = 0$.[15] This is critical. Remember, in this example μ_i are the nuisance parameters and we want a transformation that gets rid off those parameters. v_i is one such transformation since its mean is not a function of μ_i. Third, $V[v_i] = \sigma^2$, i.e. v_i has the same variance as y_{ij}.[16] That, too, is critical because we want the distribution of v_i to be a function of the parameter of interest. In sum, $v_i \sim N(0, \sigma^2)$. (Likewise it is easy to demonstrate that $w_i \sim N(\sqrt{2}\mu_i, \sigma^2)$, although this is of less immediate interest.)

The marginal log-likelihood function, $\ln[f_{\sigma^2}(v)]$, is given by

$$l_1(\sigma^2) \quad = \quad -.5N\ln(2\pi) - .5N\ln(\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{N} v_i^2$$

Optimization of this function yields $\hat{\sigma}^2 = \sum_{i=1}^{N} v_i^2/N = RSS/N$.[17] Figure 3.5 displays the normalized marginal likelihood function along with the profile likelihood that we derived earlier (both are normalized) using the data from Table 3.1. As is clear from this picture, the marginal likelihood function has much better coverage of the true parameter value ($\sigma^2 = 1$) then the profile likelihood. This illustrates the comparative advantage of marginal MLE over profile MLE.

In this example, the conditional likelihood function is identical to the marginal likelihood function, since v and w are independent.[18] So the

[15]The proof is straightforward: $\mathcal{E}[v_i] = \mathcal{E}[(y_{i1} - y_{i2})/\sqrt{2}] = (1/\sqrt{2})(\mathcal{E}[y_{i1} - \mathcal{E}[y_{i2}]) = (1/\sqrt{2})(\mu_i - \mu_i) = 0$.

[16]The proof is easy and depends on the variance algebra for linear composites. Specifically, $V[v_i] = V[(y_{i1} - y_{i2})/\sqrt{2}] = (1/\sqrt{2})^2 V[y_{i1}] + (1/\sqrt{2})^2 V[y_{i2}] + 2(1/\sqrt{2})(1/\sqrt{2})Cov[y_{i1}, y_{i2}]$. The last term drops out since y_{i1} and y_{i2} are statistically independent and their covariance is consequently zero. The remainder of the expression simplifies to $.5V[y_{i1}] + .5V[y_{i2}] = .5\sigma^2 + .5\sigma^2 = \sigma^2$.

[17]The demonstration that $\sum_{i=1}^{N} v_i^2 = RSS = \sum_{i=1}^{N}[(y_{i1} - \bar{y}_i)^2 + (y_{i2} - \bar{y}_i)^2]$ is straightforward. Substituting $v_i = (y_{i1} - y_{i2})/\sqrt{2}$ and expanding the expression for v_i^2 we get $.5y_{I1}^2 - y_{i1}y_{i2} + .5y_{i2}^2$. This is the same result that we get when we expand $(y_{i1} - \bar{y}_i)^2 + (y_{i2} - \bar{y}_i)^2$ after substituting $.5(y_{i1} + y_{i2})$ for $\bar{y}_i$.

[18]For the normal pdf, independence can be demonstrated by showing that v and w are uncorrelated, which in turn implies that their covariance is zero. We know that $Cov(v, w) = \mathcal{E}[vw] - \mathcal{E}[v]\mathcal{E}[w]$. Since $\mathcal{E}[v] = 0$ the last term drops out. For the first term it is easily demonstrated that $vw = .5(y_{i1}^2 - y_{i2}^2)$. The expectation is $\mathcal{E}[vw] = .5(\mathcal{E}[y_{i1}^2] - \mathcal{E}[y_{i2}^2])$.

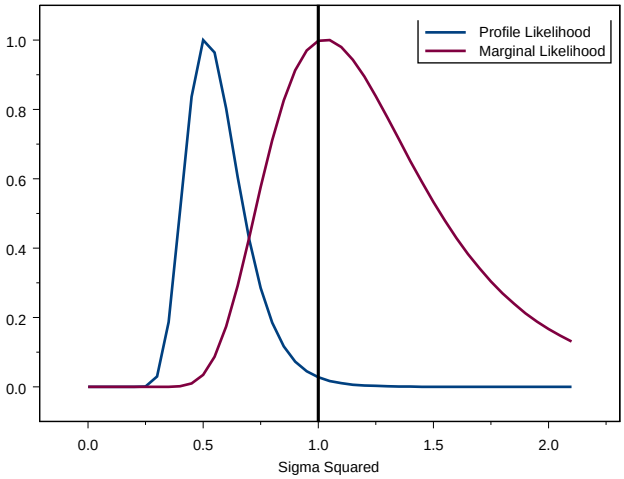

Figure 3.5: Normal Marginal Likelihood for σ^2

conditional MLE is also $\hat{\sigma}^2 = RSS/N$. Note that it is not generally the case that marginal and conditional likelihood functions are the same, so both procedures can yield different estimators.

The advantage of conditional and marginal MLE is that it can produce better estimators than the profile likelihood. However, conditional and marginal MLE are not always available. They are generally available for the **exponential family** of distributions, which includes the binomial, exponential, gamma, normal, and Poisson distributions. Outside of this family, availability of conditional and marginal MLE may be more problematic.

3.4.4 Integrated Likelihood

Berger, Liseo, and Wolpert (1999) advocate the use of integrated likelihood functions for the elimination of nuisance parameters. Integrated likelihoods are common in Bayesian statistics. As such they appeal to advocates of both classical and Bayesian models of inference.

We know that $\mathcal{E}[y_{ij}^2] = \sigma^2 + \mu_i^2$. Substitution yields $\mathcal{E}[vw] = .5[\sigma^2 + \mu_i^2 - (\sigma^2 + \mu_i^2)] = 0$. This concludes the proof that v and w are independent.

In contrast to the profile likelihood, integrated likelihoods do not maximize over the nuisance parameter but instead average over it. For the two-parameter case, this means the following.[19]

> ***Definition 3.16.*** The (uniform) **integrated likelihood** function is
>
> $$\mathcal{L}_{int}(\theta) \quad = \quad \int L(\theta, \eta) d\eta \tag{3.15}$$
>
> where η as usual is the nuisance parameter and θ is the parameter of interest.

The integrated likelihood function may be approximated using Laplace's integral approximation

$$\mathcal{L}_{int}(\theta) \quad \approx \quad c\mathcal{L}(\theta, \hat{\eta}_\theta) \left|\mathcal{I}_o(\hat{\eta}_\theta)\right|^{-.5}$$

where $\mathcal{I}_o$ is the observed Fisher information and c is free of θ. This approximation, which is highly accurate if the likelihood function is approximately quadratic, may also be stated in terms of the log-likelihood function:

$$\ell_{int}(\theta) \quad \approx \quad \ell_p(\theta) - .5 \ln \left|\mathcal{I}_o(\hat{\eta}_\theta)\right|$$

Here c has been dropped, since it is not a function of θ and will therefore not affect estimation of this parameter. Further, $\ell_p(\theta)$ is the profile likelihood function for θ.

> **Example 3.10.** Consider again the normal PDF where μ is now treated as the nuisance parameter. Based on Example 3.6, the profile log-likelihood function for σ^2 is
>
> $$\ell_p(\sigma^2) \quad = \quad -.5n \ln(2\pi) - .5n \ln(\sigma^2) - \frac{n\hat{\sigma}^2}{2\sigma^2}$$
>
> From earlier results, we also know
>
> $$\mathcal{I}_o(\hat{\mu}_{\sigma^2}) \quad = \quad \frac{n}{\sigma^2}$$

[19] See Pawitan (2001) for a multi-parameter formulation.

Thus, the integrated likelihood is given by

$$\begin{aligned}\ell_{int}(\sigma^2) &\approx -.5n\ln(2\pi) - .5n\ln(\sigma^2) - \frac{n\hat{\sigma}^2}{2\sigma^2} - .5\ln\left|\frac{n}{\sigma^2}\right| \\ &\approx -\frac{n\ln(2\pi)+\ln(n)}{2} - \frac{n-1}{2}\ln(\sigma^2) - \frac{n\hat{\sigma}^2}{2\sigma^2}\end{aligned}$$

Berger, Liseo, and Wolpert (1999) maintain that integrated likelihoods offer a number of advantages over profile likelihoods. First, they argue that integration works better when the likelihood function has sharp ridges. Second, they claim that integrated likelihood functions are better at accounting for parameter uncertainty than profile likelihoods. The reason is that $.5\ln|\mathcal{I}_o(\hat{\eta}_\theta|$ can be viewed as a penalty term that eliminates from the profile log-likelihood function the information that is gleaned from the nuisance parameter. However, integrated likelihood functions can be quite complex and the Laplace approximation does not work well under all circumstances. Thus, integrated likelihood estimation may not always be feasible.

As a final comment, it is useful to point out that integrated likelihood functions are closely associated with so-called **modified profile likelihoods**. These likelihoods can themselves be thought of as approximations of marginal or conditional likelihoods. As such, modified profile likelihoods seek to eliminate the inconsistencies that are associated with ordinary profile likelihoods. For the two-parameter case, the modified profile log-likelihood function is given by

$$\ell_m(\theta) = \ell_p(\theta) - .5\ln|\mathcal{I}_o(\hat{\eta}_\theta)| + \ln\left|\frac{\partial\hat{\eta}}{\partial\hat{\eta}_\theta}\right|$$

Here $|\partial\hat{\eta}/\partial\hat{\eta}_\theta|$ is the Jacobian, which is an invariance-preserving quantity (Pawitan 2001). This function is almost identical to the integrated log-likelihood. Indeed, it will be identical if θ and η are *information orthogonal*, which is the case when these parameters are independent (as is true of the normal distribution). In this case, $\hat{\eta}_\theta = \hat{\eta}$ and $|\partial\hat{\eta}/\partial\hat{\eta}_\theta| = 1$, which causes the last term in ℓ_m to drop out.[20] Such is the case with μ in Example 3.10, so that the integrated and modified profile likelihood functions are identical in this example.

[20]In non-orthogonal cases, the Jacobian may be very difficult to evaluate. Pawitan (2001) describes possible transformations that will ensure information orthogonality.

Chapter 4

Statistical Properties

One of the nice features of MLE is that it possess known statistical properties (unlike, for instance, least squares estimation where the properties have to be established on a case-by-case basis). These properties include invariance, consistency, asymptotic efficiency, and asymptotic normality. With the exception of efficiency, I shall not attempt to prove those properties (proofs can be found in Wilks [1962], among others). Rather, my goal is to sketch the importance of these properties for applications of MLE. First, however, I should sketch the conditions under which these properties hold true.

4.1 Regularity and Sample Size Conditions

To derive the properties of ML estimators, statisticians usually impose a mild set of so-called **regularity conditions**. These conditions are quite technical, but the most important ones are: (1)the first three derivatives of the log-likelihood function are finite, (2) the expectations of the first and second derivatives can be computed, (3) the expected Fisher information matrix is positive definite and finitely bounded, and (4) the number of parameters remains finite as n goes to positive infinity.[1] Fortunately, these conditions hold true under many circumstances.

A second point to remember is that, for the most part, ML estimators have **asymptotic** properties. As such they assume an infinite sample size (i.e. $n \to \infty$). By virtue of the law of large numbers we can be reason-

[1]For a more detailed discussion of regularity conditions, see e.g. Greene (2003) or Wilks (1962).

ably sure that the properties also hold in large samples. However, when the sample size becomes small then one of two things may occur: (1) the small sample properties of the ML estimator are unknown, or (2) the small sample properties of the ML estimator are known to be undesirable.[2] This is one reason why statisticians often admonish that MLE should only be used with large samples.

4.2 Invariance

Invariance means that any function of an ML estimator is itself an ML estimator.[3] More formally:

> ***Definition 4.1.*** If $\hat{\theta}$ is the ML estimator of θ and $g(\theta)$ is a function of θ, then $g(\hat{\theta})$ is also an ML estimator, namely of $g(\theta)$.

There are no restrictions on the function, $g(\theta)$; it could be linear or nonlinear. So $k\hat{\theta}$, with k being a constant, is an ML estimator, but so is $\ln[\hat{\theta}/(1-\hat{\theta})]$.

The latter transformation is known as the log-odds or **logit** transformation. It turns out to be particularly useful in terms of obtaining a more regularly shaped likelihood function, in the sense of approaching a quadratic function. As discussed previously (Chapter 2.3), quadratic log-likelihoods offer a number of advantages so that there may be good reason to transform ML estimators. The following example illustrates the logit transformation and the invariance property.

> **Example 4.1.** Consider the binomial distribution with $n = 20$ and $y = 16$. The log-likelihood function is given by
>
> $$\ell \;=\; \ln\binom{20}{16} + 16\ln(\pi) + 4\ln(1-\pi)$$

[2] We have seen several examples of this in Chapter 3. For instance, the ML estimator of the variance of the normal PDF is biased in small samples (see Example 3.1), while the same is true of the ML estimator of the error variance in the classical linear regression model (see Chapter 3.3).

[3] The invariance property does not automatically apply to other estimators. For example, if $\hat{\theta}$ is a least squares estimator than $g(\hat{\theta})$ is not necessarily a least squares estimator.

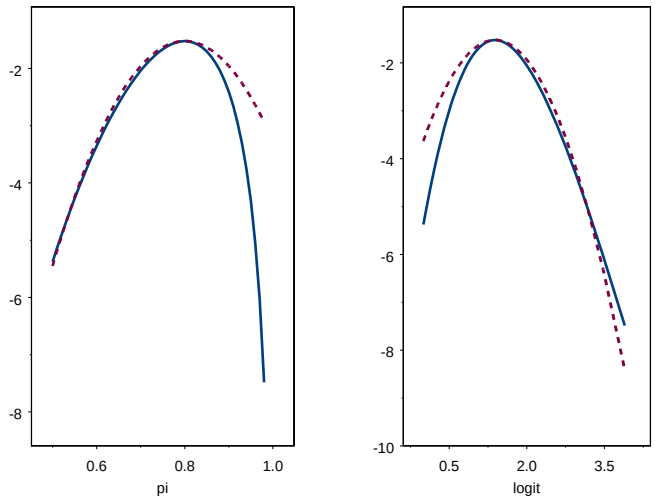

Figure 4.1: The Logit Transformation of a Binomial Log-Likelihood

This function is depicted as the blue line in the left panel of Figure 4.1, along with a dashed quadratic reference line (which reaches its maximum in the same place as the log-likelihood function, namely at $\hat{\pi} = .8$, and has the same amplitude in that location). We see that the quadratic reference line fits the log-likelihood nicely to the left of the ML estimate. To the right of this value, however, the fit is very poor. The log-likelihood drops very rapidly and as such deviates dramatically from the quadratic reference line.

What happens if we perform a logit transformation on $\hat{\pi}$? The transformation is given by

$$\psi \quad = \quad \ln\left(\frac{\pi}{1-\pi}\right)$$

Graphing the log-likelihood function associated with ψ turns out to be very simple because the invariance principle implies that $\psi = g(\pi)$ has a likelihood of $\mathcal{L}(\pi)$, the same function whose logarithm is depicted in the left-hand panel of Figure 4.1. The log-likelihood of ψ is depicted as the blue line in the right-hand side panel of this figure along with a dashed quadratic reference line. We see that $\ell(\psi)$ much better approximates a quadratic function than $\ell(\pi)$. Hence, it might

be preferable to use the logit transformation.[4]

When we use a transformation of an ML estimator, then the standard error will have to be adjusted. For an ML estimator $\hat{\theta}$ with standard error $se(\hat{\theta})$ that is transformed using $g(\hat{\theta})$ we have the following result:

$$se[g(\hat{\theta})] = se(\hat{\theta}) \left| \frac{\partial g}{\partial \theta} \right| \tag{4.1}$$

Hence computation of the standard error of $g(\hat{\theta})$ requires that we know the standard error of $\hat{\theta}$ as well as the first partial derivative of $g(\theta)$ with respect to θ.

Example 4.2. What is the standard error of the logit transformation of π in the binomial distribution? We begin by deriving the standard error for $\hat{\pi}$. For the binomial log-likelihood

$$\ell = \ln \binom{n}{y} + y \ln(\pi) + (n - y) \ln(1 - \pi)$$

the second derivative (Hessian) is

$$\ell'' = -\frac{n - y}{(1 - \pi)^2} - \frac{y}{\pi^2}$$

The expected Fisher information that is associated with this Hessian is

$$-E[\ell''] = \frac{n - y}{(1 - \pi)^2} + \frac{y}{\pi^2}$$

The estimated variance of $\hat{\pi}$ is the inverse of the expected Fisher information with the ML estimator substituted for π:

$$\hat{V}[\hat{\pi}] = \frac{(1 - \hat{\pi})^2}{n - y} + \frac{\hat{\pi}^2}{y}$$

The standard error for $\hat{\pi}$ is the square root of this expression.

Next we evaluate $|\partial g / \partial \pi|$ for $g(\pi) = \ln[\pi/(1 - \pi)]$:

$$\left| \frac{\partial g}{\partial \pi} \right| = \frac{1}{\pi - \pi^2}$$

[4]The utility of this approximation is greatest as π approaches 0 or 1. If $\pi \approx .5$ then the binomial likelihood is regularly shaped (see e.g. Figure 2.1).

This is what we use to multiply $se(\hat{\pi})$.

For our example we have $n = 20$ and $y = 16$. We also know that $\hat{\pi} = .8$.[5] Thus $\hat{V}[\hat{\pi}] = .05$. Further, $|\partial g/\partial \pi| = 1/(.8 - .8^2) = 6.25$. Hence $se(\hat{\psi}) = 6.25\sqrt{.05} = 1.398$.

4.3 Sufficiency

It can be demonstrated that ML estimators are minimally sufficient, a property that is defined in the following manner.

> ***Definition 4.2.*** A **minimally sufficient** statistic is one that is itself a function of another sufficient statistic.

A statistic is sufficient if it exhausts all of the information in the sample that is relevant to some parameter θ. More formally,

> ***Definition 4.3.*** A statistic T is **sufficient** if the conditional distribution of the sample data, $f(y_1 \cdots y_n|T)$, does not depend on θ.

It is helpful to know that a statistic is sufficient because we then know that all relevant information about a parameter has been taken into consideration. Thus, the fact that ML estimators are minimally sufficient is of considerable interest.

> **Example 4.3.** To illustrate the notion of sufficiency consider the normal PDF. If σ^2 is known, then $\sum_i y_i$ is sufficient for μ. After all, the ML estimator is $\hat{\mu} = (1/n)\sum_i y_i$ and $\sum_i y_i$ does not contain μ.[6] By similar reasoning, if μ is known, then $\sum_i (y_i - \mu)^2$ is sufficient for σ^2, since the ML estimator is $\hat{\sigma}^2 = (1/n)\sum_i (y_i - \mu)^2$ and $\sum_i (y_i - \mu)^2$ does not depend on σ^2. In practice, neither μ nor σ^2 is known. In this situation, $(\sum_i y_i, \sum_i y_i^2)$ are *jointly sufficient* for μ and σ^2. That is to say, knowing these two sums we can obtain information about μ and

[5] The first derivative of ℓ is equal to $-(n-y)/(1-\pi) + y/\pi$. Setting this to zero yields $\hat{\pi} = y/n$.

[6] This illustrates an important point: a sufficient statistic does not have to be an estimator. $\sum_i y_i$ is not an estimator, although $\hat{\mu} = (1/n)\sum_i y_i$ is.

σ^2 and those two sums do not themselves depend on the parameters of the normal PDF. Notice that sufficiency implies that all we need to know is $\sum_i y_i$ and $\sum_i y_i^2$; the rest of the data does not add any additional information about μ or σ^2.

4.4 Consistency

ML estimators are consistent. This means that the probability that the estimator differs from the population parameter(s) by some arbitrarily small amount will tend to zero as the sample size goes to infinity. More formally,

Definition 4.4. ML estimators are **consistent** in that

$$plim\ \hat{\boldsymbol{\theta}} \;=\; \boldsymbol{\theta} \tag{4.2}$$

where *plim* stands for probability limit.

This result assumes that the model that is being estimated is correctly specified.

A sufficient (but not a necessary) condition for consistency is so-called square error consistency.

Definition 4.5. An estimator is **square error consistent** if the mean square error (MSE) tends to zero when the sample size approaches positive infinity.

Since the MSE consists of the squared bias of an estimator and that estimator's variance, square error consistency implies that the bias disappears asymptotically and that the variance approaches zero. Note that there is no guarantee that ML estimators are unbiased in small samples; square error consistency implies only that any bias will disappear asymptotically.

Example 4.4. Consider the ML estimator of σ^2 in the normal PDF. As stated in Example 3.1, this estimator is biased. Recall that

$$\hat{\sigma}^2 \;=\; \frac{\sum_i (y_i - \bar{y})^2}{n}$$

The expected value of this estimator is

$$\mathcal{E}[\hat{\sigma}^2] = \sigma^2 - \frac{\sigma^2}{n}$$

There is bias because this expectation is not equal to σ^2. The degree of the bias is

$$B = -\frac{\sigma^2}{n}$$

so that the ML estimator underestimates the true variance, especially in small samples. The MSE of $\hat{\sigma}^2$ is given by

$$\begin{aligned} MSE[\hat{\sigma}^2] &= \left(-\frac{\sigma^2}{n}\right)^2 + \frac{2\sigma^4}{n} \\ &= \frac{1+2n}{n^2}\sigma^4 \end{aligned}$$

It is clear that $(1+2n)/n^2 \to 0$ as $n \to \infty$, which means that $MSE \to 0$. Thus $\hat{\sigma}^2$ is square error consistent.

4.5 Efficiency

ML estimators are efficient. This result follows from the **Cramér-Rao lower-bound** (CRLB) on the variance.

Definition 4.6. Let $T(y)$ be a statistic with expectation $\mathcal{E}[T(y)] = \theta$, i.e. the statistic is an unbiased estimator of a the population parameter. Then the CLRB is given by

$$V[T(y)] \geq \frac{1}{\mathcal{I}_e} \tag{4.3}$$

where $\mathcal{I}_e$ is the expected Fisher information.

Since we have seen that the variance of an ML estimator is equal to $\mathcal{I}_e^{-1}$ it follows immediately that ML estimators are efficient, i.e. they have the smallest possible variance of any unbiased estimator since they are at the CLRB.

It is important to emphasize that the CLRB assumes that a statistic is an unbiased estimator of the parameter of interest. Of course, ML estimators

may be biased in small samples, as we have just seen. Due to their consistency, however, we know that they are asymptotically unbiased. Thus, the CLRB may come into play only as $n \to \infty$ so that it is perhaps safer to say that ML estimators are **asymptotically efficient**.

> **Example 4.5.** Consider again the Poisson distribution with parameter $\mu > 0$. Earlier we demonstrated that $\hat{\mu} = \bar{y}$ is the ML estimator (see Example 2.4). We also demonstrated that $l'' = -\sum_i y_i/\mu^2$. The variance of our estimator is given by the inverse of the expected Fisher information. The expected Fisher information is $-E[l''] = n/\mu$ (see Example 2.7). The inverse of the expected Fisher information is μ/n. This is the CLRB and is also the variance of $\bar{y}$.[7]

The CLRB can be generalized to situations in which $\mathcal{E}[T(y)] = g(\theta)$, i.e. the expectation of the statistic is some function of the true parameter value. In this case, the CLRB can be stated as:

$$V[T(y)] \geq \frac{[g'(\theta)]^2}{\mathcal{I}_e}$$

where $g'(\theta)$ is the first derivative of $g(\theta)$ with respect to θ. This result can be generalized to the multivariate case. Let $\boldsymbol{\theta}$ be a vector of parameters and let $T(y)$ be a scalar function of the data such that $\mathcal{E}[T(y)] = g(\boldsymbol{\theta})$, then

$$V[T(y)] \geq \boldsymbol{\alpha}' \boldsymbol{\mathcal{I}}_e^{-1} \boldsymbol{\alpha}$$

where $\boldsymbol{\alpha} = \partial g(\boldsymbol{\theta})/\partial \boldsymbol{\theta}$ and $\boldsymbol{\mathcal{I}}_e$ is the expected Fisher information.

4.6 Normality

ML estimators are asymptotically normally distributed. This result can be derived in a number of ways (Le Cam 1970), for example in the context of quadratic log-likelihoods. It carries important implications for hypothesis testing, as we shall see later. However, it should be stressed that normality is an asymptotic result. In small samples, it should not be assumed that ML estimators follow a normal distribution.

[7] Technically, we would also have to demonstrate that $\bar{y}$ is an unbiased estimator of μ. This is done quite easily by taking the expectation of the MLE: $\mathcal{E}[\bar{y}] = \mathcal{E}[(1/n)\sum_i y_i] = (1/n)\sum_i \mathcal{E}[y_i] = (1/n)\sum_i \mu = \mu$.

Chapter 5

Numerical Optimization Methods

The key to MLE is that we maximize the log-likelihood function by setting the first (partial) derivative(s) equal to 0 and solving for the parameter(s). In the previous chapters this could be done analytically, i.e. through the application of calculus and algebra. Unfortunately, this is usually not possible because the derivatives are typically quite complex and do not have a closed-form solution. This occurs, for example, if the derivatives are not linear in the parameters, as is true of many of the applications that we shall consider.

In the many instances where the likelihood equation cannot be solved algebraically, ML estimators are obtained through numerical optimization methods. These are computer intensive procedures that solve the likelihood equation by updating successive "guesses" about the solution. More precisely,

> ***Definition 5.1.*** **Numerical optimization** is the process by which a computer algorithm continuously refines an initial numerical guess of the parameter(s), until the maximum of the objective function has been found (within tolerable error).

The qualifier "within tolerable error" suggests that numerical optimization may not yield solutions for which $\boldsymbol{\nabla} = \mathbf{0}$ holds exactly. Rather, the solutions may put the first derivative very close to zero, with the discrepancy being so small that it is of little practical concern. I shall say more about this in Chapter 5.8.

In this chapter, we shall encounter some of the most widely used numerical optimization algorithms, with an emphasis on so-called hill-climbing algorithms. These include the method of steepest ascent, the Newton-Raphson algorithm, Fisher scoring, and various so-called quasi-Newton methods that are designed to lower the computational burden and some other shortcomings of the Newton-Raphson algorithm. The appendix discuss the EM-algorithm, which is well-suited for handling missing data in estimation problems, as well as numerical optimization in Stata.[1]

5.1 Hill-Climbing Algorithms

To get a better sense of the logic of numerical methods, let us start by considering the problem of estimating a single parameter,θ. We could start by making an initial guess, θ_0. We call this the **starting value** or the **initial value**. More likely than not, this value does not maximize the log-likelihood function. Thus, we need to make an adjustment, ξ_0, so that we obtain a new estimate, $\theta_1 = \theta_0 + \xi_0$. We keep repeating this process until we have maximized the log-likelihood function, at which point we stop. The production of each new estimate is called an **iteration** of the algorithm. For the tth iteration we have

$$\theta_{t+1} \quad = \quad \theta_t + \xi_t$$

where θ_{t+1} is the parameter value that will be used in the next iteration. The situation for a multi-parameter estimation problem is much the same, except that we now have a vector of estimates and corresponding adjustments:

$$\boldsymbol{\theta}_{t+1} \quad = \quad \boldsymbol{\theta}_t + \boldsymbol{\xi}_t \tag{5.1}$$

he trick, of course, is how to define the adjustment. In MLE, **hill-climbing algorithms** are the most common numerical optimization methods. These algorithms define adjustments in terms of the gradient, i.e.

[1]The present discussion of numerical optimization methods only scratches the surface. For a much more extensive discussion of the computational aspects of numerical algorithms see Thisted (1988), Nocedal and Wright (1999), and Fletcher (2000). While the focus in this report is on hill-climbing algorithms, it should also be noted that several alternatives exist, including grid search, Monte-Carlo Markov Chain (MCMC) methods (e.g. the Gibbs sampler), genetic algorithms, and simulated annealing (e.g. the Metropolis-Hastings algorithm).

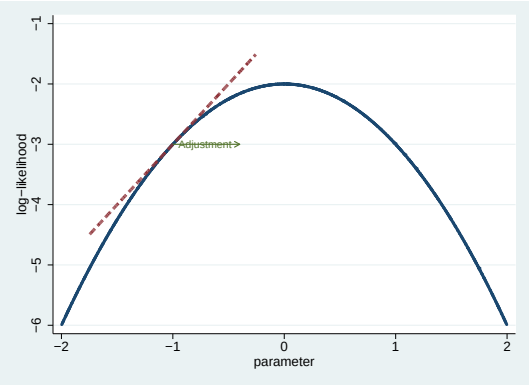

Figure 5.1: Updates of Estimates in Hill-Climbing Algorithms

$\boldsymbol{\xi} = f(\boldsymbol{\nabla})$. It makes a great deal of sense to consider the gradient while updating estimates. As the name implies, the gradient indicates something about the direction in which the log-likelihood function is changing (like the grade of a road). If the gradient is positive, this means that the log-likelihood function increases with values of the parameter. Put differently, the slope is positive, so an increase in the parameter causes an increase in the log-likelihood function. Since our goal is to maximize this function, a logical adjustment would be to increase the estimate—we would be walking/stepping up the log-likelihood function, so to speak. We would continue to do so until the gradient would turn negative. After all, a negative gradient means that the log-likelihood function decreases with values of the parameter. The slope is negative, which means that an increase in the parameter causes the log-likelihood to decrease. We want to avoid this and therefore we would decrease our estimate. Thus, hill-climbing algorithms work with variations on this simple rule: (1) if $\boldsymbol{\nabla} > \mathbf{0}$ then $\boldsymbol{\xi} > \mathbf{0}$; (2) if $\boldsymbol{\nabla} < \mathbf{0}$ then $\boldsymbol{\xi} < \mathbf{0}$ and (3) if $\boldsymbol{\nabla} = \mathbf{0}$ then $\boldsymbol{\xi} = \mathbf{0}$. The algorithm is illustrated in Figure 5.1.

We obtain greater insight into the different variations of the hill-climbing algorithm by dissecting the adjustment factor $\boldsymbol{\xi}$. In general, this factor has three components: (1) a step size, λ, (2) a direction matrix, $\boldsymbol{D}$, and (3) the gradient, $\boldsymbol{\nabla}$. Thus, the generic hill-climbing algorithm can be stated as follows.

Definition 5.2. In the generic **Hill-climbing algorithm** pa-

rameter estimates are updated via the following rule

$$\boldsymbol{\theta}_{t+1} \quad = \quad \boldsymbol{\theta}_t + \lambda_t \boldsymbol{D}_t \boldsymbol{\nabla}_t \tag{5.2}$$

Before embarking on a detailed discussion of the different variants of the algorithm, it is important to dwell a little on the different components of the adjustment factor, $\boldsymbol{\xi}$. As we have seen, the gradient, $\boldsymbol{\nabla}$, influences in what direction the parameter estimates should be updated. Should they be increased or decreased? The **direction matrix**, $\boldsymbol{D}$, controls by how much the parameter estimates should change. Should they be adjusted by just a little or by a lot? The product $\boldsymbol{d} = \boldsymbol{D}\boldsymbol{\nabla}$ is known as the **direction vector** and plays a major role in the iterative process.

This leaves us with the **step size**, λ. This parameter plays a dual role in the iterative process. First and most importantly, it ensures that the log-likelihood function increases at successive iterations. Second, it helps to direct the iterative process without having to recompute the direction vector. Computation of the direction vector requires computer cycles and helps to slow down iterations. This is so because the direction vector requires, at a minimum, computation of the first derivatives at the current estimate of $\boldsymbol{\theta}$. By contrast, manipulating the step size involves a simple scalar multiplication, which does not require much in the way of computational resources. Manipulation of the step size typically adheres to the following pattern:

1. Start by setting $\lambda = 1$.
2. If $\ell(\boldsymbol{\theta}_{t+1}) > \ell(\boldsymbol{\theta}_t)$, then increase the step size to $\lambda = 2$.
3. Continue to increase the step size by one unit as long as $\ell(\boldsymbol{\theta}_{t+1}) > \ell(\boldsymbol{\theta}_t)$.
4. If $\ell(\boldsymbol{\theta}_{t+1}) \leq \ell(\boldsymbol{\theta}_t)$, then back up and decrease the step size to $\lambda = .5$, $\lambda = .25$, or an even smaller value.

By manipulating the step size in this way, it may not be necessary to recompute the direction vector all that often, which will significantly reduce computational costs. Note that a step size parameter is not always included in the algorithm.

The place where hill-climbing algorithms differ is in the specification of the direction matrix. Table 5.1 shows common variants of the hill-climbing algorithm and their associated choices of $\boldsymbol{D}$. We shall now discuss these different algorithms in detail and discuss their strengths and weaknesses.

Table 5.1: Different MLE Algorithms

Algorithm	Direction Matrix
Steepest Ascent	Identity
Newton-Raphson	$(-\boldsymbol{\mathcal{H}})^{-1}$
Fisher Scoring	$\boldsymbol{\mathcal{I}}_e^{-1}$
BHHH	$(\sum_i \boldsymbol{\nabla}_i \boldsymbol{\nabla}_i')^{-1}$
DFP+BFGS	Arc Hessian

5.2 The Method of Steepest Ascent

In the method of steepest ascent, $\boldsymbol{D} = \mathbf{I}$, the identity matrix. Thus, the direction vector is identical to the gradient. Consequently,

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \lambda_t \boldsymbol{\nabla}_t \tag{5.3}$$

where $\boldsymbol{\nabla}_t$ is equal to the vector of first partial derivatives of the log-likelihood function evaluated at $\boldsymbol{\theta}_t$.[2]

The method of steepest ascent is the oldest and simplest of the hill-climbing algorithms. Implementation of this algorithm requires only computation of the gradient. This relative computational simplicity is one of the major advantages of the algorithm. However, it comes at a considerable cost. Because there is no informative direction matrix, the algorithm can easily make adjustments that are too large, thus "overshooting" the target. This means that the algorithm may bounce around a lot, thus slowing down convergence (see Chapter 5.8 for a discussion of convergence). In addition, the algorithm is very sensitive to the scaling of variables because this can have a large effect on the gradient.

The many problems associated with the method of steepest ascent mean that, nowadays, this algorithm is rarely used in MLE. When it is used, it is usually as a backup for the Newton-Raphson algorithm in cases where it fails due to non-concavity (see the discussion of Stata in Chapter 5.9). We now turn to a discussion of Newton-Raphson.

[2] A variation is the method of steepest descent, which simply use the negative value of the gradient to update estimates.

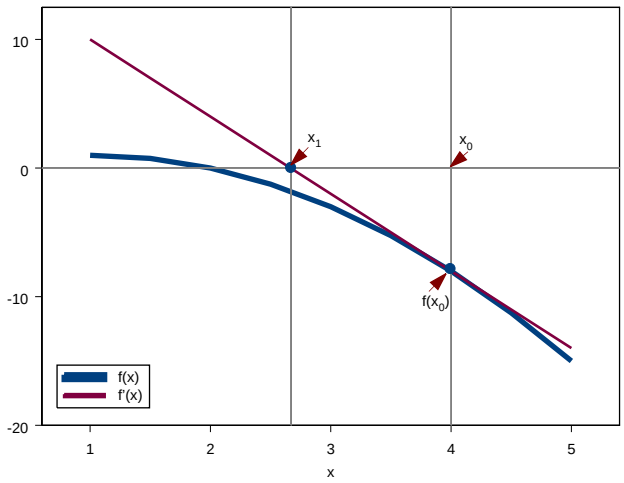

Figure 5.2: Newton's Root Finding Method

5.3 The Newton-Raphson Algorithm

The Newton-Raphson algorithm is one of the most widely used numerical procedures for finding ML estimates. The algorithm finds its origins in Newton's root finding method. It offers distinctive advantages over the method of steepest ascent and forms the building block for the quasi-Newton algorithms that will be discussed in the next section.

5.3.1 Newton's Root Finding Method

Imagine that we have a function $f(x)$. We seek to establish for which value of x the equation $f(x) = 0$ holds true. Sir Isaac Newton proposed that we can obtain the solution by starting with an initial guess x_0.Unless we are extremely lucky, x_0 will not be an actual solution (i.e. $f(x_0) \neq 0$). The question is then how we should improve our guess so that we would get closer to the truth. Newton suggested the following approach, which is illustrated in Figure 5.2:

1. Compute $f(x_0)$.
2. Compute the tangent of the function at x_0, which is given by $f'(x_0)$.

3. Extend the tangent so that it crosses the x-axis. Use the crossing point as the new guess, x_1, and repeat the steps.

The value of x_1 can be computed quite easily. Since

$$f'(x_0) = \frac{f(x_0)}{x_0 - x_1}$$

it follows that

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$$

In general, for the tth iteration of the process we have

$$x_{t+1} = x_t - \frac{f(x_t)}{f'(x_t)} \tag{5.4}$$

In general, relatively few iterations are required to find a solution via Newton's root finding method as long as we start in the vicinity of the solution.

5.3.2 The Algorithm

To apply Newton's root finding method to MLE problems, we should replace f in (5.4)by the gradient and f' by the Hessian. For an estimation problem involving a single parameter, this implies the following algorithm, which is known as the **Newton-Raphson algorithm**:

$$\theta_{t+1} = \theta_t - \lambda_t \frac{l'(\theta_t)}{l''(\theta_t)}$$

(Compared to (5.4) I have added the step size parameter, λ, although this is frequently omitted.) This can be extended to multiple parameters to produce a general formulation of the Newton-Raphson algorithm:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \lambda_t \left(-\boldsymbol{\mathcal{H}}_t\right)^{-1} \boldsymbol{\nabla}_t \tag{5.5}$$

Since $-\boldsymbol{\mathcal{H}} = \boldsymbol{\mathcal{I}}_o$, the algorithm may also be written as $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \lambda_t \boldsymbol{\mathcal{I}}_{o_t}^{-1} \boldsymbol{\nabla}_t$.[3]

[3]This formulation should be clearly distinguished from the Fisher scoring algorithm that will be discussed in the next section. That algorithm uses the expected instead of the observed Fisher information.

The inclusion of the Hessian is the main distinguishing characteristic between the Newton-Raphson algorithm algorithm and the method of steepest ascent. It is also a major advantage of the algorithm. The Hessian gives the rate of change in the gradient. This allows for the adjustments to the estimates to be fine-tuned, so that smaller adjustments are made where the rate of change is relatively fast and larger adjustments are made where the rate of change is relatively slow. The method of steepest ascent does not allow any of this fine-tuning, which is the cause of the problems noted in Chapter 5.2.

One of the big advantages of the Newton-Raphson algorithm is that it is guaranteed to converge to a fixed point, given a sufficiently close starting value. Moreover, convergence is typically fast; near the solution, it is quadratic.[4] These are good reasons for using this algorithm.

On the downside, the Newton-Raphson algorithm has several drawbacks. First, the algorithm does not guarantee that the Hessian is negative definite, or equivalently that $-\boldsymbol{\mathcal{H}}$ is positive definite, causing potential problems with finding maxima.[5] The potential non-negative definiteness of the Hessian means also that there is no guarantee that the log-likelihood function will increase monotonically with each iteration.[6]

Second, convergence can be a problem. This happens when starting values have been chosen poorly, i.e. far away from the solution. It also happens

[4]The order of convergence is important in determining how quickly an algorithm reaches a solution. Let us define the approximation error as $\epsilon_t = |\theta_t - \tilde{\theta}|$, where $\tilde{\theta}$ is the solution that is being sought. Then the sequence $\theta_1, \theta_2 \cdots$ displays convergence of order β if $\lim_{t\to\infty} \epsilon_{t+1} = c\epsilon_t^{\beta}$. If $\beta = 1$, then we have linear convergence. If $\beta = 2$, then convergence is quadratic, which means that the number of correct digits doubles at every iteration.

[5]Remember that at a maximum, $\boldsymbol{\mathcal{H}}$ is negative definite. A constrained version of the Newton-Raphson algorithm can help to overcome the problem of non-negative definiteness. Under this algorithm, $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - (\boldsymbol{\mathcal{H}}_t + \tau_t \mathbf{I})^{-1}\boldsymbol{\nabla}_t$, where $\mathbf{I}$ is the identity matrix and τ_t is chosen so as to ensure that $\boldsymbol{\mathcal{H}}_t + \tau_t\mathbf{I}$ is negative definite.

[6]To see this, consider again the general form of hill-climbing algorithms. In such algorithms, $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \lambda \boldsymbol{D}\boldsymbol{\nabla}$. This may also be written as $\Delta\boldsymbol{\theta} = \lambda\boldsymbol{D}\boldsymbol{\nabla}$, where $\Delta\boldsymbol{\theta} = \boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}_t$. By definition, we also know that $\boldsymbol{\nabla'} = \Delta\ell/\Delta\boldsymbol{\theta}$ so that $\Delta\ell = \boldsymbol{\nabla'}\Delta\boldsymbol{\theta} = \ell_{t+1} - \ell_t$. Substituting the earlier result about $\Delta\boldsymbol{\theta}$ this yields

$$\ell_{t+1} - \ell_t \quad = \quad \lambda \boldsymbol{\nabla'} \boldsymbol{D} \boldsymbol{\nabla}$$

This is a quadratic form. For the log-likelihood function to increase monotonically, we require that $\ell_{t+1} - \ell_t > 0$. This will be the case only if $\boldsymbol{D}$ is positive definite. In the case of the Newton-Raphson algorithm, $\boldsymbol{D} = (-\boldsymbol{\mathcal{H}})^{-1}$. Unless the Hessian is negative definite, Δl is not necessarily positive.

when the log-likelihood function is relatively flat, so that $-\boldsymbol{\mathcal{H}} \approx \mathbf{0}$ and inversion becomes problematic. In this case, the direction vector cannot be computed and the log-likelihood function is said to be non-concave. Obviously, when this happens the estimates cannot be updated.[7]

Third, the algorithm requires knowledge of the second (partial) derivative(s). These derivatives can be quite complex, which may make it costly to compute them, although numerical differentiation methods can be of help here (see Chapter 5.7).

Finally, each iteration requires a large number of computations. If we estimate p parameters, then we need to compute p elements of the gradient and $.5p(p+1)$ elements of the Hessian, for a total of $.5p(p+3)$ computed elements. This is a considerable computational burden. Adding to this computational burden is the need to invert the Hessian. Especially if the number of parameters is large, this can be a complex computational task, although singular value decomposition of the Hessian can help to simplify it. Moreover, clever manipulation of the step size can also help to reduce the number of times that a direction vector has to be computed.[8] All of the aforementioned drawbacks make it necessary to sometimes consider alternatives to the Newton-Raphson algorithm.

5.4 The Method of Scoring

One alternative to the Newton-Raphson algorithm is the method of scoring, which is also known as the **Fisher scoring algorithm**.[9] This algorithm uses the inverse of the expected Fisher information as the direction matrix: $\boldsymbol{D} = \boldsymbol{\mathcal{I}}_e^{-1}$. Hence,

$$\begin{aligned} \boldsymbol{\theta}_{t+1} &= \boldsymbol{\theta}_t + \boldsymbol{\mathcal{I}}_{e_t}^{-1}\boldsymbol{\nabla}_t \\ &= \boldsymbol{\theta}_t + \mathbf{V}_t\boldsymbol{\nabla}_t \end{aligned} \tag{5.6}$$

[7] There are solutions for this problem. In Appendix II, I shall describe how Stata handles this.

[8] Indeed, manipulation of the step size is key to so-called fixed derivative Newtonian root finding methods. Here the direction vector is computed only once and parameters are updated by manipulating the step size (see e.g. Barnett [1966]).

[9] In the context of Generalized Linear Models, the Fisher scoring algorithm is known under the name *iteratively re-weighted least squares* (IRLS). For a more detailed discussion of IRLS see e.g. Hardin and Hilbe (2001).

where the second equation takes advantage of Definition 3.10, which states that the variance-covariance matrix is equal to the inverse of the expected Fisher information.[10] This may also be written as

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \left\{ -\sum_i \mathcal{E}[\boldsymbol{\mathcal{H}}_{it}] \right\}^{-1} \sum_i \boldsymbol{\nabla}_{it} \tag{5.7}$$

where I have taken advantage of equations (3.6) and (3.8). Given that the algorithm is based on the expected Fisher information matrix and can be stated in terms of scores, the name Fisher scoring algorithm is easily accounted for.

The chief advantage of the Fisher scoring algorithm over Newton-Raphson is that the expected information matrix is often simpler than the observed information matrix. The inverse of the expected information matrix is also guaranteed to be positive definite, since it is a proper variance-covariance matrix, which eliminates many of the convergence problems associated with the Newton-Raphson algorithm. The Fisher scoring algorithm is not quite as fast, however; instead of being quadratically convergent, it is linearly convergent. Moreover, the algorithm is practical only if the expectation of the Hessian is known, which may not be true.

5.5 Quasi-Newton Algorithms

Most of the problems of the Newton-Raphson algorithm stem from the need to compute the Hessian. To overcome these problems, a series of Newton-like algorithms have been developed that use surrogates for the Hessian. Specifically, these methods build up an approximate Hessian matrix by utilizing gradient information obtained in one or more iterations. Collectively, these methods are referred to as quasi-Newton (or variable metric) algorithms. They include algorithms by Berndt-Hall-Hall-Hausman (BHHH and BHHH-2), Davidson-Fletcher-Powell (DFP), and Broyden-Fletcher-Goldfarb-Shanno (BFGS). These are powerful numerical methods that form a useful alternative to the Newton-Raphson algorithm, especially for complex estimation problems.

[10] While, in principle, one could add a step size to the scoring algorithm, this is not usually done.

5.5.1 The BHHH and BHHH-2 Algorithms

The Berndt-Hall-Hall-Hausman or BHHH algorithm approximates the Fisher information matrix by taking the outer product of the gradient (OPG):

$$\boldsymbol{\mathcal{B}} = \sum_{i=1}^{n} \boldsymbol{\nabla}_i \boldsymbol{\nabla}'_i$$

where $\boldsymbol{\nabla}_i$ is sample unit i's score. Thus, the algorithm can be formulated as

$$\begin{aligned} \boldsymbol{\theta}_{t+1} &= \boldsymbol{\theta}_t + \boldsymbol{\mathcal{B}}_t^{-1} \boldsymbol{\nabla}_t \\ &= \boldsymbol{\theta}_t + \left(\sum_{i=1}^{n} \boldsymbol{\nabla}_{it} \boldsymbol{\nabla}'_{it} \right)^{-1} \sum_{i=1}^{n} \boldsymbol{\nabla}_{it} \end{aligned} \tag{5.8}$$

(It is not conventional to include a step size parameter in the algorithm.) Asymptotically, it can be shown that $\boldsymbol{\mathcal{B}} = -\boldsymbol{\mathcal{H}}$.

The chief computational advantage of the BHHH algorithm is that we do not need to compute the second derivatives. Instead, these derivatives are approximated by the outer product of the gradients, which reduces the number of computations considerably. As an added benefit, $\boldsymbol{\mathcal{B}}$ is guaranteed to be positive definite so that we can be sure that ℓ increases monotonically across iterations. A drawback of the BHHH algorithm is that the algorithm may be slow to converge if the starting values are far removed from the values that maximize the log-likelihood function.

The BHHH-2 algorithm is a slight modification of the BHHH algorithm. In the BHHH algorithm, $\boldsymbol{\mathcal{B}}$ captures information about the covariances between the scores, provided that the gradient is 0, as it should be at the maximum. Of course, if we haven't quite reached the maximum, the gradient is not zero, and $\boldsymbol{\mathcal{B}}$ does not capture covariances between scores. The BHHH-2 algorithm defines a matrix that can always be interpreted in terms of covariances between scores:

$$\boldsymbol{\mathcal{B}}^* = \sum_{i=1}^{n} (\boldsymbol{\nabla}_i - \boldsymbol{\nabla}) (\boldsymbol{\nabla}_i - \boldsymbol{\nabla})'$$

This matrix is then used in lieu of $\boldsymbol{\mathcal{B}}$ to update the estimates. The advantages of the BHHH-2 algorithm are similar to those of the BHHH algorithm.

5.5.2 The DFP and BFGS Algorithms

The DFP and BFGS algorithms compute an ever improving estimate of $(-\boldsymbol{\mathcal{H}})^{-1}$. The way this is done is that the estimate incorporates information from multiple points on the log-likelihood function. This stands in contrast to the other algorithms considered here, which calculate or approximate the Hessian only using the information contained in $\boldsymbol{\theta}_t$. Using only one piece of information becomes problematic if the log-likelihood function is not nearly quadratic.[11] In this case, the DFP and BFGS algorithms provide a better sense of the curvature of ℓ.

Both of these algorithms compute a so-called arc Hessian. This is the change in the slope from one point to the next. Imagine, for example, that the slope at $x = 3$ is 10 and at $x = 4$ the slope is 15, then the arc Hessian would be the difference in these slopes, namely 5. The two algorithms differ in the way they update the estimate of the Hessian, with BFGS adding a correction factor that DFP does not have.

The specific details of these algorithms are complex. They use as their approximation of $(-\boldsymbol{\mathcal{H}})^{-1}$ a matrix $\boldsymbol{A}$ such that

$$\Delta\boldsymbol{\theta}_{t+1} \quad = \quad \boldsymbol{A}_{t+1}\Delta\boldsymbol{\nabla}_{t+1}$$

where $\Delta\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}_t$, i.e. the change in estimates, and $\Delta\boldsymbol{\nabla}_{t+1} = \boldsymbol{\nabla}_{t+1} - \boldsymbol{\nabla}_t$, i.e. the change in slope or arc Hessian. DFP updates $\boldsymbol{A}$ using the following mechanism

$$\boldsymbol{A}_{t+1} \quad = \quad \boldsymbol{A}_t + \frac{\Delta\boldsymbol{\theta}_{t+1}\Delta\boldsymbol{\theta}'_{t+1}}{\Delta\boldsymbol{\theta}'_{t+1}\Delta\boldsymbol{\nabla}_{t+1}} - \frac{\boldsymbol{A}_t\Delta\boldsymbol{\nabla}_{t+1}\Delta\boldsymbol{\nabla}'_{t+1}\boldsymbol{A}_t}{\Delta\boldsymbol{\nabla}'_{t+1}\boldsymbol{A}_t\Delta\boldsymbol{\nabla}_{t+1}} \tag{5.9}$$

The updating mechanism for BFGS is

$$\begin{aligned}\boldsymbol{A}_{t+1} \quad = \quad & \boldsymbol{A}_t + \frac{\Delta\boldsymbol{\theta}_{t+1}\Delta\boldsymbol{\theta}'_{t+1}}{\Delta\boldsymbol{\theta}'_{t+1}\Delta\boldsymbol{\nabla}_{t+1}} - \frac{\boldsymbol{A}_t\Delta\boldsymbol{\nabla}_{t+1}\Delta\boldsymbol{\nabla}'_{t+1}\boldsymbol{A}_t}{\Delta\boldsymbol{\nabla}'_{t+1}\boldsymbol{A}_t\Delta\boldsymbol{\nabla}_{t+1}} + \\ & \Delta\boldsymbol{\nabla}'_{t+1}\boldsymbol{A}_t\boldsymbol{\nabla}_{t+1}\boldsymbol{u}\boldsymbol{u}'\end{aligned} \tag{5.10}$$

[11] Quadratic log-likelihood functions are ideal because the Newton-Raphson algorithm converges within one iteration for these functions. This is easily demonstrated for a single-parameter case. Let $\ell = a + b\theta + c\theta^2$. Then, $\ell' = b + 2c\theta$ and $\ell'' = 2c$. Given the formulation of the Newton-Raphson algorithm, $\theta_1 = \theta_0 - \ell'(\theta_0)/\ell''(\theta_0) = \theta_0 - (b/2c + \theta_0) = -b/2c$. This is also the solution to the likelihood equation $b + 2c\theta = 0$. Thus, in quadratic log-likelihood functions evaluating the Hessian at one point, namely the starting values, is fine. However, as the log-likelihood function starts to depart from quadrature, then it may become useful to resort to the DFP and BFGS algorithms.

where

$$\boldsymbol{u} = \frac{\Delta\boldsymbol{\theta}_{t+1}}{\Delta\boldsymbol{\theta'}_{t+1}\Delta\boldsymbol{\nabla}_{t+1}} - \frac{\boldsymbol{A}_t\Delta\boldsymbol{\nabla}_{t+1}}{\Delta\boldsymbol{\nabla'}_{t+1}\boldsymbol{A}_t\Delta\boldsymbol{\nabla}_{t+1}}$$

The staring value of $\boldsymbol{A}$ in both algorithms is an identity matrix, while the gradient is initially set to zero.[12]

The DFP and BFGS algorithms are typically fast in convergence—they display superlinear convergence, which means that the number of significant digits "liberated" by a given amount of computation increases as the algorithm nears convergence.[13] Moreover, these methods are designed to make the approximation of $(-\boldsymbol{\mathcal{H}})^{-1}$ positive definite. These features make them a popular choice in MLE.

5.6 Numerical Standard Errors

Just like the parameter estimates are derived through numerical methods for all but the simplest estimation problems, so are the standard errors. Dependent on the algorithm, the approach to computing the standard errors will take on a specific form. More specifically, there are four numerical procedures for computing the variance-covariance matrix of the estimators and, hence, for computing standard errors. Table 3.2 summarizes these four procedures.[14]

First, if parameter estimates are computed using the Fisher scoring algorithm, then it makes sense to base the standard errors on the inverse of the expected Fisher information matrix since the algorithm computes it anyway. This produces $\mathbf{V}_1$ as the numerical estimate of the variance-covariance matrix. Second, if one uses the Newton-Raphson algorithm for computing parameter estimates, then it makes sense to compute the standard errors

[12]A recent modification of the BFGS algorithm is the L-BFGS-B algorithm that was developed by Byrd et al. (1995). Here box constraints are added, so that a parameter may be given an upper and/or lower bound. This frees up memory and can facilitate convergence, especially in complex estimation problems. However, this algorithm appears to be less robust than BFGS.

[13]Superlinear convergence occurs when there is a sequence of constants, δ_t, that converges to zero and $\epsilon_{t+1} \leq \delta_t\epsilon_t$, where $\epsilon_{t+1} = |\theta_{t+1} - \tilde{\theta}|$ and $\epsilon_t = |\theta_t - \tilde{\theta}|$ are errors of approximation on successive iterations of the true solution $\tilde{\theta}$. Quadratic convergence is one form of superlinear convergence.

[14]The acronyms $\mathbf{V}_1$-$\mathbf{V}_3$ correspond to those used by Long (1997). $\mathbf{V}_4$ is not discussed by Long.

Table 5.2: Numerical Estimates of the VCE

Algorithm	V
Fisher Scoring	$\mathbf{V}_1 = \boldsymbol{\mathcal{I}}_e^{-1}$
Newton-Raphson	$\mathbf{V}_2 = \boldsymbol{\mathcal{I}}_o^{-1}$
BHHH	$\mathbf{V}_3 = (\sum_{i=1}^{n} \boldsymbol{\nabla}_{it} \boldsymbol{\nabla}'_{it})^{-1}$
DFP & BFGS	$\mathbf{V}_4 = \boldsymbol{\mathcal{A}}$

from the observed Fisher information matrix since this also is the direction matrix that the algorithm uses. This suggests $\mathbf{V}_2$ as the numerical estimate of the variance-covariance matrix. Third, the BHHH and BHHH-2 algorithms suggest using $\mathbf{V}_3$ as the numerical estimate of the variance covariance matrix, since the inverse of the outer-product of gradients is anyway required to compute the parameter estimates. Finally, in the DFP and BFGS algorithms, the variance-covariance matrix would be based on the approximation of $(-\boldsymbol{\mathcal{H}})^{-1}$, which is given by $\mathbf{V}_4 = \boldsymbol{\mathcal{A}}$, the same matrix that is used in computing the parameter estimates.

The four approaches all produce consistent estimates of the variance-covariance matrix of the estimators. Thus, they are asymptotically equivalent. In small samples, however, these procedures may yield slight differences in the computed standard errors.

5.7 Numerical Derivatives

In the discussion so far, it has been assumed that the gradient and Hessian can be obtained with relative ease. This is true for many of the applications that we shall encounter, but not for all. In a number of cases, the derivatives of the log-likelihood function may be so complex that they, too, have to be derived through numerical methods. Here, I shall describe one of the simplest procedures of numeric differentiation.

As you may recall, the mathematical definition of the first derivative of the function $\ell(\theta)$ with respect to θ is

$$\ell'(\theta) = \lim_{\Delta \to 0} \frac{\ell(\theta + \Delta) - \ell(\theta)}{\Delta}$$

In keeping with this definition, the numerical derivative may be evaluated as

$$\ell'(\theta) \approx \frac{\ell(\theta + \Delta) - \ell(\theta)}{\Delta} \tag{5.11}$$

where $\Delta < 1$ is a small numerical value. Thus (5.11) chooses a small increment for the parameter, evaluates the log-likelihood function at two points (an original value and the original plus the increment), and interprets the change as the gradient.[15] Numerical second derivatives can be computed in a similar manner. Specifically,

$$\begin{aligned} \ell''(\theta) &\approx \frac{\ell'(\theta + \Delta) - \ell'(\theta)}{\Delta} \\ &\approx \frac{\ell(\theta + 2\Delta) - 2\ell(\theta + \Delta) + \ell(\theta)}{\Delta^2} \end{aligned} \tag{5.12}$$

Most statistical programs are smart about choosing Δ and do it in such a way that approximation errors are minimized. Nevertheless, the use of numerical derivatives is second best to using analytical derivatives. As such, one should rely on numerical differentiation only in complex problems, where numerical derivatives may be the only alternative, or in very simple ones, where the derivatives are so unproblematic that numerical procedures will recover them without any difficulty.

5.8 Algorithmic Convergence

We have now seen how numerical optimization algorithms update estimates at successive iterations. But how do these algorithms determine when to stop updating? And is the updating process always successful? What problems could occur and how would one be able to tell if there was a problem?

[15] A variation that yields greater precision is the three-point method, which uses three points to derive the derivative. Another variant, used for example by Stata, computes centered derivatives:

$$\begin{aligned} \ell'(\theta) &\approx \frac{\ell(\theta + .5\Delta) - \ell(\theta - .5\Delta)}{\Delta} \\ \ell''(\theta) &\approx \frac{\ell(\theta + \Delta) - 2\ell(\theta) + \ell(\theta - \Delta)}{\Delta^2} \end{aligned}$$

While slower to compute, the three-point and centered derivative algorithms tend to be more precise.

5.8.1 Stopping Rules

An important question in numerical methods is when to stop the iterative process. This question may seem redundant because the theory of MLE already specifies such a criterion: $\boldsymbol{\nabla} = \mathbf{0}$. But numerical approximations of the gradient will never be exactly zero because this is technically impossible.[16] The question is thus how close we should be to zero to justify stopping the iterations. In practice, two stopping criteria are widely used.

One approach is to evaluate the following quantity at each iteration.

$$m_t \quad = \quad \boldsymbol{\nabla}'_t(-\boldsymbol{\mathcal{H}}_t)^{-1}\boldsymbol{\nabla}_t \tag{5.13}$$

We then compare m_t to an a priori determined **tolerance**, q, which should be small (e.g. .0001). If $m_t \geq q$ then the iterations continue. If $m_t < q$ then the iterations stop and the algorithm is said to have converged. The reason this criterion works is that m_t approaches 0 as we near the maximum of the log-likelihood function, since $\boldsymbol{\nabla} \to \mathbf{0}$ near a maximum. By setting a value of q close to 0, we could thus legitimately stop the algorithm. The advantage of this criterion is that m_t is the *test statistic* for the hypothesis that all of the elements of the gradient vector are zero; this statistic follows a chi-squared distribution with as many degrees of freedom as there are parameters.[17] A drawback of m_t is that it is tailored toward the Newton-Raphson algorithm.

A more general alternative, which is widely used in statistical software, is the **relative convergence criterion**, which is based on the relative change in the log-likelihood function between successive iterations. Convergence is declared if

$$\left| \frac{\ell_{t+1} - \ell_t}{\ell_t} \right| < q \tag{5.14}$$

where q is again an a priori defined tolerance. When the tolerance is set sufficiently small, improvements between successive iterations become trivial, affecting digits that we typically do not care about. Thus one rationale for the relative convergence criterion is pragmatic: small relative changes in the value

[16] A series of computer calculations will return a result of exactly 0 only through multiplication by 0 or through a Boolean operator. Neither of these situations apply to the computation of the gradient.

[17] The practical use of this result is limited because q is usually chosen to be much smaller than the critical value of the chi-squared distribution. Consequently, convergence is hardly ever declared on the basis of a test of the gradient.

of the log-likelihood function are simply not very important from a practical standpoint (e.g. they will never get reported in a table). The theoretical rationale is that small changes in the log-likelihood function should occur around the maximum, since this is where $\ell' \to 0$.

5.8.2 How Should Algorithms Converge?

How do we know that an algorithm has converged properly? Three diagnostics are particularly useful for identifying convergence problems: (1) the number of iterations, (2) concavity and backup warnings, and (3) the overall convergence path.

The number of iterations is an indicator of the ease with which the algorithm can find ML estimates. In general, we want fast convergence because this suggests that the log-likelihood function has a clear maximum. Of course, the speed of convergence also depends on the complexity of the model and properties of the data. Thus, quick convergence may not always be possible. On the other hand, if the algorithm takes hundreds of iterations, this is usually a warning sign no matter how complex the estimation problem.

Concavity warnings also serve as an indicator of potential convergence problems. A warning that the log-likelihood function is not concave means that $-\boldsymbol{\mathcal{H}}$ cannot be inverted and that the direction vector is undefined. If such a problem arises early on in the iterations, this is generally no cause for worry. All it means is that the log-likelihood function for the current estimate is (very-nearly) flat or that there is a saddle point or ridge (in the multi-parameter case). Most programs have a way around this problem and can improve subsequent iterations (see e.g. Chapter 5.9). However, a non-concavity problem in the last iteration is reason for worry. After all, it means that the last estimate is rather poor, and this is the one that is reported.

Similarly, backup warnings can signal convergence problem. These warnings arise in connection to the step size. If, for the current step size, the log-likelihood function does not increase from one iteration to the next, then the step size is reduced. Some software packages will then issue a warning that the algorithm has "backed up." If there are many of such backup warnings, this suggests that the log-likelihood function may be poorly behaved and that the algorithm has a difficult time finding a maximum.

Finally, the convergence path provides a useful diagnostic of algorithmic convergence. This shows the change of the log-likelihood function across iterations. We can graph this convergence path by placing the log-likelihood

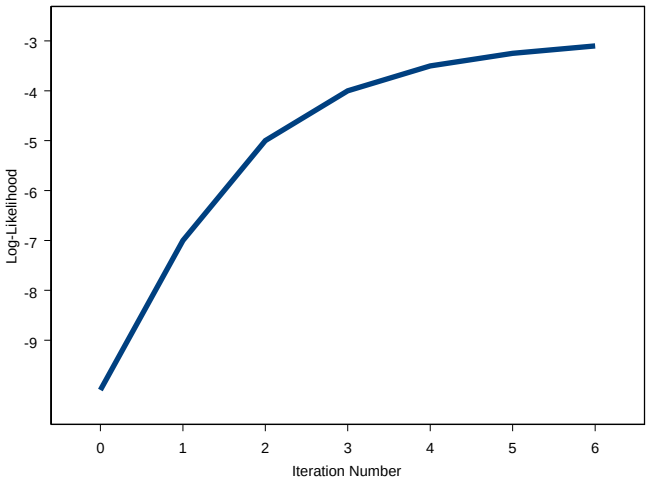

Figure 5.3: The Ideal Convergence Path

function on the vertical axis and the iteration number on the horizontal axis. Ideally, we would like to see a concave (i.e. marginally declining) function like the one that is depicted in Figure 5.3. Such a function suggests that initially great improvements in the log-likelihood occur as initial guesses give way to better guesses. However, when we approach the final estimate, the improvements become relatively small, as the algorithm is settling on its final estimate. Minor perturbations of this pattern should not worry us. But if the trajectory of the log-likelihood function is linear across iterations, this could indicate a problem—e.g., the premature ending of the algorithm. If the trajectory is marginally increasing, this could indicate a serious problem—the algorithm seems to be "hunting" for an estimate and is doing ever more poorly.

In sum, indications of proper convergence include relatively few iterations (as gauged against the complexity of the estimation problem), no concavity warnings at the final iteration, few or no backup warnings, and a marginally declining convergence path. When all of these things fall into place, once can be quite confident that the algorithm has converged properly and that one can trust the final estimates and standard errors.

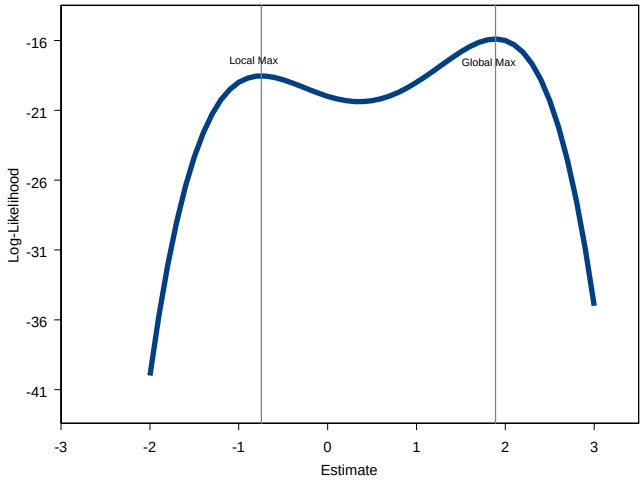

Figure 5.4: Local and Global Maxima

5.8.3 Local versus Global Maxima

All of the algorithms that we have discussed are susceptible to settling on a local instead of global maximum, unless we are in the fortunate situation that the log-likelihood function is single-peaked (e.g. quadratic), which is not always the case. Settling on a local maximum is a problem because this does not produce true ML estimates. This is illustrated in Figure 5.4 for the single-parameter case. This figure shows that local maxima can be a big problem. In this case, if the algorithm were to settle onto the local maximum, we would draw the inference that θ is negative. However, at the global maximum, the estimate of θ is actually positive, which would produce a completely different inference. For example, think of θ as a regression coefficient; at the local maximum we would say that the corresponding predictor has a negative effect when, in actuality, the effect is positive.

Figure 5.4 also indicates the importance of starting values. If we choose a starting value near the local maximum, the algorithm may well settle on this maximum instead of the global maximum. That is to say, the direction vector may never be large enough to move out of the vicinity of the local maximum. This is particularly likely, if the drop-off on either side of the local maximum is large (unlike Figure 5.3.). There are several ways around the problem

of local maxima. One approach is to perform the estimation several times, using different starting values. The value of the log-likelihood function should then be evaluated for the different starting values, to see if (roughly) identical solutions emerge. In a similar vein, it may also be useful to re-estimate a model using different algorithms to see if the estimates converge. Another approach is to use a different estimation technique to produce reasonable starting values, i.e. ones near the global maximum.[18] Finally, an intriguing new idea is the use of genetic algorithms (GAs) to generate starting values in a particular area of the parameter space. A GA generates a population of solutions, which are then refined through selection and mutation. The best solution from the GA can then serve as a reasonable starting value for a numerical algorithm. This approach is especially promising for complex models such as the multinomial probit model (see Liu and Mahmassani 2000).

5.8.4 Convergence Problems: Causes and Remedies

Convergence problems can arise due to any number of reasons. However, statisticians have identified a couple of causes that frequently produce convergence issues. It is useful to describe these causes in some detail so that you can be aware of them.

A first cause of convergence problems may be starting values. A poor choice of starting values can cause problems with convergence and, as we have just seen, local maxima. Most statistical packages will choose reasonable starting values for their built-in MLE routines (see the discussion above). When you program your own likelihood function, however, the choice of starting values may become a bigger problem. Chapter 5.9 discusses the selection of starting values in Stata and describes how one can ensure reasonable initial guesses of the parameters.

A small sample size is a second common cause of convergence problems. Convergence is generally faster the larger the sample size is relative to the number of estimated parameters. If the sample size is too small, the log-likelihood function may become relatively flat and this could cause non-concavity. Of course, using MLE in small samples is anyway precarious

[18] As a general note, it is quite common to construct starting values in this manner. For example, re-scaled OLS estimates are frequently utilized as starting values for logit and probit analysis. Even though the logit/probit log-likelihood functions are single-peaked (see Ch. 9), and hence not subject to the problem of local maxima, the usage of "smart" starting values can still help to speed up convergence.

because all of the desirable properties manifest themselves only asymptotically.

> **Example 5.1.** As we have seen in Chapters 2 and 3, the curvature is given by -1 times the Hessian, which is the observed Fisher information. For the exponential distribution this amounts to
>
> $$\mathcal{I}_o = -\frac{n}{\bar{y}^2}$$
>
> Imagine that two samples have been drawn from the same population. In both cases, $\bar{y} = 2$. However, the first sample has a size of $n = 10$ and the second sample a size of $n = 1000$. Curvature of the log-likelihood function in the first case is 2.5, whereas it is 250 in the second case. Thus, the rate of change in the gradient is much faster for the larger sample, allowing us to pin-point much more precisely where the maximum occurs. You can look through the other examples that we have discussed so far and determine that this is a general pattern—i.e. larger samples lead to more pronounced curvature and more precision in locating the maximum.

Third, complex likelihood functions can create convergence problems. Some likelihood functions are so complex that they challenge even the most powerful of computers, leading to slow convergence or frequent instances of non-convergence. This is particularly true of likelihood functions that involve multiple integrals such as those associated with the multivariate normal distribution. We shall encounter an instance of such a likelihood function when we discuss the multinomial probit model in Chapter 11. There, we shall also discuss the method of maximum *simulated* likelihood estimation as a way to overcome the considerable estimation difficulties that arise in the context of multivariate normal distributions.

Fourth, estimation at the boundary of parameter space can cause convergence problems. When the true estimate lies at the boundary of parameter space, then algorithms may be pushed into considering inadmissible estimates, which may in turn make it impossible to compute the log-likelihood function and its derivatives. This problem arises frequently when we are estimating probabilities, which are bounded between 0 and 1, or correlations, which are bounded between -1 and 1. If the true probability is close to 0 or 1, or the true correlation is close to -1 or 1, then the algorithm may end up exploring estimates on the wrong side of these bounds. This may cause the algorithm to halt completely or to produce nonsensical results such as

negative variance estimates. The solution in these cases is to find an appropriate transformation of the parameter that is itself not bounded. One can then rely on the invariance principle (Definition 4.1), which ensures that the transformation is also a ML estimate. A common transformation of probabilities is the logit transformation, which we already encountered in Example 4.1. A common transformation of correlations is

$$\tanh^{-1}\rho \;=\; .5\ln\left(\frac{1+\rho}{1-\rho}\right)$$

where $\tanh^{-1}$ is the inverse of the hyperbolic tangent (sometimes written as atanh), which is also known as Fisher's z-transform.

Fifth, the scaling of variables may cause convergence problems. The larger the ratio between the largest and smallest standard deviations, the more likely convergence problems are. Long (1997) suggests that this ratio should not exceed 10. The solution in this case is as simple as changing the units on a variable (e.g. income in \$1000 units instead of \$1 units).

Sixth, convergence problems can arise when the model is inappropriate for the data. This can happen when the wrong PDF is specified or, more generally, when the model makes assumptions that do not match the data generation process. In this case, the algorithm will try hard to find the best estimates, but this may simply not be possible. This is one reason why considerable thought should be placed in model specification, i.e. selection of predictors as well as distributional assumptions, as was emphasized in Chapter 2.5.

Finally, data cleaning problems may be a cause of convergence difficulties. If the data contain errors (e.g. negative values where positive ones are expected) then convergence can become a problem. As always, you should inspect your data through descriptive statistics and graphical displays before touching complex statistical techniques like MLE. This will also help you to assess whether the distributional assumptions that you make seem reasonable for the data at hand.

5.9 Numerical Optimization in Stata

Stata provides an efficient, user-friendly platform for numerical optimization of log-likelihood functions. It includes the following helpful features: (1) fast

convergence (under most circumstances) using the Newton-Raphson or quasi-Newton algorithms; (2) a conservative approach to declaring convergence, which leads to more trustworthy estimates; (3) simplifying features that allow for implementation of MLE with a minimum of calculus; (4) robust variance estimation; (5) Wald and likelihood ratio test procedures (see Chapter 6); (6) a search routine that chooses improved starting values; (7) estimation under linear constraints; and (8) extensive post-estimation commands. In this section, I provide a basic overview of Stata's numerical optimization routines.[19] It is assumed that you have access to version 8 or 9 of Stata.

5.9.1 Overview

Numerical optimization in Stata requires a series of steps. These steps are indicated below in the order in which they would normally be entered (optional steps are marked by an asterisk).

1. `program`: This command allows the user to write a program containing the parameters and log-likelihood function. All of this is done in general terms, so that the commands can be used in any application where they are relevant. (The program may be kept in a separate `do` file.)

2. `ml model`: This command specifies the model that is to be estimated (i.e., response variable and covariates), as well as the program that should be run and the way in which it should be run. This command is application-specific: it specifies the model in terms of a particular set of variables that is loaded into memory.

3. `ml check`*: This command checks the program syntax for mistakes. While optional, it is extremely useful for debugging MLE routines. Beginning programmers are advised to use this command.

4. `ml search`*: This optional command causes Stata to search for better starting values for the numerical optimization algorithm.

[19] I will limit myself to a discussion of the `lf` method. For a discussion of other programming methods, the reader is referred to Gould, Pitblado, and Sribney (2006). Note that while the `lf` method is the easiest way to program ML estimators, it is not necessarily the most accurate. In my programming experience, however, I have found that it works well for a large variety of estimation problems.

5. `ml plot`*: Like `ml search`, this command helps with finding starting values. It is also a useful utility for plotting log-likelihood functions.

6. `ml init`*: This is another command that let's you manipulate the starting values.

7. `ml maximize`: This command starts the execution of the estimation commands and generates the output.[20]

8. `ml graph`*: This is an optional command that produces a graph showing the iteration path of the numerical optimization algorithm (cf. Figure 5.3). I recommend using this command so that one can monitor convergence.

5.9.2 Detailed Syntax

Program

In most cases, writing and MLE program requires only a couple of lines of syntax. At least, this is the case if the `lf` method is used. This method is appropriate if the log-likelihood function meets the linear form (lf) restriction—i.e. the observations are independent. Under the `lf` method, all derivatives are computed numerically and the log-likelihood function can be stated in terms of a single sample unit.

A program is started by entering

`program` [`define`] *program_name*

where *program_name* is any name up to 32 characters in length. (It is preferable to choose a descriptive name.) To end the program, one should type

```
end
```

In between these keywords, the user declares the parameters and the log-likelihood function.[21] First, the log-likelihood function and its parameters

[20] This command is optional if the estimation is performed non-interactively, e.g. as part of an `do` file. In that case, the `maximize` command is added as an option to the `ml model command`.

[21] It may also be useful to declare several temporary variables as is illustrated in Example 5.4.

have to be labeled. This is done through the command `args` (which is an abbreviation for the computer term "arguments"). Next, the log-likelihood function has to be defined; this is done using the `quietly replace` command.[22] In addition to these specifications, it is often useful to declare the program version, especially if you are planning to make changes to the program over time.

Example 5.2. To illustrate `program define` let us consider the Poisson log-likelihood function from Example 2.4:

$$\ell \;=\; \sum_i y_i \ln(\mu) - n\mu - \sum_i \ln(y_i!)$$

To program this function, we could use the following syntax:

```
program define poisson
  version 1.0
  args lnf mu
  quietly replace `lnf' = $ML_y1*ln(`mu') - `mu' ///
  - lnfact($ML_y1)
  end
```

Let us analyze what this program does. In the first line we define the program, calling it `poisson`. In the second line, we show the version of the program (`version 1.0`). The third line provides a name for the log-likelihood function (`lnf`) and its one parameter (`mu`). The fourth and fifth line specify the log-likelihood function and the sixth line ends the program.[23]

The action, of course, is in the fourth and fifth line.[24] This line is based on the arguments specified in `args`. Because we are referring to arguments, they should be placed in apostrophes.[25] The fourth

[22] "Replace" indicates that the user is substituting a new expression. "Quietly" implies that Stata does not echo this substitution—i.e., it is not displayed on the screen or in the output.

[23] The `///` character at the end of the fourth line is Stata's command continuation character. This allows the user to continue a command onto the next line.

[24] The fifth line is not actually all that interesting because it does not contain any terms involving the parameter μ. If we had programmed just the kernel of the log-likelihood function, then we could have omitted the fifth line altogether.

[25] In fact, the leading apostrophe is a back-apostropher and is typically located on the same key as the tilde; the second apostrophe is straight and is typically located on the same key as the double apostrophe.

line also contains the variable `$ML_y1`, which is the internal label for the (first) response variable. Stata will replace this with an appropriate variable from the data set after the `ml model` command has been specified.By not referring to a specific variable name here, the program can be used for any data set. Finally, the fourth line specifies a function. (The last term in this expression, `lnfact($ML_y1)`, stands for $\ln(y!)$.)

A careful inspection of the fourth line of code shows that it looks a lot like the log-likelihood function, except that it does not include summations. In fact, this line gives the log-likelihood function for a single observation:

$$\ell \quad = \quad y_i \ln(\mu) - \mu - \ln(y_i!)$$

As long as the observations are independent (i.e., the linear form restriction on the log-likelihood function is met), this is all you have to specify. Stata knows that it should evaluate this function for each observation in the data and then sum the results. This greatly simplifies programming log-likelihood functions.

Example 5.3. Consider again the normal PDF that we discussed in Example 3.1. A slightly different way of defining this PDF is:

$$\begin{aligned} f(y) &= \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{1}{2}\left(\frac{y-\mu}{\sigma}\right)^2\right\} \\ &= \frac{1}{\sigma}\phi(z) \end{aligned}$$

where $z = (y - \mu)/\sigma$ and $\phi(.)$ denotes the standard normal distribution.[26] If we draw a sample of n independent observations from this

[26]To derive the second line of this equation, we proceed as follows. First, we substitute z in the formula for the normal distribution:

$$f(y) \quad = \quad \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-.5z^2\right\}$$

Next, we compare this result to the standard normal distribution:

$$\phi(z) \quad = \quad \frac{1}{\sqrt{2\pi}} \exp\left\{-.5z^2\right\}$$

We see that the generic normal distribution is almost identical to the standard normal

PDF, then the log-likelihood function is

$$\ell \;=\; -n\ln(\sigma) + \sum_i \ln[\phi(z_i)]$$

We can program this function using the following syntax:

```
program define normal
  version 1.0
  args lnf mu sigma
  quietly replace `lnf' = ln(normd(($ML_y1-`mu')/`sigma')) ///
  - ln(`sigma')
  end
```

Here `normd` is $\phi(.)$ and `($ML_y1-`mu')/`sigma'` is z_i. Again, we only have to specify the log-likelihood function for a single observation. Stata will evaluate this function for all observations and accumulate the results to obtain the overall log-likelihood.

Example 5.4. The following example shows how temporary variables may help to simplify the task of programming a complex log-likelihood function. Consider the bivariate normal distribution:

$$f(y_1, y_2) \;=\; \frac{1}{2\pi\sigma_1\sigma_2\sqrt{1-\rho^2}} \exp\left\{-\frac{z_1^2 + z_2^2 - 2\rho z_1 z_2}{2(1-\rho^2)}\right\}$$

where $z_1 = (y_1 - \mu_1)/\sigma_1$, $z_2 = (y_2 - \mu_2)/\sigma_2$, μ_1 and μ_2 are the means of y_1 and y_2, respectively, σ_1 and σ_2 are the standard deviations of X_1 and X_2, respectively, and $\rho = \sigma_{12}/\sigma_1\sigma_2$ is the Pearson product-moment correlation coefficient between the variables, which is bounded between -1 and 1. Drawing a sample of n independent observations from the bivariate normal distribution yields the following

---

distribution; the only difference is a term in σ^2. Factoring this term out, we get

$$\begin{aligned} f(y) &= \frac{1}{\sqrt{\sigma^2}}\frac{1}{\sqrt{2\pi}} \exp\left\{-.5z^2\right\} \\ &= \frac{1}{\sqrt{\sigma^2}}\phi(z) \\ &= \frac{1}{\sigma}\phi(z) \end{aligned}$$

log-likelihood function kernel:

$$\begin{aligned}\ell &= -.5n \ln \left[\sigma_1^2 \sigma_2^2 (1-\rho^2)\right] + \frac{1}{2(1-\rho^2)} \sum_{i=1}^{n} \left\{ z_{i1}^2 + z_{i2}^2 + 2\rho z_{i1} z_{i2} \right\} \\ &= .5n \ln \sigma_1^2 + .5n \ln \sigma_2^2 + .5n \ln(1-\rho^2) + \\ &\quad \frac{1}{2(1-\rho^2)} \sum_{i=1}^{n} \left\{ z_{i1}^2 + z_{i2}^2 + 2\rho z_{i1} z_{i2} \right\}\end{aligned}$$

We can program this function using the following syntax:[27]

```
program define bivnorm
  version 1.0
  args lnf theta1 theta2 theta3 theta4 theta5
  tempvar v1 v2 r z1 z2 t
  quietly {
  gen double ‘v1’=exp(‘theta3’)
  gen double ‘v2’=exp(‘theta4’)
  gen double ‘r’=(exp(‘theta5’)-1)/(exp(‘theta5’)+1)
  gen double ‘z1’=($ML_y1-‘theta1’)/sqrt(‘v1’)
  gen double ‘z2’=($ML_y2-‘theta2’)/sqrt(‘v2’)
  gen double ‘t’=1-(‘r’2)
  replace ‘lnf’=-.5*log(‘v1’*‘v2’*‘t’)- ///
  (1/(2*‘t’))*(‘z1’2+‘z2’2-2*‘r’*‘z1’*‘z2’)
  }
  end
```

In this program, we declare five parameters: θ_1-θ_5, which are labeled `theta1`-`theta5`. Of these parameters, only two are equivalent to the parameters of the bivariate normal distribution, namely $\theta_1 = \mu_1$ and $\theta_2 = \mu_2$. The remaining estimated parameters are linked to the bivariate normal distribution through more complex mathematical functions. For instance, σ_1^2, which is declared as the temporary variable `v1`, is equal to $\exp \theta_3$ or, equivalently $\theta_3 = \ln \sigma_1^2$.[28] It is the equivalent

[27] This syntax closely follows Eliason (1993: 56-57).

[28] All of the temporary variables are stored numerically in double precision, which is the most precise format that Stata has. This storage format is indicated by declaring `double` in the `gen` command. When declaring temporary variables you should always use double precision.

form that can be readily recognized in the log-likelihood function. A similar function relates σ_2^2 to θ_4. Finally, the correlation coefficient, which is declared as the temporary variable `r`, is related to `theta5` by way of $\rho = [exp(\theta_5) - 1]/[\exp(\theta_5) + 1]$. This is the kind of transformation discussed earlier that allows one to avoid optimization at the boundaries of parameter space. While $-1 \leq \rho \leq 1$, $-\infty < \theta_5 < \infty$. Thus, we are obtaining an ML estimate of a parameter that is not bounded; this estimate can then be transformed into an ML estimate of ρ, the parameter that is of real interest.[29]

Three more temporary variables are declared in the program. Of these, `z1` and `z2` correspond to z_1 and z_2 in the log-likelihood function. Finally, `t` contains $1-\rho^2$. The log-likelihood function is declared entirely in terms of the temporary variables. This allows for a more compact notation of this function and it may also cut down on mistakes.

As the examples above show, programming log-likelihood functions is relatively straightforward in Stata. I should stress that this simplicity arises because of the `lf` method. This method frees us from having to use explicit summation commands and from specifying the gradient and the Hessian. It is not always possible to use the `lf` method and in that case, Stata MLE programs will generally be much more complex (for examples see Gould, Pitblado, and Sribney 2006).

ML Model

To apply a program to a particular data set, you need to issue the `ml model` command. The abridged syntax for the `lf` method is

```
ml model lf prog_name eq [ eq ... ] [ weight ] [ if ] [ in ] [ , robust
cluster(varname) constraints(numlist|matname)
technique(nr|bhhh|dfp|bfgs) vce(oim|opg|native|robust) ]
```

Here *prog_name* is the name of the program that contains the log-likelihood function and *eq* specifies one or more equations that indicate the model that is being estimated.

[29] It is easily ascertained that $\rho \to -1$ when $\theta_5 \to -\infty$ and that $\rho \to 1$ when $\theta_5 \to \infty$. Further, $\rho = 0$ when $\theta_5 = 0$.

Equations Equations are a critical aspect of the `ml model` command. The reason is that Stata needs to know the model that you want to estimate, including the response variable and, if relevant, the covariates. These variables are declared by specifying one ore more equations. The user can specify these equations before running `ml model` by using an alias. It is also possible to specify the equations in the `ml model` command, placing each equation in parentheses.

The general rule is that a separate equation is specified for each parameter that appears on the `args` line of the program. The equation can be empty if there are no covariates. It can be more elaborate if there are covariates. At least one of the equations needs to be specify the response variable; if there are multiple response variables, then these will have to be declared through multiple equations. One of the very nice features of this setup is that the same program can be used to serve multiple roles, as the following two examples illustrate.

> **Example 5.5.** Imagine that we seek to estimate the parameters of the normal distribution using the program described in Example 5.3. Our data set contains the variable `y`, which we assume to be a realization of the normal PDF. To estimate the parameters of this distribution we start by loading the `normal` program into memory (by typing it in or by executing the `do` file containing this program). Then we can type
>
> ```
> ml model lf normal (y=) ()
> ```
>
> (This is the most minimalist version of the `ml model` command.) The first equation—`(y=)`—pertains to the mean of the normal PDF, since this is the first parameter in the `args` line of the `normal` program, while the second equation—`()`—pertains to the standard deviation. At least one of the equations has to specify the response variable. Here I have done that in the equation for μ by specifying `(y=)`. Otherwise, the equations are empty because there are no covariates to consider.

> **Example 5.6.** Now imagine that our data set contains two covariates, `x` and `z`. We would like to run the classical normal linear regression model (CNLRM—see Chapter 3.3) to determine how these covariates are related to the response variable. How would we do this? Quite simply, we would amend the `ml model` command from Example 5.5 in the following way:

```
ml model lf normal (y=x z) ()
```

By adding the covariates to the first equation, this will now refer to the conditional mean of `y`. The second equation now refers to the root MSE, i.e. the square root of the residual variance.

What happens behind the scenes in these two examples is that the `ml model` command substitutes the variable `y` for the generic placeholder `$ML_y1` that is specified in the `normal` program. This is one of the beauties of Stata because it means that the same program can be applied to any data set. Another nice feature of Stata is that no placeholders for covariates need to be specified in the program. This allows the same program to be used for estimation problems with and without covariate information (e.g. estimating the parameters of a normal PDF or a CNLRM).

Aliasing Rather than specifying the model equations in the `ml model` command, they may be specified ahead of time. Each equation receives a name or alias that will then be referred to in the `ml model` command. These names will also appear in the output, which can help with the interpretation.

Example 5.7. Consider again the problem of estimating the parameters of a normal PDF. In Example 5.5, the equations were specified as part of the `ml model` command. However, we could also have used aliasing by specifying the equations for μ and σ ahead of time and referencing their names in the `ml model` command. The following syntax shows how this can be done.

```
eq mean:  y=
eq sigma:
ml model lf normal mean sigma
```

Model Options The `ml model` command provides a large number of options. The more important of these options are the following.

1. The option `robust` causes Stata to compute robust standard errors using the sandwich estimator of the variance-covariance matrix of the estimators. Specifically, Stata computes the robust variance estimator as

$$\mathbf{V}[\hat{\boldsymbol{\theta}}] \;=\; \{-\boldsymbol{\mathcal{H}}\}^{-1}\left\{\frac{n}{n-1}\sum_{i=1}^{n}\boldsymbol{\nabla}_i\boldsymbol{\nabla}'_i\right\}\{-\boldsymbol{\mathcal{H}}\}^{-1}$$

The `robust` option comes in useful if one is concerned that the postulated density is not the true density. The robust standard errors do not depend on the assumed density being the true density, whereas the normal estimators of $\mathbf{V}$ do depend on this.[30] To highlight this difference, Stata reports the log-likelihood function as the *log pseudo-likelihood*.

2. The `cluster` option allows the standard errors to be corrected for any un-modeled dependencies among the observations, for example dependencies arising from shared contexts.[31]

3. Stata allows for constrained MLE. The constraints are specified using the `constraints` option, which will be discussed in greater detail in Chapter 5.9.3.

4. By default, Stata uses the Newton-Raphson algorithm (`nr` in Stata speak) for purposes of optimizing the log-likelihood. However, it is also possible to use one of the quasi-Newton methods that we have discussed. The `technique` option allows the user to declare what algorithm should be used. For example, `technique(bfgs)` causes Stata to rely on the BFGS algorithm.

5. By default, Stata bases the standard errors on the observed Fisher information matrix (`oim` in Stata jargon), unless one is using the BHHH algorithm. Using the `vce` option, one can alter the default. Other than `oim`, the available options are: (1) outer-product of gradients (`opg`), (2) `robust`, and (3) `native`. The `opg` method is the default for the BHHH algorithm. The `robust` option provides just another way of asking for the sandwich variance estimator. Finally, `native` causes Stata to base the standard errors on whatever information matrix is estimated by the algorithm.

[30] However, if the true density departs a great deal from the assumed density, then parameter estimates may still be inconsistent. Thus, the `robust` option is no panacea.

[31] Many dependencies are artifacts of the sampling design. For instance, cluster sampling designs tend to produce dependencies among subsets of sample units. Stata has an extensive array of survey options that can handle these kinds of design effects. A discussion of these options is beyond the scope of this report but a detailed discussion can be found in Gould, Pitblado, and Sribney (2006).

ML Maximize

In interactive mode, `ml model` does not produce any output because the command does not actually start execution of the algorithm. To start the numerical optimization process, the user will have to specify one more command: `ml maximize`. An abridged version of the command is as follows.

```
ml maximize [ , difficult level(#) nolog iterate(#)
ltolerance(#) ]
```

Once the `ml maximize` command is issued, the program will execute the algorithm and report the results, assuming there are no errors in the program.

In my experience, the `difficult` option has been quite useful, especially in conjunction with the Newton-Raphson algorithm. This option asks Stata to temporarily switch algorithms should a concavity issue arise. Stata computes the eigenvalues of $\{-\boldsymbol{\mathcal{H}}\}^{-1}$ and switches temporarily to the method of steepest ascent for the orthogonal subspace where the eigenvalues are negative or small positive numbers. Oftentimes, this can help to overcome convergence problems.

Another useful option is `level`. This sets the level for the confidence interval that `ml maximize` computes. The default level is 95, which results in a 95% confidence interval. If you want a 90% confidence interval instead you would specify `level(90)`, while you would declare `level(99)` for a 99% confidence interval, and so forth.

The other options shown above are less helpful and, in general, I would recommend against tinkering with them. First, `nolog` causes Stata to suppress the iteration log, showing only the log-likelihood function at the last iteration (and any efforts at finding feasible starting values). This reduces the amount of output on the screen, but it also suppresses useful information such as the number of iterations and backup or concavity warnings. Second, `iterate` lets you change the maximum number of iterations that Stata will work through before it stops. Since the default is quite high (16000 iterations), there is usually no reason to play with this option. Finally, `ltolerance` allows you to change the tolerance value that is associated with (5.14). The default tolerance is 1e-7. By increasing this number, it may be possible to speed up convergence. However, it is risky to relax the tolerance too much because the final estimates may be far removed from those that truly optimize the log-likelihood function.

The results that Stata will report after running the `ml maximize` command include the iteration history (unless this was suppressed via the `nolog` option), the value of the log-likelihood function at the last iteration, the sample size, the parameter estimates (reported under the rubric of "coefficients"), the estimated standard errors, z-test statistics and their p-values (see Chapter 6), the Wald confidence interval (see Chapter 7), and, where appropriate, the Wald test. Example 5.8 illustrates these features for the normal PDF.[32]

> **Example 5.8.** Consider that we have $n = 1000$ draws from the standard normal PDF, which are stored in the variable `y`. We run the program `normal` that we created in Example 5.3. Having run the `maximize` command, Stata produces the output shown in Figure 5.5. The column with the heading `coef.` shows the parameter estimates. For the first equation, this is the estimate of μ; for the second equation it is the estimate of σ.

Other Useful Commands

The `program`, `ml model`, and (in interactive settings) `ml maximize` commands are the minimum required commands to perform MLE in Stata. However, several other commands, while optional, can be quite useful. This is particularly true for complex estimation problems or when you are novice at programming your own log-likelihood functions in Stata.

In many cases, we may wish to debug the programs that we have written to perform MLE. This can be done via the

```
ml check
```

command. This command is run after the `ml model` command and performs a series of tests to see if there are problems with the program and/or its specific application. These tests include checks on whether the log-likelihood function can be computed and updated.

The `ml search`, `ml plot`, and `ml init` commands are used to influence how Stata selects its starting values. Before describing these optional commands, it is useful to dwell a little on Stata's default procedure for selecting

[32]Since no covariates are included in this example, the Wald test is inappropriate and receives a missing value.

```
. use "C:\data\Example 5.8.dta", clear
. do "C:\data\normal.do"
. program define normal
  1. version 1.0
  2. args lnf mu sigma
  3. quietly replace `lnf'=ln(normd(($ML_y1-`mu')/`sigma'))-ln(`sigma')
  4. end
.
end of do-file
. eq mean: y=
. eq sigma:
. ml model lf normal mean sigma
. ml max
initial:       log likelihood =     -<inf>  (could not be evaluated)
feasible:      log likelihood = -2677.8307
rescale:       log likelihood = -1901.6308
rescale eq:    log likelihood = -1412.2134
Iteration 0:   log likelihood = -1412.2134
Iteration 1:   log likelihood = -1412.1638
Iteration 2:   log likelihood = -1412.1637
                                                Number of obs   =       1000
                                                Wald chi2(0)    =          .
Log likelihood = -1412.1637                     Prob > chi2     =          .
```

y	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
mean						
_cons	.0106352	.0314093	0.34	0.735	-.0509258	.0721963
sigma						
_cons	.9932481	.0222097	44.72	0.000	.9497179	1.036778

Figure 5.5: MLE of the Normal PDF in Stata

starting values. Ordinarily, Stata sets all of the starting values to zero: $\boldsymbol{\theta}_0 = \mathbf{0}$. If the log-likelihood function cannot be evaluated with these starting values, then Stata uses a pseudo-random number generator to produce a new set of starting values (and will continue to do so until a valid set of starting values has been found). The example in Figure 5.5 illustrates the process. On the initial evaluation, Stata produces a value of negative infinity (`-<inf>`) followed by the message `could not be evaluated`. This happens because the program sets $\sigma = 0$ as its initial guess. One of the terms of the log-likelihood function is $\ln \sigma$, which is not defined for this starting value. The subsequent lines (`feasible`, `rescale`, and `rescale eq`) use a new set of starting values generated via a pseudo-random number generator; those starting values are valid and the algorithm converges rapidly.[33]

In many instances, relying on Stata's "brute force" method for choosing starting values will work just fine. But you can also provide your own starting values to speed up the estimation process. One way to do this is to use the `ml search` command. The abridged syntax for this command is

`ml search` [[`/`]*eqname*[`[`]`:`] *#lb #ub*][`...`] [`, repeat(`*#*`)`]

Here *# lb* is a lower-bound and *# ub* is an upper-bound. The command searches for starting values at the equation level, not at the level of particular parameters. For example, when running a regression analysis one cannot specify upper and lower-bounds for particular regression coefficients. Rather, one would generate one set of lower and upper bounds. Since the search procedure is highly efficient, it generally does not matter much how one sets these bounds. The procedure sets the constant in each equation to a randomly chosen number, while initially setting all other coefficients to zero.[34] By default, it repeats the pseudo-random number generation 10 times, but this can be increased or decreased via the `repeat` option.

A second command for selecting starting values is graphically oriented. The `ml plot` command obeys by the following syntax:

`ml plot` [`eqname:`] *name*

[33] Stata re-scales the starting values by multiplying them with a constant k such that $\ell(k\boldsymbol{\theta}_0) > \ell(\boldsymbol{\theta}_0)$. This is what is indicated by `rescale`. It repeats the re-scaling for each equation and this is indicated by `rescale eq`.

[34] All equations contain at least a constant, which is marked as `_cons`, as Figure 5.5 shows.

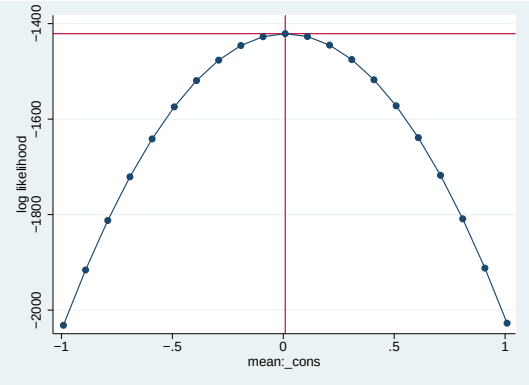

Figure 5.6: Stata Plot of a Log-Likelihood Function

This causes Stata to draw a picture of the log-likelihood function for a particular parameter. The focus on specific parameters sets this command apart from the `ml search` command. A by-product of the picture is that `ml plot` uses the value at which the maximum occurs as the starting value of that parameter. Note that `ml plot` can be seeded with the starting values generated by `ml search`, as is done in Example 5.9.[35]

> **Example 5.9.** For the normal PDF, we can plot the log-likelihood function of μ using the following syntax.
>
> ```
> ml model lf normal mean sigma
> ml search mean -2 2
> ml plot mean:_cons
> ```
>
> This creates the plot shown in Figure 5.6.

A third command for declaring starting values is `ml init`. This command comes in several guises, including:

`ml init` { [`eqname:`]*name*=#| */eqname*=#}
`ml init` *matname*

For example, `ml init mean:_cons=2` sets the constant in the mean model to an initial value of 2. We can also include an entire vector of starting values

[35] Stata recommends this command sequence since it ensures that `ml plot` starts with reasonable values (Gould, Pitblado, and Sribney 2006).

using the second version of the command. This makes it possible to run a different estimation procedure first and to use the saved estimates of that procedure as the starting values for ML estimates.

One final command is useful, namely the `ml graph` command. This command causes Stata to plot the convergence path, producing a graphic like the one shown in Figure 5.3. Along with other information contained in the output, this is a useful diagnostic for evaluating if the algorithm has converged properly.

5.9.3 Constrained Optimization

In certain instances, it is useful to place constraints on the parameters and to perform the estimation subject to those constraints. Stata handles linear constraints quite easily within the `ml model` command. The key is to begin by specifying the appropriate constraint via `constraint`. One version of the syntax for this command is:

`constraint` [`define`] # [*exp*=*exp*| *coefflist*]

Here # is the constraint number, which is then referenced in the `ml model` command. Example 5.9 illustrates how the process works.

Example 5.10. The 2-parameter Weibull PDF is given by

$$f(y) = \frac{\eta}{\beta}\left(\frac{y}{\beta}\right)^{\eta-1}\exp\left\{-\left(\frac{y}{\beta}\right)^{\eta}\right\}$$

where $\beta > 0$ is the scale parameter and $\eta > 0$ is the shape parameter. If $\eta = 1$, then this distribution reduces to the exponential PDF discussed in Example 2.5. Thus, we can estimate the exponential PDF as a constrained version of the 2-parameter Weibull distribution.

To do so, we begin by deriving the Weibull likelihood and log-likelihood functions. Writing the Weibull PDF slightly differently we have

$$f(y) = \eta\beta^{-\eta}y^{\eta-1}\exp\left\{-\left(\frac{y}{\beta}\right)^{\eta}\right\}$$

(You should verify this result.) Drawing a sample of n independent observations from the Weibull distribution we obtain the following

likelihood function

$$\begin{aligned}
\mathcal{L} &= \eta\beta^{-\eta}y_1^{\eta-1}\exp\left\{-\left(\frac{y_1}{\beta}\right)^{\eta}\right\}\times \\
&\quad \eta\beta^{-\eta}y_2^{\eta-1}\exp\left\{-\left(\frac{y_2}{\beta}\right)^{\eta}\right\}\times\cdots\times \\
&\quad \eta\beta^{-\eta}y_n^{\eta-1}\exp\left\{-\left(\frac{y_n}{\beta}\right)^{\eta}\right\}\times \\
&= \eta^n\beta^{-n\eta}\prod_{i=1}^{n}y_i^{\eta-1}\exp\left\{-\sum_{i=1}^{n}\left(\frac{y_i}{\beta}\right)^{\eta}\right\}
\end{aligned}$$

The corresponding log-likelihood function is

$$\ell = n\ln\eta - n\eta\ln\beta + (\eta-1)\sum_{i=1}^{n}\ln y_i - \sum_{i=1}^{n}\left(\frac{y_i}{\beta}\right)^{\eta}$$

This function can be programmed using the following syntax:

```
program define weibull2
  version 1.0
  args lnf beta eta
  quietly replace `lnf' = ln(`eta') - `eta'*ln(`beta') ///
  + (`eta'-1)*ln($ML_y1) - (($ML_y1/`beta')^`eta')
  end
```

Imagine that we have drawn $n = 1000$ observations from a Weibull distribution with a shape parameter of 1.2 and a scale parameter of .25. First, we estimate an unconstrained model. The syntax and results for this model are shown in Figure 5.7.

Figure 5.8 shows the results of a constrained estimation, where we impose the restriction $\eta = 1$ to achieve equivalency between the Weibull and exponential distributions. Notice that the output no longer shows standard errors, test statistics, and confidence intervals for η, as this is no longer an estimated parameter.

Constrained optimization can be particularly helpful for the purpose of computing the likelihood ratio test (see Chapter 6). One should realize that the estimates of the constrained model are not true ML estimates if the constraints happen to be false. I shall have more to say about constrained optimization when we discuss the Lagrange multiplier test in Chapter 6.

```
. use "C:\data\Example 5.10.dta", clear
. do "C:\data\weibull2.do"
. program define weibull2
  1.         version 1.0
  2.         args lnf beta eta
  3.         quietly replace `lnf'=ln(`eta')-`eta'*ln(`beta')+(`eta'-1)*ln($ML_
> y1)- ///
>         ($ML_y1/`beta')^`eta'
  4. end program
.
end of do-file
. eq beta: y=
. eq eta:
. ml model lf weibull2 beta eta
. ml max
initial:       log likelihood =     -<inf>  (could not be evaluated)
feasible:      log likelihood = -4.8385422
rescale:       log likelihood = -4.8385422
rescale eq:    log likelihood =  484.28818
Iteration 0:   log likelihood =  484.28818
Iteration 1:   log likelihood =  512.67475
Iteration 2:   log likelihood =  515.46236
Iteration 3:   log likelihood =    515.478
Iteration 4:   log likelihood =    515.478
                                                Number of obs   =       1000
                                                Wald chi2(0)    =          .
Log likelihood =    515.478                     Prob > chi2     =          .
```

y	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
beta						
_cons	.2398657	.0066124	36.28	0.000	.2269057	.2528257
eta						
_cons	1.206755	.0299909	40.24	0.000	1.147973	1.265536

Figure 5.7: Unconstrained MLE of the 2-Weibull PDF

```
. cons 1 [eta]_cons=1
. ml model lf weibull2 beta eta, const(1)
. ml max
initial:       log likelihood =     -<inf>  (could not be evaluated)
feasible:      log likelihood = -4.8385422
rescale:       log likelihood = -4.8385422
rescale eq:    log likelihood =  484.28818
Iteration 0:   log likelihood =  484.28818
Iteration 1:   log likelihood =  489.05092
Iteration 2:   log likelihood =  489.42715
Iteration 3:   log likelihood =  489.42827
Iteration 4:   log likelihood =  489.42827
                                                Number of obs   =       1000
                                                Wald chi2(0)    =          .
Log likelihood =  489.42827                     Prob > chi2     =          .
 ( 1)  [eta]_cons = 1
```

y	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
beta						
_cons	.2255016	.007131	31.62	0.000	.2115251	.239478
eta						
_cons	1	.	.	.	.	.

Figure 5.8: Constrained MLE of the 2-Weibull PDF

```
. estat vce
Covariance matrix of coefficients of ml model
```

e(V)	beta _cons	eta _cons
beta _cons	.00004372	
eta _cons	.00006157	.00089946

```
. estat vce, corr
Correlation matrix of coefficients of ml model
```

e(V)	beta _cons	eta _cons
beta _cons	1.0000	
eta _cons	0.3105	1.0000

Figure 5.9: VCE for the 2-Parameter Weibull PDF

5.9.4 The Variance-Covariance Matrix

In certain instances, it is useful to obtain the variance-covariance matrix of the estimators. Stata has a simple command for this:

`estat vce` [`, covariance correlation equation(`*spec*`)`]

The command is run after `ml maximize`. By default, Stata displays variances and covariances. If instead you would like to see the correlations among the estimates, then you should specify the `correlation` option. The `equation(`*spec*`)` option allows you to show the VCE for only certain model equations.

> **Example 5.11.** Figure 5.9 shows the VCE for the unconstrained 2-parameter Weibull distribution described in the previous example. The first part of the output shows the VCE proper. Taking the square root of the diagonal elements of this matrix will give you the standard errors reported in Figure 5.7. The second part of the output shows the VCE converted to a correlation matrix.

5.10 The EM Algorithm*

The algorithms that we have considered so far do not have special facilities for dealing with missing data. Such an algorithm does exist, however, and it is called the EM or **expectation-maximization** algorithm (Dempster et al. 1977). Since this algorithm plays an important role in several applications of MLE, it is worthwhile sketching how it works.

Before doing so, it is useful to say something about the meaning of the term "missing data." In the context of the EM algorithm, missingness may mean that part of the observed data is missing—the classical missing values problem. However, it may also mean that several of the variables are **latent variables**, i.e. variables that are inherently unobservable. We shall rely on this second meaning of the term when discussing some of the most noteworthy applications of the algorithm.

5.10.1 The Algorithm

Imagine that we have a set of observed data, y. Without loss of generality, we assume this data is continuous. It can be described through the PDF $f(y|\theta)$. The complete data is given by x; it consists of y as well as the unobserved values. This data is characterized by the PDF $g(x|\theta)$. The EM algorithm now proceeds in two steps. The expectation (or E-) step of the algorithm computes the following conditional expected value:

$$\begin{aligned} Q(\theta) &= Q(\theta|\theta_t) \\ &= \mathcal{E}\left[\ln g(x|\theta)|y, \theta_t\right] \end{aligned} \tag{5.15}$$

Here θ_t is the parameter value at the tth iteration. The essence of the E-step is that it seeks to obtain an expectation for the complete data set based on the observed data and the current value of the parameter.[36] The maximization (or M-) step of the algorithm maximizes $Q(\theta)$ with respect to θ. The algorithm alternates between the E and M steps. It uses the current estimate θ_t to produce $Q(\theta)$. This expectation is then used to update the estimate, with the updated estimate being used to form a new expectation, etc. The process continues until at least a local maximum has been obtained.

[36]The assumption is that the observed data are informative of the missing data, i.e. the observables and unobservables are related in a statistical sense. If this assumption fails, then so will the EM algorithm.

The EM algorithm has several desirable properties. First, the procedure is numerically stable because the likelihood function always increases with its steps. Second, the algorithm usually handles parameter constraints automatically. This means, for example, that estimates of probabilities are bounded between 0 and 1. The algorithm has some drawbacks, however. First, convergence tends to be slow, especially when there is lots of missing data and especially towards the end of the iterative process. Second, there are no direct estimates of the standard errors. These could be bootstrapped (see Chapter 7), computed via the Delta method (see Chapter 9), or computed from the observed Fisher information if there is an explicit log-likelihood for y (see Pawitan 2001).

5.10.2 Applications

Finite Mixture Models

One of the most important applications of the EM algorithm is the estimation of finite **mixture models**. A mixture model arises when a unit can come from one of several groups (also called components), each of which is described by a density $f_j(y|\theta_j)$, where $j = 1 \cdots J$ and y is the value of the variable of interest.[37] The different densities are mixed by way of **mixing parameters**, π_j, such that $\sum_j \pi_j = 1$ and

$$f(y) = \sum_{j=1}^{J} \pi_j f_j(y|\theta_j)$$

If we knew from which group a particular unit was drawn, then we could apply the log-likelihood function for that group and estimation would be straightforward. In this case, we would have data on an indicator variable that takes on the value 1 if unit i belonged to group j and zero otherwise:

$$z_{ij} = \begin{cases} 1 & \text{if } y_i \text{ is from group } j \\ 0 & \text{otherwise} \end{cases}$$

Thus, the complete data is given by $x_{ij} = (y_i, z_{ij})$. The log-likelihood for the

[37] In finite mixture analysis it is assumed that we know how many groups there are.

complete data is then

$$\begin{aligned}\ell(\theta, \pi|x) &= \sum_i \sum_j z_{ij} \ln [\pi_j f_j(y_i|\theta_j)] \\ &= \sum_i \sum_j z_{ij} \ln \pi_j + \sum_i \sum_j z_{ij} \ln f_j(y_i|\theta_j)\end{aligned}$$

The practical problem is, of course, that we do not have complete data. Specifically, while we can observe y_i, we lack any information about z_{ij}. However, the EM algorithm allows us to estimate mixture models without the indicator variable. In this case, the E-step evaluates

$$\begin{aligned}z'_{ijt} &= \mathcal{E}[z_{ij}|y_i, \theta_t, \pi_t] \\ &= \frac{\pi_{jt} f(y_i|\theta_{jt})}{\sum_k \pi_{kt} f(y_i|\theta_{kt})}\end{aligned}$$

Here π_{jt} is the value of π_j in the tth iteration. Likewise, θ_{jt} is the value of θ_j in that same iteration.[38]

The M-step produces estimates of θ_j and π_j. From the complete data log-likelihood function, we know that each θ_j requires maximization of the weighted log-likelihood:

$$\sum_i z'_{ij} \ln f_j(y_i|\theta_j)$$

This is done for each group separately. Estimates of π_j are obtained via maximization of

$$\sum_i z'_{ij} \ln \pi_j$$

[38]The proof of this result hinges on an application of Bayes' theorem. Here we treat π_j as the **prior** probability that a unit belongs to group j. We are interested in the conditional expectation of z_{ij}. Since, z_{ij} is a dummy variable, the expectation, i.e. z'_{ij}, is a probability. Indeed, this probability can be considered a **posterior** that comes about by considering the information contained in y_i and the current parameter values. Bayes' theorem now indicates that

$$\begin{aligned}z'_{ij} &= \Pr(z_{ij} = 1|y_i, \pi, \theta) \\ &= \frac{\Pr(y_i|z_{ij} = 1, \pi, \theta)\pi_j}{\Pr(y_i)}\end{aligned}$$

Since $\Pr(y_i) = \sum_k \Pr(y_i|z_{ik} = 1, \pi, \theta)\pi_k$ we obtain the expression for z'_{ijt} that is given in the text. Note that the application of Bayes' theorem suggests that the EM algorithm combines elements of MLE and Bayesian inference.

subject to the constraint $\sum_j \pi_j = 1$. This yields the following, intuitively plausible, estimator:[39]

$$\hat{\pi}_j = \frac{\sum_i z'_{ij}}{n}$$

(The estimator is intuitive because it implies that we get information about π_j by aggregating across the probabilities z'_{ij}, which are individual-level markers of the likelihood of belonging to group j.)

> **Example 5.12.** As an example we consider Gaussian or normal mixtures (e.g. Basford and McLachlan 1985). In the general case, Gaussian mixtures mix different multivariate normal distributions, each with a specific mean vector $\boldsymbol{\mu}_j$ and covariance structure $\boldsymbol{\Sigma}_j$. Here we focus on the simpler case of a univariate normal distribution.

[39] We obtain this result through constrained optimization (see also Chapter 6, where Lagrange multipliers are discussed in greater detail). The object function in this case is $f = \sum_i z'_{ij} \ln \pi_j$ whereas the constraint function is $c = \sum_j \pi_j - 1 = 0$. Using Lagrange multipliers, we optimize

$$\begin{aligned} h &= \sum_i z'_{ij} \ln \pi_j + \lambda(\sum_j \pi_j - 1) \\ &= \sum_i z'_{ij} \ln \pi_j + \lambda \sum_j \pi_j - \lambda \end{aligned}$$

The first partial derivative of h with respect to π_j is given by $\partial h / \partial \pi_j = \sum_i (z'_{ij}/\pi_j) - \lambda$. Setting this derivative equal to zero yields $\pi_j = (1/\lambda) \sum_i z'_{ij}$. Taking the first partial derivative of h with respect to λ and setting the result to zero yields $\sum_j \pi_j = 1$. Thus

$$\begin{aligned} \sum_j \pi_j &= 1 \Leftrightarrow \\ \sum_j (1/\lambda) \sum_i z'_{ij} &= 1 \Leftrightarrow \\ (1/\lambda) \sum_i (\sum_j z'_{ij}) &= 1 \Leftrightarrow \\ (1/\lambda) \sum_i 1 &= 1 \Leftrightarrow \\ n/\lambda &= 1 \end{aligned}$$

(Here we take advantage of the fact that z'_{ij} is a probability so that $\sum_j z'_{ij} = 1$.) From the last equation it follows immediately that $\lambda = n$. Back-substitution into the formula for π_j then yields the estimator.

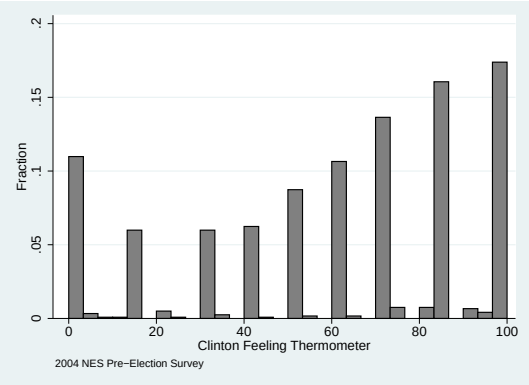

Figure 5.10: Clinton Feeling Thermometer Ratings

For this example we consider the feeling thermometer ratings of former President Bill Clinton taken in the pre-election survey of the 2004 American National Election Studies. These ratings are depicted in Figure 5.10. Inspecting this figure, we see two clear peaks, one arising at around 0 and the other at around 100 on the 101-point thermometer scale. The presence of two peaks is clearly inconsistent with the idea that a single normal distribution underlies the histogram. But what if we postulate two normal distributions, one on the left side of the histogram and another on the right side? Or, in mixture modeling speak, what if we assume two groups/components both of which can be characterized by normal PDFs, albeit with different means and variances?[40] In this case, we can run a 2-component finite mixture model.

The 2-component mixture model can be formulated in the following manner:

$$\pi_1 N(\mu_1, \sigma_1^2) + \pi_2 N(\mu_2, \sigma_2^2)$$

Here $\pi_2 = 1 - \pi_1$, so that $\theta = (\pi_1, \mu_1, \sigma_1^2, \mu_2, \sigma_2^2)$ is the set of parameters that need to be estimated. The complete data log-likelihood function (ignoring the constant $\ln \pi$) is given by

$$\ell \;=\; \sum_i \sum_j z_{ij} \ln \pi_j - .5 \sum_i \sum_j z_{ij} \left\{ \ln \sigma_j^2 + \frac{(y_i - \mu_j)^2}{\sigma_j^2} \right\}$$

[40]Normally, we would predict group membership from partisanship or other attributes. For the sake of this example, we assume that we do not have access to any of this information.

From here, π_j can be estimated via $\sum_i z'_{ij}/n$, where

$$\begin{aligned} z'_{i1} &= \frac{\pi_1 N(y_i|\mu_1,\sigma_1^2)}{\pi_1 N(y_i|\mu_1,\sigma_1^2) + (1-\pi_1)N(y_i|\mu_2,\sigma_2^2)} \\ z'_{i2} &= \frac{(1-\pi_1) N(y_i|\mu_2,\sigma_2^2)}{\pi_1 N(y_i|\mu_1,\sigma_1^2) + (1-\pi_1)N(y_i|\mu_2,\sigma_2^2)} \end{aligned}$$

We can estimate μ_j and σ_j^2 by optimizing the weighted log-likelihood

$$-.5 \sum_i z'_{ij} \left\{ \ln \sigma_j^2 + \frac{(y_i - \mu_j)^2}{\sigma_j^2} \right\}$$

This produces

$$\begin{aligned} \mu_j &= \frac{\sum_i z'_{ij} y_i}{\sum_i z'_{ij}} \\ \sigma_j^2 &= \frac{\sum_i z'_{ij}(y_i - \mu_j)^2}{\sum_i z'_{ij}} \end{aligned}$$

All of these steps have been programmed in the `denormix` command developed by Kolenikov. The abridged syntax for this command is

`denormix` *varname*`,` `ncomp(#)` `loglevel(2)` `difficult`

Here `ncomp` specifies the number of components, which we set to two.[41] The `loglevel` option influences how much detail is presented in the output; the current specification shows the iteration history and estimates for each component. The `difficult` option should look familiar, but it is useful to add because estimation of normal mixtures can be quite complex.

Figure 5.11 shows the results for the 2-component model. The first component has a large share ($\hat{\pi}_1 = .84$) and a high mean ($\hat{\mu}_1 = 69.5$) and variance ($\hat{\sigma}_1 = 23.6$). The second component has a much smaller share ($\hat{\pi}_2 = .16$) and a small mean ($\hat{\mu}_2 = 4.8$) and variance ($\hat{\sigma}_2 = 7.0$). This group clearly consists of Americans who have a strong dislike for Clinton, whereas the first group consists of people with a much more positive view of the former President.

[41] If the number of components is left unspecified, then the procedure will explore from one to `nmax(#)` components, where `nmax` is another option that can be specified.

```
. denormix clinton, ncomp(2) loglev(2) diff

Number of components:  2

initial:       log likelihood = -5857.1076
rescale:       log likelihood = -5857.1076
rescale eq:    log likelihood = -5857.1076
Iteration 0:   log likelihood = -5857.1076  (not concave)
Iteration 1:   log likelihood = -5856.4896  (not concave)
Iteration 2:   log likelihood = -5745.3696
Iteration 3:   log likelihood = -5742.6924
Iteration 4:   log likelihood = -5737.4933
Iteration 5:   log likelihood = -5736.4983
Iteration 6:   log likelihood = -5736.4959
Iteration 7:   log likelihood = -5736.4959

Log likelihood = -5736.4959             Number of obs   =    1202
------------------------------------------------------------------
 Parameters  |  Estimate    Std. Err.      [95% Conf. Interval]
-------------+----------------------------------------------------
Component 1
        Mean |    69.476    .8429029      67.82394    71.12806
       Sigma |  23.58122    .1922128      23.20449    23.95795
       Share |  .8407127    .0121515      .8168961    .8645292
-------------+----------------------------------------------------
Component 2
        Mean |  4.757874    .6297113      3.523663    5.992086
       Sigma |  6.995654    .2317376      6.541457    7.449851
       Share |  .1592873    .0121515      .1354708    .1831039
------------------------------------------------------------------
Note: standard errors are (asymptotic appoximations) by the delta
method and are obtained from std.errs of the auxiliary parameters.
```

Figure 5.11: Normal Mixture of the Clinton Feeling Thermometer

It should be noted that we could have computed the model with a different number of components, for example 3 instead of 2. The efficacy of such a specification against the leaner 2-component specification could be assessed by way of the entropy measures discussed in Chapter 8. In general, unless one has strong theoretical priors concerning the number of components, it is wise to try out models with different numbers of components to see which one fits the data the best.

Switching Regression Models

Switching regression models arise when the regression equation varies across units. We say that different units exist in different **regression regimes**. Sometimes we know with certainty which regression regime pertains to a given unit. Those situations are accommodated quite easily in the standard regression framework by creating one or more regime dummies and interacting them with the covariates. Here we focus on cases where it is not known with certainty which regression regime is relevant to the unit. Those cases can be handled quite nicely via the EM algorithm.[42]

As a simple example of a switching regression model let us consider a linear regression model with two possible regimes. This model is given by the following system of equations:

$$\begin{aligned} y_{i1} &= \boldsymbol{x}_{1i}\boldsymbol{\beta}_1 + \epsilon_{1i} \\ y_{2i} &= \boldsymbol{x}_{2i}\boldsymbol{\beta}_2 + \epsilon_{2i} \\ y_{3i} &= \boldsymbol{x}_{3i}\boldsymbol{\beta}_3 + \epsilon_{3i} \\ \epsilon_{1i} &\sim N(0, \sigma_1^2) \\ \epsilon_{2i} &\sim N(0, \sigma_2^2) \\ \epsilon_{3i} &\sim N(0, 1) \end{aligned}$$

Here y_{i1}-y_{i3} are latent variables—we do not observe them. What we observe instead is y_i, which may either represent y_{i1} or y_{i2}, depending on the regression regime. Which regime applies depends on y_{i3}. This variable sorts units

[42] I will focus on the relatively simple case of exogenous switching. Endogenous switching models are more complex because the disturbance associated with the switching equation is now correlated with the outcome variable of interest (see Maddala and Nelson 1975 and the voluminous literature following that paper).

into regimes by way of the following mechanism:

$$y_i = \begin{cases} y_{i1} & \text{if } y_{i3} < 0 \\ y_{i2} & \text{if } y_{i3} \geq 0 \end{cases}$$

Let $\lambda_i = \Phi(-\mathbf{x}_{i3}\boldsymbol{\beta}_3)$ be the probability that $y_i = y_{i1}$. Then $1 - \lambda_i$ is the probability that $y_i = y_{i2}$, where $\Phi(.)$ denotes the standard normal CDF.[43] The probability density function for a particular unit is given by the following mixture:

$$h(y_i) = \lambda_i \phi_1(y_{i1}) + (1 - \lambda_i)\phi_2(y_{i2})$$

where $\phi(.)$ is the standard normal PDF and

$$\begin{aligned} \phi_1(y_{i1}) &= \frac{1}{\sqrt{2\pi\sigma_1^2}} \exp\left\{-\frac{1}{2}\frac{(y_{i1} - \boldsymbol{x}_{i1}\boldsymbol{\beta}_1)^2}{\sigma_1^2}\right\} \\ \phi_2(y_{i2}) &= \frac{1}{\sqrt{2\pi\sigma_2^2}} \exp\left\{-\frac{1}{2}\frac{(y_{i2} - \boldsymbol{x}_{i2}\boldsymbol{\beta}_2)^2}{\sigma_2^2}\right\} \end{aligned}$$

As the equation for $h(y_i)$ reveals, the switching regression model in essence is a mixture model. As such, it can be estimated using the same tool as the finite mixture model, namely the EM algorithm. In the estimation, the E-step involves estimating the classification probabilities λ_i. In the M-step, we estimate $(\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \sigma_1^2, \sigma_2^2)$. In practice, this means that we use a combination of OLS and WLS to obtain the parameter estimates.

The details of the estimation process are complex. Since we usually leave those details to a computer program you can safely skip the current paragraph and move to the example. However, if you are interested in how the computer finds solutions to the 2-component switching regression model than you should read the breakdown of the computations that follows.

1. Use the initial estimate of $\boldsymbol{\beta}_3$, to obtain an estimate of $\lambda_i = \Phi(-\boldsymbol{x}_{i3}\boldsymbol{\beta}_3)$.
2. Using λ_i as well as the initial estimates of $\boldsymbol{\beta}_1$, $\boldsymbol{\beta}_2$, σ_1^2, and σ_2^2, obtain a set of weights:

$$\begin{aligned} w_{i1} &= \lambda_i \left[\frac{\phi_1(y_{i1})}{h(y_i)}\right] \\ w_{i2} &= (1 - \lambda_i) \left[\frac{\phi_2(y_{i2})}{h(y_i)}\right] \end{aligned}$$

[43] If you skip ahead to Chapter 9, you will notice that this is a probit specification, although in principle another specification (e.g. logit or gompit) could be chosen as well.

Compile the weights into matrices $\boldsymbol{W}_j = \texttt{diag}(w_{1j}, w_{2j} \cdots w_{nj})$ (for $j = 1, 2$).

3. Update $\boldsymbol{\beta}_j$ (for $j = 1, 2$) using WLS:

$$\boldsymbol{\beta}_j = \left[\boldsymbol{X}'_j \boldsymbol{W}_j \boldsymbol{X}_j\right]^{-1} \boldsymbol{X}'_j \boldsymbol{W}_j \boldsymbol{y}$$

4. Update $\boldsymbol{\beta}_3$ by first computing the residuals

$$e_{i3} = \boldsymbol{x}_{i3}\boldsymbol{\beta}_3 - w_{i1}\frac{\phi_3(0)}{\lambda_i} + w_{i2}\frac{\phi_3(0)}{1-\lambda_i}$$

and then computing the OLS estimator

$$\boldsymbol{\beta}_3 = \left[\boldsymbol{X}'_3 \boldsymbol{X}_3\right]^{-1} \boldsymbol{X}'_3 \boldsymbol{e}_3$$

5. Update σ_j^2 (for $j = 1, 2$) using

$$\sigma_j^2 = (\sum_i w_{ij})^{-1} (\boldsymbol{y}_j - \boldsymbol{X}_j\boldsymbol{\beta}_j)' \boldsymbol{W}_j (\boldsymbol{y}_j - \boldsymbol{X}_j\boldsymbol{\beta}_j)$$

These steps are repeated until convergence has been reached.

Example 5.13. To what extent are candidate evaluations driven by issues or candidate traits? Imagine that there are two types of voters, those driven (mostly) by issues and those driven (mostly) by traits.[44] We could hypothesize that a voter's type depends on his or her level of political sophistication as well as partisanship, although this relationship is probably far from deterministic.

To test these ideas, I ran a 2-component switching regression model on feeling thermometer ratings of George W. Bush in the 2000 NES. Rather than performing the computations myself, I relied on the `switchr` command developed by Zimmerman. This command executes the EM algorithm for the normal linear regression model with two groups. The abbreviated syntax for the command is

```
switchr eq1 eq2
```

[44] Working within the switching regression framework we have to assume that these types are mutually exclusive, which may not be true.

Here `eq1` is the outcome equation, i.e. the equation for y_i. The `switchr` command expects that you include both $\boldsymbol{x}_{1i}$ and $\boldsymbol{x}_{2i}$ in the outcome equation; the command estimates effects for both sets of predictors in both groups. `eq2` is the classification equation. The command expects that you provide initial estimates for λ_i, which I have placed in the variable `guessprob`; this variable is nothing more than a set of random draws from the uniform (0,1) distribution.

The results from the classification equation (Figure 5.12) reveal that there is an average probability of .189 of belonging to the first group. Whether one belongs to this group depends on both political knowledge and partisanship. Knowledge (`know`) has a positive effect, implying that more knowledgeable respondents were more likely to belong to the first group. Partisanship (`pid`) has a negative effect, which means that Democrats are more likely to be included in the first group than Republicans.

Moving to the outcome equation (Figure 5.13), we see that in the first group issues (`bushissue`) but not traits (`bushtrait`) exert a strong significant effect such that evaluations of Bush decreased the further a respondent was removed from him on the issues. As such, one could dub the first group the issue voting group. In the second group, issues remain statistically significant but their effect is much smaller than in the first group. By contrast, traits now exercise a powerful effect, such that respondents with positive trait evaluations of Bush also tended to give him a favorable overall evaluation.

Of the control variables, partisanship, `age`, education (`educ`), and household income (`hhinc`) matter in the second but not in the first group. Race (`white`) is statistically significant in the first but not in the second group. Gender (`male`) and being a born again Christian (`bornagain`) did not reach statistical significance in either group.

```
. eq main: bushfeel pid male white bornagain age educ hhinc
bushtrait bushissue
. eq regime: guessprob pid know
. switchr main regime
  (output omitted)

Done ...Here are the probabilities of being in the first component
regression..
> ..
                          __00000D
```

Percentiles		Smallest		
1%	.0210541	.0210541		
5%	.0256258	.0210541		
10%	.0282068	.0256258	Obs	192
25%	.0499333	.0256258	Sum of Wgt.	192
50%	.1181623		Mean	.1889695
		Largest	Std. Dev.	.1471338
75%	.3106112	.4489573		
90%	.400166	.4654328	Variance	.0216484
95%	.4325693	.4654328	Skewness	.4262418
99%	.4654328	.4654328	Kurtosis	1.664794

```
Final Results Follow
Here is the regression for the switching eq'n
```

Source	SS	df	MS		
Model	74.8629927	2	37.4314963	Number of obs	192
Residual	31.7232518	189	.167847893	F(2, 189)	223.01
				Prob > F	0.0000
				R-squared	0.7024
				Adj R-squared	0.6992
Total	106.586244	191	.558043165	Root MSE	.40969

guessprob	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
pid	-.2827471	.0133915	-21.11	0.000	-.3091631	-.2563311
know	.0415375	.0177693	2.34	0.020	.0064858	.0765892
_cons	-.3359943	.0675534	-4.97	0.000	-.4692498	-.2027388

```
On iter 165 the mean absolute change in the probability vector is : 0.00000
Average of the probability vector  is: 0.189
```

Figure 5.12: Classification Equation for Bush Evaluations

```
Linear regression                               Number of obs =     192
                                                F(  9,   182) =   10.77
                                                Prob > F      =  0.0000
                                                R-squared     =  0.3554
                                                Root MSE      =  24.419
```

bushfeel	Coef.	Robust Std. Err.	t	P>|t|	[95% Conf.	Interval]
pid	-3.198905	2.666659	-1.20	0.232	-8.460447	2.062637
male	-3.817474	8.228825	-0.46	0.643	-20.05364	12.41869
white	-19.56147	9.79631	-2.00	0.047	-38.89042	-.2325295
bornagain	4.278938	9.869381	0.43	0.665	-15.19418	23.75206
age	.0314668	.1946505	0.16	0.872	-.3525949	.4155285
educ	1.309318	2.989205	0.44	0.662	-4.588636	7.207271
hhinc	.8641361	.7870186	1.10	0.274	-.6887178	2.41699
bushtrait	9.071761	8.06687	1.12	0.262	-6.844851	24.98837
bushissue	-21.30266	4.617819	-4.61	0.000	-30.41401	-12.19131
_cons	61.21047	28.05726	2.18	0.030	5.851142	116.5698

```
Second component regression
(sum of wgt is   1.5574e+02)
Linear regression                               Number of obs =     191
                                                F(  9,   181) =  129.94
                                                Prob > F      =  0.0000
                                                R-squared     =  0.8322
                                                Root MSE      =  9.1184
```

bushfeel	Coef.	Robust Std. Err.	t	P>|t|	[95% Conf.	Interval]
pid	3.408126	.3800508	8.97	0.000	2.658226	4.158026
male	-.6375646	1.413394	-0.45	0.652	-3.426413	2.151283
white	3.008486	2.194639	1.37	0.172	-1.321881	7.338854
bornagain	2.080209	1.367262	1.52	0.130	-.6176146	4.778032
age	.1068003	.0444183	2.40	0.017	.0191559	.1944446
educ	-1.957422	.4941237	-3.96	0.000	-2.932405	-.982438
hhinc	.5391276	.1904776	2.83	0.005	.1632854	.9149697
bushtrait	20.84556	1.361557	15.31	0.000	18.159	23.53213
bushissue	-3.972292	1.604954	-2.48	0.014	-7.139118	-.8054666
_cons	12.2864	5.103363	2.41	0.017	2.216659	22.35613

```
(1518 real changes made)
This Switching Regression took 10 seconds.
```

Figure 5.13: Outcome Equation for Bush Evaluations

Chapter 6

Hypothesis Testing

One of the nicest aspects of MLE is that it provides a unified approach to estimation and hypothesis testing—one of the reasons why Gary King (1989) considers MLE a "unifying political methodology". In discussing hypothesis testing, it is useful to distinguish between tests of joint hypotheses, which involve multiple parameters, and tests of simple hypotheses, which involve only one parameter. Tests that fall into the former category include the likelihood ratio (LR), Wald (W), and Lagrange multiplier (LM) tests. All of these procedures can also be used to test simple hypotheses, although such hypotheses are most frequently tackled by way of Wald or score tests.

6.1 Joint Hypothesis Tests

When we speak of a joint hypothesis test we mean that several parameters are tested simultaneously. A common variant of such a test is when we set a subset of the parameters equal to zero. For instance, if $\boldsymbol{\theta'} = (\theta_1, \theta_2, \theta_3, \theta_4)$, then we might wish to test the hypothesis:

$$H_0 : \theta_1 = \theta_2 = 0$$

Another common variant of joint hypothesis testing involves linear restrictions (see also Chapter 6.1.2). Here a parameter is modeled as a linear function of a subset of the other parameters. For instance,

$$H_0 : \theta_1 = \theta_2$$

is an equality constraint, which may also be formulated as $H_0 : \theta_1 - \theta_2 = 0$.

Table 6.1: Summary of Different Test Approaches

	Estimated Models	
Method	**Unrestricted**	**Restricted**
Likelihood Ratio	Yes	Yes
Wald	Yes	No
Lagrange Multiplier	No	Yes

In the ML framework, three alternative approaches are used for testing joint hypotheses. These are the LR, W, and LM approaches, with the first two being particularly common.[1] These tests are *asymptotically equivalent*: as $n \to \infty$, they all produce the same results. In small samples, however, they may produce different conclusions since they are based on different logics.

The differences between the tests are spelled out in Table 6.1. These differences concern the models need to be estimated in order to perform the test, where the key distinction is between restricted and unrestricted models. The restricted model incorporates the hypothesized values of the parameters under H_0, while the unrestricted model does not and leaves these parameters free for estimation. We can also say that the restricted model is *nested* inside the unrestricted model, in that the restricted model is a structurally simpler form of the unrestricted model that is obtained by constraining some of the parameters to the values hypothesized under H_0.

> ***Definition 6.1.*** Two models are **nested** if both contain the same terms, including the same specifications of the response variable, and if one of them contains at least one additional term.

Considering the null hypothesis in these terms, we see that the LR test requires that both the restricted and unrestricted models are estimated, whereas the W test requires estimation only of the unrestricted model, and the LM test requires estimation only of the restricted model.

The tests also consider different information, as is illustrated in Figure 6.1. Specifically, the LR test considers the difference in the log-likelihood values of the unrestricted and restricted models. The LM test instead considers

[1] For an extensive discussion of these tests see Engle (1984) and Greene (2002).

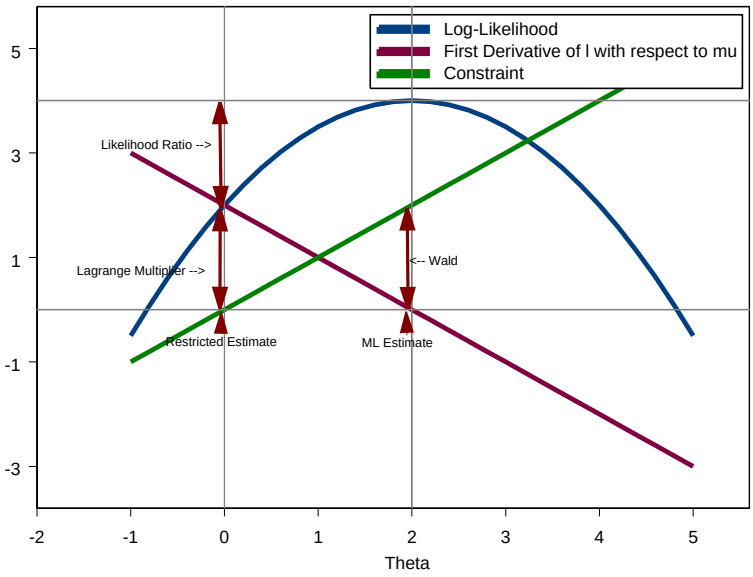

Figure 6.1: Three Bases for Hypothesis Tests

the gradient (first derivative of the log-likelihood function) at the restricted parameter value. Finally, the W test considers the difference between the ML estimate and the constrained value of the parameter. We now discuss these tests in greater detail.

6.1.1 The Likelihood Ratio Test

The LR test is based on the evaluation of both the unrestricted and restricted models. Each of these models yields a value of the likelihood function. As its name suggests, the likelihood ratio test considers the ratio of these two values.[2] Specifically, the likelihood ratio is defined as

$$\lambda = \frac{\mathcal{L}_R}{\mathcal{L}_U} \tag{6.1}$$

[2]The test is justified by the Neyman-Pearson lemma, named after the statisticians Jerzy Neyman and Egon Pearson. For a discussion of this lemma see e.g. Lehmann (1986).

where $\mathcal{L}_R$ and $\mathcal{L}_U$ are the values of the likelihood functions of the restricted and unrestricted models, respectively. The fit of the unrestricted model can be no worse than that of the restricted model: $\mathcal{L}_U \geq \mathcal{L}_R$. Consequently, $0 < \lambda \leq 1$. If the restrictions imposed by H_0 are correct, then $\hat{\boldsymbol{\theta}}_R = \hat{\boldsymbol{\theta}}_U$,where $\hat{\boldsymbol{\theta}}_U$ denotes the estimates of the unrestricted model and $\hat{\boldsymbol{\theta}}_R$ the estimates of the restricted model. Put simply, the ML estimates coincide with the hypothesized values of the parameters under H_0. In this case, $\mathcal{L}_R = \mathcal{L}_U$ and $\lambda = 1$. However, if the restrictions implied by H_0 are incorrect, then $\hat{\boldsymbol{\theta}}_R \neq \hat{\boldsymbol{\theta}}_U$. Moreover, since the restricted estimates are no longer the true ML estimates, $\mathcal{L}_R < \mathcal{L}_U$ and λ will be small. Thus, smaller values of the likelihood ratio indicate evidence against the restricted model and therefore H_0.

We want to be able to obtain p-values, so we need to know the sampling distribution of λ. This turns out to be quite complicated, but the asymptotic sampling distribution of negative two times the log of the likelihood ratio is relatively straightforward:

$$\begin{aligned} LR &= -2\ln\lambda \\ &\sim \chi^2_q \end{aligned} \tag{6.2}$$

where q is the number of restrictions imposed under H_0. Through some algebra it is possible to write this in terms of the log-likelihood functions of the restricted and unrestricted models:

$$\begin{aligned} LR &= -2\ln\lambda \\ &= -2\ln\left(\frac{\mathcal{L}_R}{\mathcal{L}_U}\right) \\ &= -2\left[\ln(\mathcal{L}_R) - \ln(\mathcal{L}_U)\right] \\ &= 2\left(\ell_U - \ell_R\right) \end{aligned} \tag{6.3}$$

Example 6.1. In example 5.10, we discussed the 2-parameter Weibull distribution and showed that it can be turned into the exponential distribution by restricting the shape parameter, η, to 1. We used Stata to estimate both the unrestricted and restricted versions of the Weibull model. Looking at Figure 5.6, we see that $\ell_U = 515.478$. Looking at Figure 5.7, it is clear that $\ell_R = 489.428$. By equation (6.3), $LR = 2 \times [515.478 - 489.428] = 52.100$. Referring this statistic to a

χ^2_1 distribution yields $p = .000$.[3] Thus, we soundly reject $H_0 : \eta = 0$. Apparently, for the present data, the 2-parameter Weibull distribution does not reduce to an exponential distribution.

I should note that there are actually two different types of LR test. The previous example showed an instance of the so-called **LR chi-squared test**, which is also designated as G^2. Here the unrestricted model contains all of the parameters that are relevant for the distribution. A second variant is the **(scaled) deviance** test. In this test, a given model is compared to a *saturated model*, i.e. a model that has one parameter per observation. Clearly, the saturated model now serves as the unrestricted model. Moreover with one parameter per observation, this model fits the data perfectly so that $\mathcal{L}_U = 1$ and $\ell_U = 0$. Thus, the deviance is given by

$$\begin{aligned} D &= 2\,(0 - \ell_R) \\ &= -2\ell_R \end{aligned} \tag{6.4}$$

Deviances play a role in some applications of MLE, although the LR chi-square test is more common.

6.1.2 The Wald Test

For the Wald test procedure, only the unrestricted model is estimated.[4] The test takes into consideration two pieces of evidence: (1) the distance between the unconstrained parameter estimates and the constrained values of those estimates and (2) sampling variance. Together, these pieces of information determine the fate of the null hypothesis.

To motivate the Wald test let me start by considering the concept of restricted model in some greater detail. Consider the parameter vector $\boldsymbol{\theta}$. A restricted or constrained model impose restrictions on the elements of this vector. For the sake of simplicity, let us limit ourselves to linear restrictions, since they are the most common.[5] These restrictions can be captured through the following system of equations:

$$\boldsymbol{R\theta} = \mathbf{r} \tag{6.5}$$

[3] There is one degree of freedom because only one parameter of the Weibull distribution is being restricted.

[4] The Wald test is named after the Hungarian mathematician Abraham Wald (1902-1950).

[5] Greene (2003) discusses applications of the Wald test to non-linear restrictions.

where $\mathbf{r}$ is a $q \times 1$ vector of values (q is the number of constraints) and $\mathbf{R}$ is a $q \times p$ design matrix that connects parameters (of which there are p in total) to the hypothesized values.

> **Example 6.2.** Consider the parameter vector $\boldsymbol{\theta}' = (\theta_1\ \theta_2\ \theta_3)$. Imagine that we want to test the hypothesis $H_0 : \theta_2 = \theta_3 = 0$. That is, we seek to impose two constraints. In this case, we can define $\mathbf{r}$ as the following 2×1 vector
>
> $$\mathbf{r} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$
>
> Further, $\mathbf{R}$ is the 2×3 matrix
>
> $$\mathbf{R} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$
>
> If our hypothesis is $H_0 : \theta_1 + \theta_2 - \theta_3 = k$, where k is a constant, then we impose one constraint (once two parameters have been estimated, the value of the third one follows from H_0). In this case,
>
> $$\mathbf{r} = \begin{pmatrix} k \end{pmatrix}$$
>
> and
>
> $$\mathbf{R} = \begin{pmatrix} 1 & 1 & -1 \end{pmatrix}$$

To evaluate the validity of the linear constraints one piece of evidence suggests itself immediately: the distance of the ML estimates from the constrained values. If the constraints are valid, then $\hat{\boldsymbol{\theta}}$ should satisfy them, at least approximately. However, if the constraints are invalid, then they should be far removed from $\hat{\boldsymbol{\theta}}$. To capture the difference between the constraints and the estimates we evaluate

$$\boldsymbol{R}\hat{\boldsymbol{\theta}} - \mathbf{r}$$

(We multiply $\hat{\boldsymbol{\theta}}$ by $\mathbf{R}$ to select the estimates of those parameters that are restricted under H_0.) If the null hypothesis is true, then we would expect

$$\boldsymbol{R}\hat{\boldsymbol{\theta}} - \mathbf{r} \approx \mathbf{0}$$

To obtain a proper test statistic, however, we would also want to include knowledge of the sampling variance of $\boldsymbol{R\hat{\theta}} - \mathbf{r}$. By a well-known result on linear composites, we know that

$$\mathbf{V}[\boldsymbol{R\hat{\theta}} - \mathbf{r}] \;=\; \mathbf{R}\mathbf{V}[\hat{\boldsymbol{\theta}}]\mathbf{R}'$$

This is the VCE based on the restrictions.

The Wald test statistic is now given by

$$\begin{aligned} W &= \left(\mathbf{R}\hat{\boldsymbol{\theta}} - \mathbf{r}\right)' \left(\mathbf{R}\mathbf{V}[\hat{\boldsymbol{\theta}}]\mathbf{R}'\right)^{-1} \left(\mathbf{R}\hat{\boldsymbol{\theta}} - \mathbf{r}\right) \\ &\sim \chi^2_q \end{aligned} \tag{6.6}$$

where it should be emphasized that the sampling distribution of W holds only asymptotically. If the constraints are correct and $\boldsymbol{R\hat{\theta}} - \mathbf{r} \approx \mathbf{0}$, then W tends to 0, providing support for H_0. If the constraints are invalid, then W will tend to be large and will exceed the critical value, providing a foundation for rejecting H_0.

Example 6.3. Consider again the 2-parameter Weibull distribution and the hypothesis $H_0 : \eta = 1$. Specifying the parameter vector as $\boldsymbol{\theta}' = (\beta\ \eta)$, the relevant linear restriction can be expressed as $\mathbf{R}\boldsymbol{\theta} = \mathbf{r}$, where $\mathbf{R} = (0\ 1)$ and $\mathbf{r} = 1$. Using the estimates from Figure 5.6, we know that $\hat{\beta} = .240$ and $\hat{\eta} = 1.207$. Thus

$$\begin{aligned} \mathbf{R}\hat{\boldsymbol{\theta}} - \mathbf{r} &= \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} 0.240 \\ 1.207 \end{bmatrix} - 1 \\ &= 1.207 - 1 \\ &= 0.207 \end{aligned}$$

From Example 5.11, we know that the VCE is given by

$$\mathbf{V}[\hat{\boldsymbol{\theta}}] \;=\; \begin{bmatrix} .00004372 & .00006157 \\ .00006157 & .00089946 \end{bmatrix}$$

Hence,

$$\begin{aligned} \mathbf{R}\mathbf{V}[\hat{\boldsymbol{\theta}}]\mathbf{R}' &= \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} .00004372 & .00006157 \\ .00006157 & .00089946 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\ &= .00089946 \end{aligned}$$

The Wald statistic is then given by

$$\begin{aligned} W &= \left(\mathbf{R}\hat{\boldsymbol{\theta}} - \mathbf{r}\right)' \left(\mathbf{R}\mathbf{V}[\hat{\boldsymbol{\theta}}]\mathbf{R}'\right)^{-1} \left(\mathbf{R}\hat{\boldsymbol{\theta}} - \mathbf{r}\right) \\ &= 0.207 \times \frac{1}{.00089946} \times 0.207 \\ &= 47.639 \end{aligned}$$

When referred to a χ^2_1-distribution we obtain $p = .000$, so that H_0 is rejected.

6.1.3 The Lagrange Multiplier Test

In the LM or **efficient score** test approach, only the restricted model is estimated. This test is particularly useful if the unrestricted model is complex and difficult to estimate. Rather than having to estimate that model, the LM test allows you to estimate the simpler restricted model and will then tell you if there is a need to lift the restrictions.

Constrained Optimization and Lagrange Multipliers

The LM test is based on Lagrange's ideas concerning constrained optimization.[6] If we seek to optimize the object function $f(x, y)$ subject to a constraint function $c(x, y) = 0$, then Lagrange suggested that we should create a function

$$h(x, y, \lambda) = f(x, y) + \lambda c(x, y)$$

where λ is the Lagrange multiplier; the function h is the Lagrangean. The constrained optimization solution is found by taking the partial derivatives of h with respect to x, y, and λ and by setting these equal to 0. In the process, we obtain values of λ, x, and y that provide the constrained maximum (or minimum).

Example 6.4. Consider the function $f(x, y) = 2 - x^2 - 2y^2$ subject to the constraint $x^2 + y^2 = 1$ or $c(x, y) = x^2 + y^2 - 1 = 0$ (i.e. x and y lie on the unit circle). We create the Lagrangean

$$\begin{aligned} h(x, y, \lambda) &= 2 - x^2 - 2y^2 + \lambda(x^2 + y^2 - 1) \\ &= 2 - x^2 - 2y^2 + \lambda x^2 + \lambda y^2 - \lambda \end{aligned}$$

[6] Joseph Louis Lagrange was an Italian mathematician who lived from 1736-1813.

Taking partial derivatives and setting them to 0, we have

$$\begin{aligned}\frac{\partial h}{\partial x} &= -2x + 2\lambda x = 0 \\ \frac{\partial h}{\partial y} &= -4y + 2\lambda y = 0 \\ \frac{\partial h}{\partial \lambda} &= x^2 + y^2 - 1 = 0\end{aligned}$$

(Notice that the last equation is equal to the constraint function $c = 0$. This is always true.) From the first equation we deduce that $\lambda = 1$. Substitution in the second equation gives $y = 0$ and substitution of this result in the third equation yields $x = \pm 1$. From the second equation we deduce that $\lambda = 2$. Substitution in the first equation yields $x = 0$ and substitution of this result in the third equation gives $y = \pm 1$. Hence, the constrained optima are $(0, -1)$, $(0, 1)$, $(-1, 0)$, and $(1, 0)$. (Notice that the unconstrained extreme is $(0, 0)$. Since this does not lie on the unit circle, this solution is not a constrained optimum.)

The Test

Now that we have seen the general idea behind Lagrange multipliers, how can it be used in the context of MLE? Consider again the constraint $\mathbf{R}\boldsymbol{\theta} = \mathbf{r}$. Let us start by rewriting this as

$$\mathbf{R}\boldsymbol{\theta} - \mathbf{r} = \mathbf{0}$$

We seek to optimize the log-likelihood function subject to this constraint. This can be captured through the Lagrangean

$$h(\boldsymbol{\theta}, \boldsymbol{\lambda}) = \ell(\boldsymbol{\theta}) + (\mathbf{R}\boldsymbol{\theta} - \mathbf{r})'\boldsymbol{\lambda}$$

Taking the derivative of this function with respect to $\boldsymbol{\theta}$ and $\boldsymbol{\lambda}$ and setting the results to $\mathbf{0}$ yields two equations

$$\begin{aligned}\frac{\partial h(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} &= \boldsymbol{\nabla} + \mathbf{R}'\boldsymbol{\lambda} = \mathbf{0} \\ \frac{\partial h(\boldsymbol{\theta})}{\partial \boldsymbol{\lambda}} &= \mathbf{R}\boldsymbol{\theta} - \mathbf{r} = \mathbf{0}\end{aligned}$$

If the constraints are valid, then imposing them should not produce a significant difference in the maximized value of the log-likelihood function. That is,

$\partial h/\partial \boldsymbol{\theta}$ should be identical to $\partial \ell/\partial \boldsymbol{\theta}$. Hence, $\partial h/\partial \boldsymbol{\theta} - \partial \ell/\partial \boldsymbol{\theta} = \partial h/\partial \boldsymbol{\theta} - \boldsymbol{\nabla} = \mathbf{R}'\boldsymbol{\lambda}$ should tend to 0, which means $\boldsymbol{\lambda} \to \mathbf{0}$. On the other hand, if the constraints are not valid, then the difference between $\partial h/\partial \boldsymbol{\theta}$ and $\partial \ell/\partial \boldsymbol{\theta}$ should be quite large, which would mean that $\boldsymbol{\lambda}$ is quite large.

We can develop this result further. From the derivatives of the Lagrangean with respect to $\boldsymbol{\theta}$ it follows that, at the constrained maximum, $\boldsymbol{\nabla} = -\mathbf{R}'\boldsymbol{\lambda}$. We have also seen that $\boldsymbol{\lambda} = \mathbf{0}$ if the constraint is valid. Thus, it follows (within sampling variability) that, if the constraint is valid,

$$\frac{\partial \ell_R}{\partial \hat{\boldsymbol{\theta}}_R} = \boldsymbol{\nabla}_R = \mathbf{0}$$

where underscore R implies that we are dealing with the restricted model.

We are now well on our way to the Lagrange multiplier test. All we have to do is to incorporate sampling variability and we are there. This is done in the following manner

$$\begin{aligned} LM &= \boldsymbol{\nabla'}_R \mathbf{V}[\hat{\boldsymbol{\theta}}_R] \boldsymbol{\nabla}_R \\ &\sim \chi^2_q \end{aligned} \tag{6.7}$$

Here $\mathbf{V}[\hat{\boldsymbol{\theta}}_R]$ is usually (although not necessarily) computed via the OPG method (see Chapter 5.6). As usual, the χ^2_q-distribution applies asymptotically and q refers to the number of restrictions.

Example 6.5. Consider once more the 2-parameter Weibull distribution and the hypothesis $H_0 : \eta = 1$. How would we test this hypothesis using the LM test approach? We begin by obtaining the elements of the gradient. After some tedious math we find

$$\begin{aligned} \frac{\partial \ell}{\partial \beta} &= -\frac{n\eta}{\beta} + \left(\frac{1}{\beta}\right)^{\eta} \frac{\eta}{\beta} \sum_i y_i^{\eta} \\ \frac{\partial \ell}{\partial \eta} &= \frac{n}{\eta} - n\ln(\beta) + \sum_i \ln(y_i) - \left(\frac{1}{\beta}\right)^{\eta} \ln\left(\frac{1}{\beta}\right) \sum_i y_i^{\eta} - \\ &\quad \left(\frac{1}{\beta}\right)^{\eta} \sum_i y_i^{\eta} \ln(y_i) \end{aligned}$$

At the constraint, $\eta = 1$, these elements become

$$\begin{aligned}
\frac{\partial \ell_R}{\partial \beta} &= -\frac{n}{\beta} + \frac{1}{\beta^2}\sum_i y_i \\
\frac{\partial \ell_R}{\partial \eta_R} &= n - n\ln(\beta) + \sum_i \ln(y_i) - \frac{1}{\beta}\ln\left(\frac{1}{\beta}\right)\sum_i y_i - \\
&\quad \frac{1}{\beta}\sum_i y_i \ln(y_i)
\end{aligned}$$

We can now substitute $\hat{\beta} = .240$ from the output shown in Figure 5.6. We also need to compute $\sum_i y_i^{\eta}$, $\sum_i \ln y_i$, and $\sum_i y_i^{\eta} \ln y_i$ subject to the constraint $\eta = 1$. These statistics work out to 225.502, -1911.983, and -265.313, respectively. Performing the computations we get $\partial \ell_R / \partial \beta = -251.709$ and $\partial \ell_R / \partial \eta_R = 279.701$ so that

$$\boldsymbol{\nabla}_R = \begin{bmatrix} -251.709 \\ 279.701 \end{bmatrix}$$

We now need to find the VCE. Here I will use the observed Fisher information matrix. The unconstrained Fisher information matrix is

$$\boldsymbol{\mathcal{I}}_o = \begin{bmatrix} -\frac{\partial^2 \ell}{\partial \beta^2} & -\frac{\partial^2 \ell}{\partial \beta \partial \eta} \\ -\frac{\partial^2 \ell}{\partial \eta \partial \beta} & -\frac{\partial^2 \ell}{\partial \eta^2} \end{bmatrix}$$

where

$$\begin{aligned}
-\frac{\partial^2 \ell}{\partial \beta^2} &= -\frac{n\eta}{\beta} + \frac{\eta}{\beta^2}\left\{(\eta+1)\sum_{i=1}^{n}\left(\frac{y_i}{\beta}\right)^{\eta}\right\} \\
-\frac{\partial^2 \ell}{\partial \beta \partial \eta} &= -\frac{\partial^2 \ell}{\partial \eta \partial \beta} \\
&= \frac{n}{\beta} - \frac{\eta}{\beta}\sum_{i=1}^{n}\left(\frac{y_i}{\beta}\right)^{\eta}\ln\left(\frac{y_i}{\beta}\right) - \frac{1}{\beta}\sum_{i=1}^{n}\left(\frac{y_i}{\beta}\right)^{\eta} \\
-\frac{\partial^2 \ell}{\partial \eta^2} &= \frac{n}{\eta^2} + \sum_{i=1}^{n}\left(\frac{y_i}{\beta}\right)^{\eta}\ln\left(\frac{y_i}{\beta}\right)^2
\end{aligned}$$

At the constraint $\eta = 1$, this becomes

$$\boldsymbol{\mathcal{I}}_{o_R} = \begin{bmatrix} -\frac{n}{\beta} + \frac{2}{\beta^3}\sum_{i=1}^{n} y_i & \frac{n}{\beta} - \frac{1}{\beta^2}\sum_{i=1}^{n}\left\{y_i \ln\left(\frac{y_i}{\beta}\right) + y_i\right\} \\ \frac{n}{\beta} - \frac{1}{\beta^2}\sum_{i=1}^{n}\left\{y_i \ln\left(\frac{y_i}{\beta}\right) + y_i\right\} & n + \frac{1}{\beta}\sum_{i=1}^{n} y_i \ln\left(\frac{y_i}{\beta}\right)^2 \end{bmatrix}$$

Substituting $\hat{\beta}$ and the data, we obtain

$$\boldsymbol{\mathcal{I}}_{o_R} = \begin{bmatrix} 12145.660 & -729.258 \\ -729.258 & 1509.101 \end{bmatrix}$$

We obtain the VCE by inverting this. Hence,

$$\begin{aligned} LM &= \boldsymbol{\nabla'}_R \{\boldsymbol{\mathcal{I}}_{o_R}\}^{-1} \boldsymbol{\nabla}_R \\ &= \begin{bmatrix} -251.709 & 279.701 \end{bmatrix} \begin{bmatrix} 12145.660 & -729.258 \\ -729.258 & 1509.101 \end{bmatrix}^{-1} \begin{bmatrix} -251.709 \\ 279.701 \end{bmatrix} \\ &= 52.992 \end{aligned}$$

When referred to a χ^2_1-distribution we obtain $p = .000$ and H_0 is rejected. Thus, the LM test, too, suggests that the 2-parameter Weibull distribution cannot be reduced to an exponential distribution for the data at hand.

6.1.4 Comparing the LR, W, and LM Tests

It is no coincidence that Examples 6.1, 6.3, and 6.5 all produce the same conclusion, since asymptotically the LR, W, and LM tests are equivalent. Thus, in a large sample it does not matter much which of these test procedures is used. Typically, researchers will opt for whatever is easiest to calculate or most robust for the problem at hand. There are some statisticians who have a distinct preference for the LR test because it is the only test that requires explicit estimation of both the restricted and unrestricted models. However, as long as the sample size is sufficiently large one could, under normal circumstances, also use the W and LM approaches.

In small samples, there is no guarantee that the three test procedures will produce the same result (see Berndt and Savin [1977]). Indeed, it is quite common for the tests to diverge rather than converge when the sample size is small. Unfortunately, when this happens it is not clear which test should be trusted. As a general rule, then, these tests are best used in large samples.

6.1.5 The LR and W Tests in Stata

Performing LR and W tests is computationally intensive. Luckily, Stata automates the computations through a few simple commands.[7] Specifically, the

[7] At present, Stata does not have a general automated routine for LM testing, although LM tests are available for certain statistical procedures.

`lrtest` command can be used to perform LR tests, while the `test` command handles W tests.

The abridged syntax for the `lrtest` command is as follows:

`lrtest` *modelspec1 modelspec2* [, `stats dir`]

Here *modelspec1* and *modelspec2* are the specifications of the unrestricted and restricted models. The `stats` option asks for statistical information about the model such as the AIC and BIC (see Chapter 8), while `dir` gives descriptive information. One obtains the model specifications by storing the estimates of separate estimations of the restricted and unrestricted models, using the `estimates store` command. The process is illustrated in Example 6.6.

> **Example 6.6.** Figure 6.2 shows the relevant syntax and test results when we let Stata perform the LR test that we conducted in Example 6.1. As you can see, the end result is exactly the same as what we computed by hand. That is, the LR test statistic (`LR chi2(1)`) comes out to 52.10. The number in parentheses is the degrees of freedom. The associated p-value (`Prob > chi2`) is .000, which means that the null hypothesis that $\eta = 1$ has to be rejected.

Performing a Wald test is even easier than this, since one only has to estimate one model, namely the unrestricted model. The `test` command can take on a number of different forms

(1) `test` *coeflist*

(2) `test` *exp=exp* [`= ...`]

(3) `test` [*eqno*] [`:` *varlist*]

(4) `test` [*eqno=eqno* [`= ...`]] [`:` *varlist*]

It is issued after running the unrestricted model. Example 6.7 illustrates the process.

> **Example 6.7.** Figure 6.3 shows how to perform a Wald test on $H_0 : \eta = 1$ after running the unrestricted 2-parameter Weibull model. We see that the test statistic that Stata computes is (within rounding error) identical to the one we computed in Example 6.3.

```
. ml model lf weibull2 beta eta
. ml max
  (output omitted)
. est sto U
. cons 1 [eta]_cons=1
. ml model lf weibull2 beta eta, const(1)
. ml max
  (output omitted)
. est store R
lrtest U R, stats

(log likelihoods of null models cannot be compared)
Likelihood-ratio test LR chi2(1)  = 52.10 (Assumption: R nested in
U) Prob > chi2 = 0.0000
```

Model	Obs	ll(null)	ll(model)	df	AIC	BIC
R	1000	.	489.4283	1	-976.8565	-971.9488
U	1000	.	515.478	2	-1026.956	-1017.14

Figure 6.2: Example of a Likelihood Ratio Test in Stata

```
. ml model lf weibull2 beta eta
. ml max
  (output omitted)
. test [eta]_cons=1
 ( 1)  [eta]_cons = 1
           chi2(  1) =   47.53
         Prob > chi2 =    0.0000
```

Figure 6.3: Example of a Wald Test in Stata

6.2 Simple Hypothesis Tests

Researchers are frequently interested in testing a single parameter. A common setup is:

$$H_0 : \theta = \theta_0$$
$$H_A : \theta \neq \theta_0$$

where θ_0 is a constant specifying the hypothesized value of the parameter. In many cases, $\theta_0 = 0$ so that the interest is in testing if a particular parameter is statistically distinguishable from zero. This is the default in all statistical packages. One-tailed tests can be formulated in a similar vein.

Testing a single parameter can be considered a special case of testing a subset of parameters, the distinctive feature being that the subset contains only one element. As such, the procedures for testing joint hypothesis are also applicable to testing a single parameter. While we could end our discussion here, it may be useful to determine what the specific test procedures look like for single parameters. Here, I focus on the Wald and score tests, although one could also use the LR test.

6.2.1 The z-Test

z-tests of single parameters are widely used in statistics and the default in most statistical packages. They are a special case of the Wald test. Consider the null hypothesis $H_0 : \theta = \theta_0$. Applying the formula for the Wald test statistic now yields

$$\begin{aligned} W &= (\hat{\theta} - \theta_0) V[\hat{\theta}]^{-1} (\hat{\theta} - \theta_0)' \\ &= \frac{(\hat{\theta} - \theta_0)^2}{V[\hat{\theta}]} \\ &= \left(\frac{\hat{\theta} - \theta_0}{se(\hat{\theta})} \right)^2 \\ &= z^2 \end{aligned}$$

The last step comes about because we recognize $(\hat{\theta} - \theta_0)/se(\hat{\theta})$ as the z-transformation under H_0. Under the null hypothesis, we know that W is asymptotically distributed as χ^2_1, i.e. as a χ^2-variate with one degree of freedom. From mathematical statistics, we know that the square of a standard

normal variate also follows a χ^2 distribution with one degree of freedom. Hence the asymptotic distribution of z follows immediately, suggesting the following test statistic for the null hypothesis:

$$z = \frac{\hat{\theta} - \theta_0}{se(\hat{\theta})} \sim N(0,1) \tag{6.8}$$

This is usually described as a z-test of the parameter.

We can derive the distribution of the z-test statistic in a different (but related) manner. From the asymptotic properties of MLE we know that, under H_0, $\hat{\theta} \sim N(\theta_0, V[\hat{\theta}])$. It is then not difficult to demonstrate that $(\hat{\theta} - \theta_0)/se(\hat{\theta}) \sim N(0,1)$. It does not matter that $se(\hat{\theta})$ has to be estimated in practice, since the asymptotic distribution of T is still standard normal. Here we see an important advantage of MLE. Known asymptotic properties of MLEs can be used to derive the asymptotic sampling distribution of W.

Example 6.8. Consider again the 2-parameter Weibull distribution. As before, we seek to test $H_0 : \eta = 0$. From Figure 5.6 we know that $\hat{\eta} = 1.207$ with an estimated standard error of .030. Hence, the z-test statistic is:

$$z = \frac{1.207 - 1}{.030} = 6.900$$

When referred to a standard normal distribution this yields $p = .000$ and hence, H_0 is rejected. Notice that $z^2 = 47.61$, which is (within rounding error) the Wald test statistic that we computed in Examples 6.3 and 6.7.

6.2.2 The Score Test*

Whereas the z-test is related to the W test procedure, the score test is related to the LM test. This is true in the sense that both tests rely on the gradient or score function. The score test statistic is given by

$$T = \frac{S(\theta_0)}{\sqrt{\mathcal{I}_e(\theta_0)}} \sim N(0,1) \tag{6.9}$$

where θ_0 is the hypothesized value of the parameter of interest and $\mathcal{I}_e(\theta_0)$ is the expected Fisher information at the hypothesized value. The standard normal distribution applies asymptotically. This test is also known as Rao's test, since its development occurred mostly in the works of the statistician Rao.

Example 6.9. Consider the Poisson distribution with parameter $\mu > 0$. In Example 2.4, we saw that

$$\begin{aligned} \ell' &= \frac{\sum_i y_i}{\mu} - n \\ &= \frac{n}{\mu}(\bar{y} - \mu) \\ &= S(\mu) \end{aligned}$$

where S denotes the score. In Example 2.7, we further saw that

$$\mathcal{I}_e(\mu) = \frac{n}{\mu}$$

Now imagine that we have obtained a sample of $n = 10$ observations with $\bar{y} = 4$. We seek to test the hypothesis $H_0 : \mu = 5$. In this case,

$$\begin{aligned} S(\mu_0) &= \frac{10}{5}(4 - 5) \\ &= -2 \end{aligned}$$

and

$$\begin{aligned} \mathcal{I}_e(\mu_0) &= \frac{10}{5} \\ &= 2 \end{aligned}$$

The score test statistic is now

$$\begin{aligned} T &= \frac{-2}{\sqrt{2}} \\ &= -1.414 \end{aligned}$$

When referred to the standard normal distribution we obtain $p > .10$ so that H_0 cannot be rejected at the .05 level.

An important variant on the score test replaces the denominator with the square root of the observed Fisher information evaluated at $\hat{\theta}$ in lieu of θ_0. There is some evidence that this procedure may work better in some contexts (see Pawitan 2001: p. 247), although I am unaware of a general result to this effect.

6.2.3 Comparing the z and Score Tests*

One of the main advantages of the score test is that it is a **locally most powerful** (LMP) test. This means that it retains a high degree of statistical power even for small deviations from the hypothesized value.[8] A drawback of the score test is that we need to know the gradient, which is not always reported in statistical software and can be complex.

The z-test has the benefit that it can be computed from the estimate and its standard errors, which are key statistics that we need to compute and report anyway. Another benefit is that the test is usuaully computed for us in statistical software packages, so that our work is really minimized to nothing. A drawback is that the Wald test is not necessarily LMP in finite samples, whereas the score test is (both tests are asymptotically LMP—see Engle 1984).

[8] Almost any test, no matter how blunt, can detect large deviations from the hypothesized value. The ability to detect small deviations from the hypothesis, however, is not such a common attribute in testing. LMP tests provide this ability, making them quite potent tests. One way to demonstrate that the score test is LMP is to consider two competing hypotheses about θ. According to the first hypothesis, $H_0 : \theta = \theta_0$. According to the second hypothesis, $H_1 : \theta = \theta_1 = \theta_0 + \delta$, where δ is very small so that θ_1 is in the vicinity of θ_0. The log of the likelihood ratio is

$$\ln\left(\frac{\mathcal{L}(\theta_1)}{\mathcal{L}(\theta_0)}\right)$$

(We consider this statistic because it speaks to the relative merits of the two models.) Performing a Taylor expansion on this expression and retaining only the first term we get $S(\theta_0)\delta$. Since this depends on the score function, we see that tests based on the score function are quite powerful even when δ is small.

Chapter 7

Confidence Intervals

Confidence intervals are relevant to both estimation and hypothesis testing. In the context of estimation, they allow us to provide **interval estimates** rather than the point estimates we have discussed so far. Interval estimates reflect better the uncertainty inherent in basing estimates on a single sample. In the context of hypothesis testing, confidence intervals provide a traditional means of ascertaining if the hypothesized value seems reasonable. Specifically, if the hypothesized value falls outside the confidence interval, then we are inclined to reject the hypothesis.

Confidence intervals can be defined in the following manner (e.g. Kmenta 1997).

> ***Definition 7.1.*** The $100\times(1-\alpha)\%$ **confidence interval** for the parameter θ is a random interval about $\hat{\theta}$ such that the probability that the interval covers θ is equal to $1-\alpha$.

Here α is known as the **level of significance** and $1-\alpha$ is known as the **confidence coefficient**. The definition implies that if we drew an infinite number of samples of size n and, for each sample, we computed the $100\times(1-\alpha)\%$ confidence interval in the identical manner, then $100\times(1-\alpha)\%$ of those intervals would contain the true parameter θ.

Many methods exist for computing confidence intervals. In some cases, an exact confidence interval can be obtained. This is not always true, however, and in general researchers may have to rely on approximate confidence intervals. Confidence intervals based on the Wald-test are widely employed

(they are the default in many statistical programs), but sometimes likelihood-based confidence intervals perform better. Finally, researchers can employ bootstrapping in order to obtain a confidence interval.

7.1 Exact Confidence Intervals

Exact confidence intervals are preferable when they are available. Availability of such intervals depends critically on knowledge of the precise sampling distribution of an estimator, which is not always known. The confidence intervals are exact in the sense that they do not rely on asymptotic theory or other approximations. They are also exact in terms of their **coverage probability**, which is the probability that a procedure for constructing confidence intervals covers the true parameter, θ. A well-known example can illustrate how exact confidence intervals may be derived.

> **Example 7.1.** Consider the normal distribution. In Example 3.1., we saw that the ML estimator of the mean is given by $\hat{\mu} = \bar{y}$. Without having to rely on limit theorems, we also know that $\bar{y} \sim N(\mu, \sigma^2/n)$.[1] From here it follows that
>
> $$\frac{(\bar{y} - \mu)\sqrt{n}}{\sigma} \quad \sim \quad N(0, 1)$$
>
> The problem is that σ is unknown and has to be estimated. Because we are substituting an estimated quantity for σ, the sampling distribution is affected. We now have
>
> $$\frac{(\bar{y} - \mu)\sqrt{n}}{\hat{\sigma}}$$
>
> For reasons that will become apparent in a moment, we divide the numerator and the denominator of this statistic by σ:
>
> $$\frac{(\bar{y} - \mu_0)\sqrt{n}/\sigma}{\sqrt{\hat{\sigma}^2/\sigma^2}}$$

[1] Remember that if y follows a normal distribution, then $\bar{y}$ follows a normal distribution as well, regardless of the sample size. This is because $\bar{y}$ involves a sum of independent, normally distributed random variables and we know that such a sum is itself normally distributed.

For reasons that will also become clear soon, we multiply both terms of the denominator by $n-1$:[2]

$$\frac{(\bar{y}-\mu_0)\sqrt{n}/\sigma}{\sqrt{(n-1)\hat{\sigma}^2/(n-1)\sigma^2}}$$

Notice that these manipulations do not alter anything. We introduce them because they lead to important distributional consequences. First, consider the numerator. This is identical to what we had when σ was assumed to be known. Hence, we know that the numerator follows a standard normal distribution. Next, consider the denominator. A well-known result in mathematical statistics is that $((n-1)\hat{\sigma}^2)/\sigma^2 \sim \chi^2_{n-1}$. Substitution of these results yields:

$$\frac{N(0,1)}{\sqrt{\chi^2_{n-1}/(n-1)}}$$

From mathematical statistics, we know that this ratio follows a Student's t-distribution with $n-1$ degrees of freedom. Thus

$$\frac{(\bar{y}-\mu)\sqrt{n}}{\hat{\sigma}} \quad \sim \quad t_{n-1}$$

We let $t_{n-1,\alpha/2}$ be the point beyond which the t_{n-1}-distribution attains a probability of $\alpha/2$. Due to the symmetry of the t-distribution, it follows that the probability to the left of $-t_{n-1}$ is also $\alpha/2$. Combining these tail areas we get

$$\Pr\left(-t_{n-1,\alpha/2} < \frac{(\bar{y}-\mu)\sqrt{n}}{\hat{\sigma}} < t_{n-1,\alpha/2}\right) \quad = \quad 1-\alpha$$

It is now a matter of manipulating this equation to obtain the $(1-\alpha)\times 100\%$ confidence interval for μ:

$$\Pr\left(\bar{y} - t_{n-1,\alpha/2}\frac{\hat{\sigma}}{\sqrt{n}} < \mu < \bar{y} + t_{n-1,\alpha/2}\frac{\hat{\sigma}}{\sqrt{n}}\right) \quad = \quad 1-\alpha$$

This is the exact confidence interval for μ. It holds for any sample size. Note that as $n \to \infty$, $t_{n-1} \to N(0,1)$ so that in large samples we may rely again on the standard normal distribution.

[2]We divide by $n-1$ instead of n because we lose a degree of freedom in estimating μ.

7.2 Wald Confidence Intervals

In many cases, the precise sampling distribution of an estimator in a finite sample is unknown. (Remember, one of the drawbacks of MLE is that finite properties are frequently unknown.) In this case, one can rely on asymptotic approximations to form a confidence interval. As you may recall, ML estimators follow an asymptotic normal distribution. As you also may recall, the square root of the Wald test statistic asymptotically follows a standard normal distribution. Using these asymptotic results, a $100 \times (1-\alpha)\%$ confidence interval presents itself immediately:

$$\Pr\left(-z_{\alpha/2} < \frac{\hat{\theta} - \theta}{se(\hat{\theta})} < z_{\alpha/2}\right) \quad = \quad 1 - \alpha$$

where $z_{\alpha/2}$ is a value of a standard normal variate such that $\alpha/2$ of the probability lies to the right of it. Rearranging terms we get

$$\Pr\left(\hat{\theta} - z_{\alpha/2}se(\hat{\theta}) < \theta < \hat{\theta} + z_{\alpha/2}se(\hat{\theta})\right) \quad = \quad 1 - \alpha \tag{7.1}$$

Equation (7.1) is known as the Wald confidence interval and most statistical packages will compute it by default. Specific intervals are obtained by setting $z_{\alpha/2}$. For example, the 95% Wald confidence interval is given by $\hat{\theta} \pm 1.96se(\hat{\theta})$, while the 90% confidence interval is given by $\hat{\theta} \pm 1.64se(\hat{\theta})$, and the 99% confidence interval is given by $\hat{\theta} \pm 2.57se(\hat{\theta})$.

> **Example 7.2.** Consider the binomial distribution with $n = 10$, $y = 2$, and parameter π. For this distribution, $\hat{\pi} = y/n = .2$. Further $V[\hat{\pi}] = n^{-1}[\hat{\pi}(1 - \hat{\pi})] = .016$. Thus, the 95% confidence interval is
>
> $$.2 - 1.96\sqrt{.016} < \pi < .2 + 1.96\sqrt{.016} \quad = \quad -.048 < \pi < .448$$

Example 7.2 shows an important property of Wald confidence intervals, namely that they are symmetrical around $\hat{\theta}$, in this case $\hat{\pi} = .2$. The example also shows an important drawback of such intervals, namely that they can extend to inadmissible values. In the binomial distribution, π is a probability so that we know it is bounded between 0 and 1. But the lower bound of the 95% C.I. is negative, which defies the laws of probability.

Apart from creating inadmissible bounds, the more general problem with Wald confidence intervals is that they depend on asymptotic normality. That

is fine when we are dealing with a large sample, but it becomes questionable when the sample size is small, as is the case in Example 7.2 (n is only 10). If the likelihood function is irregularly shaped (i.e. deviates from quadrature), then asymptotic normality may hold only in very large samples and may not be a safe assumption even in moderately sized data sets. Reliance on asymptotic distribution theory in those situations may cause the coverage probability to be less than we think (i.e. less than the nominal level, namely the α-level that we adopted).

7.3 Likelihood-Based Confidence Intervals

Likelihood-based confidence intervals, also known as likelihood ratio or profile likelihood confidence intervals, work with the likelihood ratio. For a given θ, the **relative likelihood** is given by

$$R(\theta) \;=\; \frac{\mathcal{L}(\theta)}{\mathcal{L}(\hat{\theta})}$$

If evaluated at the true θ, then $-2\ln[R(\theta)]$ follows an asymptotic χ^2_1 distribution (i.e. a χ^2-distribution with 1 degree of freedom). An approximate $100 \times (1-\alpha)\%$ C.I. is then given by the set of values of θ that comply with

$$-2\ln[R(\theta)] \;<\; \chi^2_{1,1-\alpha} \tag{7.2}$$

Equivalently, the $100 \times (1-\alpha)\%$ C.I. corresponds to values of θ such that $R(\theta) > \exp[-\chi^2_{1,1-\alpha}/2]$. Using the cutoffs of the χ^2_1 distribution, this means that the 90% C.I. consists of values for which $-2\ln[R(\theta)] < 2.71$, the 95% C.I. consists of values for which $-2\ln[R(\theta)] < 3.84$, and the 99% C.I. consists of values for which $-2\ln[R(\theta)] < 6.63$.

Unlike Wald confidence intervals, there is no closed-form solution for this interval. Instead, the interval will have to be found through an iterative procedure.[3]

[3]There are several algorithms for this. One of these finds the lower bound of the 95% confidence interval by optimizing

$$\Big(-2\ln\mathcal{L}(\theta) + 2\ln\mathcal{L}(\hat{\theta}) - 3.84\Big)^2 - \Big(\hat{\theta} - \theta\Big)$$

where θ is the current parameter value and $\hat{\theta}$ is the MLE. The upper bound is found by

One may wonder why we should go through with this computational complexity when Wald confidence intervals are readily available. The reason is that likelihood-based confidence intervals overcome some of the pitfalls of Wald confidence intervals. First, for probabilities, likelihood-based confidence intervals are bounded between 0 and 1 so that we do not have to worry about an interval extending into an inadmissable range. Second, likelihood-based confidence intervals tend to be closer to the alleged coverage probability than Wald confidence intervals, especially in smaller samples. In addition, likelihood-based confidence intervals are often shorter than the corresponding Wald intervals, i.e. they are more precise. Moreover, they work better than Wald confidence intervals when we are operating at the boundary of the parameter space.

Example 7.3. Consider again the binomial distribution from Example 7.2. What would the 95% likelihood-based confidence interval look like? It is actually quite easy to figure this out by creating an Excel worksheet with possible values of π. Since the binomial likelihood function is given by

$$\mathcal{L} = \binom{n}{y}\pi^y(1-\pi)^{n-y}$$

we know that $\mathcal{L} = .302$ when evaluated at $\hat{\pi}$ (we get this by substituting $\hat{\pi} = .2$, $n = 10$, and $y = 2$ into the function for $\mathcal{L}$). For the range of values in the Excel worksheet we can evaluate $\mathcal{L}$ and then $-2\ln[R(\pi)]$. It is then just a matter of determining the lowest value of π for which $-2\ln[R(\pi)] < 3.84$. Following this procedure yields

$$.037 < \pi < .499$$

as the 95% likelihood-based confidence interval.

---

optimizing

$$\left(-2\ln calL(\theta) + 2\ln\mathcal{L}(\hat{\theta}) - 3.84\right)^2 - \left(\theta - \hat{\theta}\right)$$

With modern computers, finding the bounds of the confidence interval has become very easy. Note, however, that we can only compute likelihood-based confidence intervals when we have access to the log-likelihood function. If we do not have this function, but just an approximation of it, then it is unwise to compute likelihood-based confidence intervals. This can happen, for example, in complex estimation problems.

Example 7.3 shows an important property of likelihood-based confidence intervals: unlike Wald intervals, they are not symmetrical about $\hat{\theta}$. In Example 7.3., the distance between $\hat{\pi}$ and the lower bound of the interval is .163; the distance of the MLE with the upper bound is larger than that, namely .299. This is a general attribute of likelihood-based confidence intervals.

7.4 The Bootstrap*

7.4.1 The Bootstrap Sampling Distribution

When the asymptotic sampling distribution is unknown, or if there is reason to be suspicious about the applicability of a known sampling distribution to the data at hand, then the bootstrap provides a useful framework for creating confidence intervals.[4] The advantage of the bootstrap is that it allows researchers to draw statistical inferences without having to make strong distributional assumptions and without having to derive analytical formulas for the parameters of the sampling distribution. Instead the sampling distribution is derived empirically and from there confidence intervals can be computed. The method works even when sample sizes are relatively small.[5]

The basic idea of bootstrapping is quite simple. As you should remember, one way to think of a sampling distribution is that it describes the values of an estimator calculated from an infinite number of samples of size n taken from a population. The bootstrap approach takes this idea literally. The complication, of course, is that we do not have access to the population. The bootstrap method, which was developed by Efron and others (Efron 1979; Efron and Tibshirani 1986; see also Davison and Hinkley 1997; Mooney and Duval 1993), treats the original sample as if it were the population and then

[4]The bootstrap is closely associated with the jackknife, which is another useful non-parametric procedure. As we shall see, the idea behind bootstrapping is to take repeated samples with replacement. The idea behind the jackknife is to drop subsets of observations one at a time and to ascertain how this influences the estimates (see Miller 1974; Quenouille 1956). The jackknife procedure is particularly useful for assessing the impact of influential observations, although it is also used to obtain sampling distributions for complex sampling designs.

[5]These notes describe only non-parametric bootstrap procedures. Here bootstrap samples are generated directly from the data. In parametric bootstrapping, a probability distribution is first fitted to the data and bootstrap samples are generated from this distribution. For a discussion of the parametric bootstrap see Davison and Hinkley (1997).

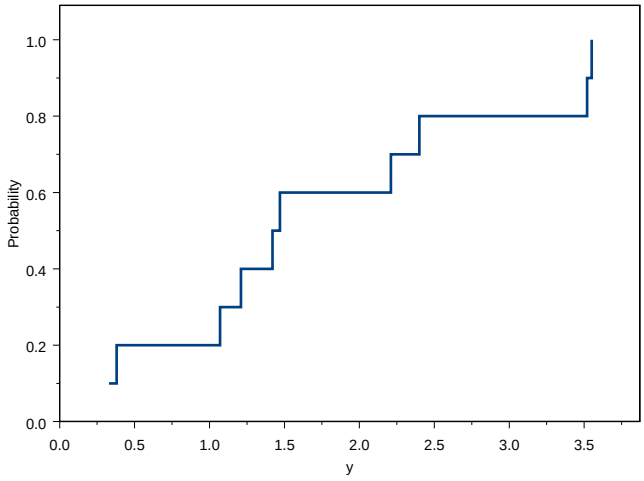

Figure 7.1: Empirical Distribution Function

takes repeated samples with replacement. It is essential that the sampling is with replacement and that the original sample is a reasonable representation of the population.

Bootstrapping occurs in five steps, which are now routinely carried out by computer programs such as Stata.

1. Create the **empirical distribution function** (EDF), which is the empirical probability distribution, $\hat{F}(x)$, that arises by placing a weight of $1/n$ on each of the observations in the sample. The EDF is the non-parametric MLE of the population distribution function, $F(x)$. It looks like a step function, with probabilities increasing in steps of $1/n$ at each successive value. Figure 7.1 shows an example of an EDF.

2. Draw a sample of size n with replacement from the EDF; this is known as a re-sample.[6] Note that sampling with replacement means that a particular value in the original sample may occur multiple times in a re-sample, while another value may not occur in that re-sample. This gives bootstrapping its variability.

[6]Some procedures set the re-sample size, n^*, lower than the original sample size but it is more common to set $n^* = n$.

Table 7.1: Hypothetical 2-Parameter Weibull Data

i	x_i	i	x_i
1	1.07	6	0.33
2	3.52	7	3.55
3	1.47	8	1.21
4	0.38	9	2.21
5	1.42	10	2.40

3. Compute the estimate of interest in the re-sample. This yields $\hat{\theta}_i$, which is a building block for the sampling distribution.

4. Repeat (2)-(3) k times, where k is a large number (typically 50-200 re-samples to compute standard errors and 1000 or more re-samples to compute confidence intervals—see Efron and Tibshirani [1986]).

5. Construct a probability distribution from the $\hat{\theta}_i$ values by giving each of the values $\hat{\theta}_1, \hat{\theta}_2 \cdots \hat{\theta}_k$ a probability of $1/k$. This is the bootstrapped estimate of the sampling distribution of $\hat{\theta}$, i.e. $\hat{F}(\hat{\theta})$.

Example 7.4. To illustrate the ideas of bootstrapping, consider the data from Table 7.1, which represent draws from the 2-parameter Weibull distribution considered in Chapter 5. The EDF for this data is displayed in Figure 7.1. Table 7.2 further shows three different bootstrap re-samples and their estimates of β and η. Notice that these re-samples differ because some values from Table 7.1 occur multiple times in a particular re-sample, while other values do not occur. For instance, the values 1.21 and 1.42 occur twice in the first re-sample, while the value 2.40 does not occur at all. It is this variation in the case selection that accounts for the variation in the parameter estimates that is observed in Table 7.2. Figure 7.2 shows the bootstrapped sampling distribution for $\hat{\eta}$ over 100 re-samples. Although this distribution is based on relatively few re-samples, it nevertheless indicates the risks of relying on asymptotic distribution theory when the sample size is small. Asymptotically, the distribution of $\hat{\eta}$ is normal, but the distribution that emerges in Figure 7.2 is not normal, displaying a positive skew.

Table 7.2: Three Bootstrap Re-Samples from Table 7.1

	Re-Sample No.		
Estimates	**1**	**2**	**3**
	.33	.33	.33
	.38	.33	.38
	1.07	.33	1.07
	1.21	.38	1.07
	1.21	1.07	1.21
	1.42	1.07	1.42
	1.42	1.42	1.42
	1.47	1.42	2.40
	3.52	2.40	3.55
	3.55	3.52	3.55
$\hat{\beta}$	1.74	1.33	1.83
$\hat{\eta}$	1.55	1.29	1.54

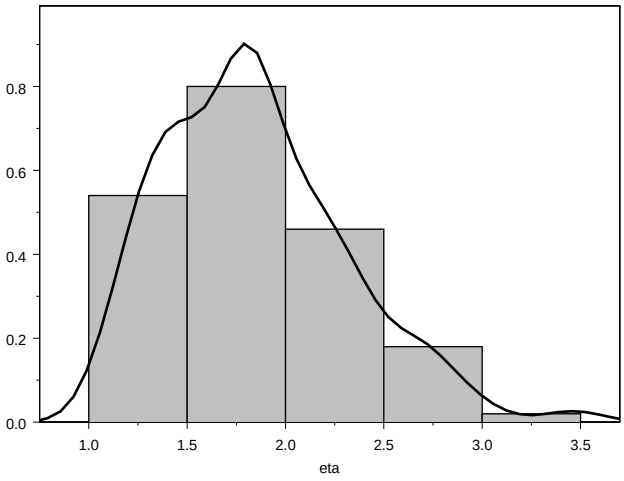

Figure 7.2: Bootstrapped Sampling Distribution of $\hat{\eta}$

One may wonder why bootstrapping is a valid methodology. The reasons can be found in two important asymptotic results. First, as n approaches the population size, then the EDF approaches the true population distribution (see Bickel and Freedman 1981). This makes sense because an infinitely large sample effectively is the population. This is an important result because it provides a solid statistical footing for treating the original sample as if it were the population. The second asymptotic result is that when the number of re-samples approaches infinity, $\hat{F}(\hat{\theta}) \rightarrow F(\hat{\theta})$ (see Babu and Singh 1983). This, too, makes sense. Because of the first asymptotic result, sampling from the EDF should be (nearly) the same as sampling from the true population distribution. Thus, a sampling distribution based on the EDF should be (almost) indistinguishable from a sampling distribution based on the population, assuming that n is large.[7]

Since applied researchers rarely operate in the world of asymptotics, it is useful to consider the practical sample size and re-sample requirements for bootstrapping. First, Efron and Tibshirani (1986) suggest 50-200 re-samples for computing standard errors and a minimum of 1000 re-samples for computing confidence intervals. In most cases, not much is gained from increasing the number of re-samples beyond 1000, so that figure is a reasonable baseline for computing confidence intervals. In terms of sample size, the consensus seems to be that a minimum of 30-50 observations is sufficient for a valid bootstrap. Indeed, the fact that bootstrapping works when the sample size is relatively small is one of its main attractions. This is particularly true in an MLE context, where reliance on asymptotic properties can be quite problematic if the sample size is small and where bootstrapping provides an alternative that usually works better.

7.4.2 Bootstrap Confidence Intervals

One of the most important applications of bootstrapping is in computing confidence intervals.[8] Several methods exist for computing bootstrapped confidence intervals of which the normal approximation, percentile, and bias

[7]We also assume that k is large, simply because a larger number of draws allows for a better approximation of the sampling distribution. Note that $\hat{F}(\hat{\theta})$ differs from $F(\hat{n})$ by at least an order of magnitude of $O(n^{-.5})$ and a greater magnitude if there are differences in skewness.

[8]Another important application is bias estimation. Unbiasedness is a key property of estimators, but one that is often difficult to ascertain. In the bootstrapping framework,

corrected percentile procedures are the most common.[9]

Normal Approximation

The normal approximation method is the bootstrap analog of Wald confidence intervals. This method is used when it is reasonable to assume that an estimator is normally distributed but when no analytic formula for the standard error exists. In this case, the standard error is computed via

$$\widehat{s.e.} = \sqrt{\frac{1}{k-1}\sum_i (\hat{\theta}_i - \bar{\theta})^2}$$

where $\bar{\theta}$ is the average bootstrap estimate. It has been demonstrated that the bootstrapped standard error is a consistent estimator of the true standard error. To obtain the $100 \times (1-\alpha) \times 100\%$ confidence interval one then uses

$$\hat{\theta} - z_{1-\alpha/2}\widehat{s.e.} < \theta < \hat{\theta} + z_{1-\alpha/2}\widehat{s.e.} \tag{7.3}$$

Example 7.5. Imagine that we are interested in the ratio of the mean placements of two political candidates, in this case Bush and Kerry in 2004. Such a ratio indicates how much greater (or smaller) the mean support for one candidate was relative to the other. Drawing inferences about ratios of means is quite complicated due to the fact that the ratio of two sample means is a biased estimator of the ratio of the population means (Rao and Beegle 1967). Bootstrapping provides a useful analytical framework for this problem.

In the 2004 American National Election Studies, the mean rating of Bush was 54.85 while the mean rating of Kerry was 53.02, making for a ratio of 1.035. To obtain the normal approximation bootstrap confidence interval on the ratio, we should issue the following Stata syntax:

---

the estimated bias of $\hat{\theta}$ is defined as

$$\hat{B} = \hat{\theta} - \bar{\theta}$$

where $\bar{\theta} = (1/k)\sum \hat{\theta}_i$ is the average of the k bootstrap parameter estimates (for examples see Davison and Hinkley 1997; Mooney and Duval 1993).

[9] See Davison and Hinkley (1997) and Mooney and Duval (1993) for a discussion of other bootstrap confidence interval procedures, including the percentile t method. The procedures discussed in these notes are easily implemented using Stata, as will be discussed in Example 7.5.

```
set seed 1

bootstrap ratio=r(ratio), reps(1000):  myratio bush kerry
```

The first instruction sets the seed for the random number generator that is used for creating re-samples. As a general rule, it is recommended that you set the seed yourself (in this case to the number 1) rather than leaving it up to the program. In this way, replicability of the results is ensured. The second instruction asks Stata to run the program `myratio` and run 1000 replications. By default, Stata presents the normal approximation confidence interval, in this case for a variable called ratio that stores the ratio produced by `myratio`.[10]

Figure 7.3 shows the results from the bootstrapping procedure. The bootstrapped estimate of the standard error is .030, yielding a normal-based 95% confidence interval from .977-1.092.

The normal approximation method is the oldest technique for computing bootstrap confidence intervals. It works very well—and sometimes better than the alternatives—when the sampling distribution of $\hat{\theta}$ is indeed approximately normal. However, if that assumption is unreasonable, as it may be especially in small samples, then the normal approximation confidence intervals may be incorrect. Moreover, the normal approximation method does not correct for any bias that may be present in the bootstrap distribution. Thus, normal approximations should be used with great care and it may be necessary to rely on one of the other methods for computing bootstrap confidence intervals.

Percentile Method

The percentile method overcomes the first, though not the second, limitation of normal approximations. This method takes advantage of the fact that an entire sampling distribution is generated during the bootstrap. We can simply take the $(\alpha/2) \times 100$th and $(1 - (\alpha/2)) \times 100$th percentiles of this sampling distribution to obtain the desired confidence interval. For example, with 1000 re-samples we could arrange the bootstrap estimates $\hat{\theta}_i$ from lowest to highest and set the lower bound of the confidence interval at the 25th

[10]By default, Stata computes the 95% C.I. To obtain another interval one can add `lev(#)` between the comma and the colon in the `bootstrap` command. Here # is a value between 10 and 99.99. For example, to obtain the 90% C.I. one could type `lev(90)`.

```
. do "C:\data\myratio.do"

. cap program drop myratio

. program myratio, rclass
  1.         syntax [varlist] [if] [, *]
  2.         tokenize ‘varlist’
  3.         tempname ymean
  4.         summarize ‘1’ ‘if’
  5.         scalar ‘ymean’=r(mean)
  6.         summarize ‘2’ ‘if’
  7.         return scalar ratio=‘ymean’/r(mean)
  8. end

.
end of do-file

. set seed 1

. bootstrap ratio=r(ratio), reps(1000): myratio bush kerry
(running myratio on estimation sample)

  (output omitted)

Bootstrap results                                 Number of obs      =      1191
                                                  Replications       =      1000

      command:  myratio bush kerry
        ratio:  r(ratio)

------------------------------------------------------------------------------
             |   Observed   Bootstrap                         Normal-based
             |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
       ratio |   1.034571   .0295058    35.06   0.000     .9767403    1.092401
------------------------------------------------------------------------------
```

Figure 7.3: Normal-Based Bootstrapped Confidence Interval

lowest estimate and the upper bound at the 25th highest estimate. None of this requires that we impose a sampling distribution. Rather, we take advantage of the empirically estimated sampling distribution to obtain the confidence interval.

The percentile method is easy to implement, as we shall see in Example 7.6. It does not make any assumptions about the sampling distribution, such as the requirement of symmetry that is embedded in the normal approximation method. The percentile method does not involve limit theorems to derive an asymptotic sampling distribution. Instead, it is based entirely on the data at hand. These are major advantages compared to normal approximations.

There are two drawbacks to the method, however. First, it may not work well if the sample size is too small. The reason is that the percentile method depends heavily on the tails of the sampling distribution and it may be possible to get a good sense of these only with large samples (for a discussion see Mooney and Duval [1993]). Second, the percentile method still leaves the issue of potential bias in the bootstrap estimates. Bias-corrected and bias-corrected and accelerated methods can help to overcome this problem.

Bias-Corrected Confidence Intervals

To understand how bias-corrected (BC) confidence intervals address bias, we should introduce the concept of **median bias**. Median bias, or B, is the probability of obtaining a bootstrap estimate that is less or equal to the maximum likelihood estimate, i.e. $B = \Pr(\hat{\theta}_i \leq \hat{\theta})$. In empirical terms,

$$B = \frac{f(\hat{\theta}_i \leq \hat{\theta})}{k}$$

where f denotes the frequency, i.e. the number of elements of the bootstrap distribution that are less or equal to the estimate. If $B = .5$, then there is no median bias. In this case, half of the bootstrap estimates are underestimates relative to $\hat{\theta}$ but this means that the other half are overestimates. Since the probability of overestimation is just as large as the probability of underestimation, we would conclude that there is no systematic error in the bootstrap estimates, i.e. there is no bias. In general, we would be unconcerned about bias if $.4 \leq B \leq .6$ or, more liberally, $.35 \leq B \leq .65$. For values outside of these regions we would conclude that there is bias.

For purposes of constructing confidence intervals median bias is usually converted into a biasing constant, z_0. The notion here is that the quantity

$\hat{\theta}_i - \hat{\theta}$ is centered about $z_0\sigma$. To extract z_0 it is essential that we impose a distributional assumption. In the BC procedure it is assumed that the aforementioned quantity, or more correctly a transformation thereof, is normally distributed. In this case,

$$z_0 = \Phi^{-1}(B)$$

where Φ^{-1} is the inverse of the cumulative standard normal distribution function. Note that $z_0 = 0$ if the median bias is .5.

Once the biasing constant has been found, then the BC confidence interval is found by computing two statistics:

$$\begin{aligned} p_1 &= \Phi\left\{2z_0 - z_{1-\alpha/2}\right\} \\ p_2 &= \Phi\left\{2z_0 + z_{1-\alpha/2}\right\} \end{aligned} \tag{7.4}$$

The lower-bound of the BCconfidence interval is given by the value of $\hat{\theta}_i$ corresponding to the $100 \times p_1$th percentile, while the upper-bound is given by the value of $\hat{\theta}_i$ corresponding to the $100 \times p_2$th percentile.

Example 7.6. To compute bootstrapped confidence intervals using the percentile and bias corrected procedures in Stata, one adds the following command after running the `bootstrap` command:

```
estat bootstrap, p bc
```

Figure 7.4 shows the resulting output.

Bias-Corrected and Accelerated Confidence Intervals

An even better approach to computing bootstrap intervals is to use the bias-corrected and accelerated (BC_α) method. This approach addresses any skewness issues in the bootstrap sampling distribution and takes into consideration the rate of change in the standard error of that distribution. Key to this method is to compute an acceleration constant

$$a = \frac{\sum_{i=1}^{n}\left(\bar{\theta}_{(.)} - \hat{\theta}_{(i)}\right)^2}{6\left\{\sum_{i=1}^{n}\left(\bar{\theta}_{(.)} - \hat{\theta}_{(i)}\right)^2\right\}^{3/2}}$$

```
. estat bootstrap, p bc
Bootstrap results                                 Number of obs      =      1191
                                                  Replications       =      1000

      command:  myratio bush kerry
        ratio:  r(ratio)

------------------------------------------------------------------------------
             |   Observed               Bootstrap
             |      Coef.       Bias    Std. Err.  [95% Conf. Interval]
-------------+----------------------------------------------------------------
       ratio |   1.0345707   .0011362   .02950583    .9819977   1.095615   (P)
             |                                       .9821262   1.095651   (BC)
------------------------------------------------------------------------------
(P)    percentile confidence interval
(BC)   bias-corrected confidence interval
```

Figure 7.4: Percentile and BC Bootstrapped Confidence Interval

Here $\hat{\theta}_{(i)}$ is the deletion or jackknife estimate of θ that comes about by dropping the ith observation; $\bar{\theta}_{(.)}$ is the mean of these jackknife estimates.

Once a is known, the formulas for p_1 and p_2 described for the BC procedure are adjusted as follows:

$$\begin{aligned} p_1 &= \Phi\left\{z_0 + \frac{z_0 - z_{1-\alpha/2}}{1 - a(z_0 - z_{1-\alpha/2})}\right\} \\ p_2 &= \Phi\left\{z_0 + \frac{z_0 + z_{1-\alpha/2}}{1 - a(z_0 + z_{1-\alpha/2})}\right\} \end{aligned} \tag{7.5}$$

Notice that this reduces to the BC formulas for p_1 and p_2 when $a = 0$. The lower-bound of the BC_α confidence interval is given by the value of $\hat{\theta}_i$ corresponding to the $100 \times p_1$th percentile, while the upper-bound is given by the value of $\hat{\theta}_i$ corresponding to the $100 \times p_2$th percentile.

Example 7.7. To compute the bias-corrected and accelerated bootstrap confidence intervals in Stata the following commands will need to be used:

```
set seed 1

bootstrap ratio=r(ratio), reps(1000) bca:  myratio bush kerry

estat bootstrap, bca
```

```
. estat bootstrap, p bc
Bootstrap results                               Number of obs      =      1191
                                                Replications       =      1000

      command:  myratio bush kerry
        ratio:  r(ratio)
```

	Observed Coef.	Bias	Bootstrap Std. Err.	[95% Conf.	Interval]	
ratio	1.0345707	.0011362	.02950583	.9819977	1.095615	(P)
				.9821262	1.095651	(BC)

(P) percentile confidence interval
(BC) bias-corrected confidence interval

Figure 7.5: BC_α Bootstrapped Confidence Interval

Figure 7.5 shows the resulting output. Comparing all of the relevant intervals, we see that there are overall few differences.

Chapter 8

Model Fit

Researchers frequently want to know if their model fits the data. How well does it account for the data? How does it compare to rival models? In this chapter, we discuss a range of so-called pseudo-R^2 measures that have been developed to assess model fit. We also discuss so-called entropy-based measures, the best-known of which are the AIC and BIC. The latter two measures provide useful tools for model comparison and can be used both with nested (see Definition 6.1) and non-nested models.

8.1 Pseudo-R^2 Measures

The popularity of the coefficient of determination or, R^2, in linear regression analysis has caused statisticians to define similar measures in the MLE context. These pseudo-R^2 statistics mimic the R^2 but, as their name suggests, they do not exactly behave like the coefficient of determination and their interpretation is not in terms of explained variance like R^2. Here, we focus on general pseudo-R^2 statistics, i.e. those that can be computed for any model. Depending on the application, there may be other statistics as well, but we postpone a discussion of those more specialized measures until later chapters when they become relevant.

It should be noted that all of the pseudo-R^2 measures discussed here follow a similar logic. Somehow they contrast the fit of a substantive model with that of an empty model. Here the substantive model contains the covariates that are of interest, while the empty model contains only a set of one or more constants. This is akin to how R^2 is defined in the context of linear

regression analysis.

8.1.1 McFadden's R^2

McFadden's (1973) R^2, also known as the likelihood ratio index, is one of the most widely used pseudo-R^2 statistics. It is what Stata computes by default for most procedures that are based on MLE. The measure evaluates the ratio of the log-likelihood functions for the substantive model (ℓ_1) and the empty model (ℓ_0). The log-likelihood of the empty model can be seen as a general equivalent of the total sum of squares, while the log-likelihood of the substantive model is a general equivalent of the residual sum of squares. In light of these interpretations, McFadden's R^2 follows directly from the formula for the coefficient of determination in regression analysis:[1]

$$R^2_{McF} = 1 - \frac{\ell_1}{\ell_0} \tag{8.1}$$

where l_1 is the log-likelihood of the substantive model and l_0 is the log-likelihood of the empty model. R^2_{McF} resembles the regression R^2 most closely in the discrete case; in this case, $0 \leq R^2_{McF} \leq 1$.[2] In the continuous case, we may find $R^2_{McF} > 1$, which shows that pseudo-R^2 statistics do not quite behave like the regression R^2.[3] McFadden (1979) has proposed that $.2 \leq R^2_{McF} \leq .4$ signifies an excellent fit of the model.

[1] One way to derive this formula is to define ℓ_{max} as the maximum attainable log-likelihood. We can then decompose the quantity $\ell_{max} - \ell_0$ as $(\ell_{max} - \ell_1) + (\ell_1 - \ell_0)$, where $\ell_1 - \ell_0$ is the explained portion and $\ell_{max} - \ell_1$ is the unexplained portion. Taking the explained portion over the total portion, $\ell_{max} - \ell_0$, we get $(\ell_1 - \ell_0)/(\ell_{max} - \ell_0)$. If we assume that the maximum feasible value of the log-likelihood is 0, which is true in the discrete case, then this expression reduces to R^2_{McF}.

[2] It is easy to demonstrate this. We can think of the substantive model as the unconstrained model and of the empty model as the constrained model. From the discussion of the LR test we know that $\mathcal{L}_1 \geq \mathcal{L}_0$. In the discrete case, we also know that the likelihood function is a probability. Hence, $0 < \mathcal{L}_1 \leq 1$ and $0 < \mathcal{L}_0 \leq 1$. Taking the natural logarithms yields $\ell_1 \leq 0$ and $\ell_0 \leq 0$. Further, since $\mathcal{L}_1 \geq \mathcal{L}_0$, it follows that $\ell_1 \leq \ell_0$. Hence, $\ell_1/\ell_0 \leq 1$ and $0 \leq R^2_{McF} \leq 1$. Notice that the lower bound arises when $\ell_0 = \ell_1$. Also notice that in practice, the upper bound will usually not be reached since this requires $\mathcal{L}_1 = 1$, which would only happen if we have a saturated model (with one parameter for each observation).

[3] This happens because $\mathcal{L}_1$ and $\mathcal{L}_0$ are now based on densities, which can be greater than 1. Thus, if $\mathcal{L}_1 > 1$ and $0 < \mathcal{L}_0 < 1$, then $\ell_1/\ell_0 < 0$ and $R^2_{McF} > 1$.

Ben-Akiva and Lerman (1985) propose an adjusted version of R^2_{McF} that includes a penalty factor for including unnecessary parameters, much like the adjusted R^2 in linear regression analysis:

$$\bar{R}^2_{McF} = 1 - \frac{\ell_1 - p}{\ell_0} \tag{8.2}$$

where p is the number of estimated parameters in the model. This measure will increase only if ℓ_1 increases by more than one for every additional parameter in the model.

8.1.2 LR-Based R^2 Statistics

An alternative form of the pseudo-R^2 is based on the likelihood ratio:

$$R^2_{LR} = 1 - \lambda^{\frac{2}{n}} \tag{8.3}$$

(where $\lambda = \mathcal{L}_0/\mathcal{L}_1$—see Chapter 6.1). This statistic can also be formulated in terms of the LR test statistic: $R^2_{LR} = 1 - \exp(-LR/n)$ and is alternatively referred to as the Cox-Snell pseudo-R^2 and Maddala's pseudo-R^2 (Maddala 1983). If a set of predictors does not add to the model fit, then $\lambda = 1$ and $LR = 0$, which means that $R^2_{LR} = 0$. The theoretical upper-bound of R^2_{LR} is $1 - \mathcal{L}_0^{.5n}$.

Aldrich and Nelson (1985) proposed another transformation of the LR test statistic:

$$R^2_{A\&N} = \frac{LR}{LR + n} \tag{8.4}$$

Notice that both of these R^2 statistics include sample size information, although they do so in different ways.

Several other variants of LR-based pseudo-R^2 measures have been proposed. Cragg and Uhler (1970) proposed a measure similar to (8.3) that takes into consideration the maximum possible value of the log-likelihood function; this measure has a theoretical range between 0 and 1. Veall and Zimmermann (1990, 1992) proposed variants on (8.4), which will be discussed in greater detail in Chapters 9 and 10.

8.1.3 Other Test-Based R^2 Statistics

Magee (1990) proposed a pseudo-R^2 statistic that is based on the Wald-test but that looks a lot like the statistic proposed by Aldrich and Nelson (1985):

$$R^2_W = \frac{W}{W+n} \tag{8.5}$$

This statistic is useful because it does not require estimation of the restricted model, although in many cases the log-likelihood for the empty model is easy to determine.

Magee (1990) also proposed a pseudo-R^2 that is based on the LM-test statistic:

$$R^2_{LM} = \frac{LM}{n} \tag{8.6}$$

This statistic is based on the notion that if the empty model is reasonable, $LM = 0$ and $R^2_{LM} = 0$.

8.2 Entropy and Model Evaluation

The concept of entropy refers to uncertainty in the data in terms of supporting a particular model rather than another. It is the opposite of information (or negentropy)—the more information there is in the data, the smaller entropy is. While several entropy functions can be defined, we focus on those most closely related to the ideas of Ludwig Boltzmann (1844-1906) in physics.

8.2.1 Kullback-Leibler Information

Assume that $f(x|\theta^*)$ is the true density of a random variable X; it is the true model. We estimate the model $g(x|\theta)$. An important question is how close this model is to the truth. Kullback and Leibler (1951) defined this closeness in terms of the following information measure:

$$\begin{aligned} I\left[f(x|\theta^*), g(x|\theta)\right] &= E\left[\ln\left(\frac{f(x|\theta^*)}{g(x|\theta)}\right)\right] \\ &= E\left[\ln f(x|\theta^*) - \ln g(x|\theta)\right] \\ &= E\left[\ln f(x|\theta^*)\right] - E\left[\ln g(x|\theta)\right] \\ &= \int \ln f(x|\theta^*) f(x|\theta^*) dx - \int \ln g(x|\theta) f(x|\theta^*) dx \end{aligned}$$

The first term is a measure of entropy and the second term is a measure of cross-entropy.[4] The smaller the K-L information measure is, the closer the model is to the true distribution. When we have competing models, we should prefer the model that has the smallest K-L information value.

Example 8.1. Consider the family of normal distributions, $X \sim N(\mu, \sigma^2)$. Further, let $N(\mu^*, \sigma^{*2})$ be the true normal distribution. Then the K-L information is defined as

$$I = .5 \ln \left(\frac{\sigma^2}{\sigma^{*2}} \right) + .5 \left[\frac{\sigma^{*2}}{\sigma^2} - 1 + \frac{(\mu^* - \mu)^2}{\sigma^2} \right]$$

Suppose that $\mu^* = 0$ and $\sigma^{*2} = 1$, i.e. the true distribution is the standard normal distribution. Then

$$I = .5 \ln \sigma^2 + .5 \left[\frac{1}{\sigma^2} - 1 + \frac{\mu^2}{\sigma^2} \right]$$

Imagine that we have two competing models: $1 : X \sim N(.5, 1)$ and $2 : X \sim N(0, 2)$. Which of these models is closest to the truth? After some simple computations we find that $I_1 = .125$ and $I_2 = .097$. Since $I_2 < I_1$, it follows that the second model is closer to the truth.

8.2.2 Akaike's Information Criterion

The problem with K-L information is that we can compute it only if we know the true model, which is never the case. Akaike (1973) developed a large sample estimate of expected entropy or expected K-L information that does not require any knowledge about the true model and is simple to compute. Specifically, Akaike's Information Criterion is defined as

$$AIC_i = -2\ell_i + 2p_i \tag{8.7}$$

where i denotes a particular model, ℓ_i is the log-likelihood for this model, and p_i is the number of estimated parameters.[5] The log-likelihood component of AIC considers model fit, while the parameter component considers

[4] These terms originate from applying the definition of an expected value. If $f(x)$ is the distribution of X, then by definition $E[X] = \int x f(x) dx$. In this case, we should substitute $\ln f(x|\theta^*)$ or $\ln g(x|\theta)$ for X, and $f(x|\theta^*)$ for $f(x)$.

[5] Some authors define AIC by dividing the right-hand side of (8.7) by n (see Long and Freese 2003).

parsimony. The model with the smallest value of AIC is the preferred model. It should be emphasized that AIC can be used to compare both nested and non-nested models; as such it is quite versatile.

The AIC is based on asymptotic theory. In small samples, it is typically recommended to compute a version of AIC that contains a finite sample correction (see Hurvich and Tsai 1989):

$$AIC_i^c = -2\ell_i + 2p_i + \frac{2p_i(p_i+1)}{n-p_i-1} \tag{8.8}$$

This version should be used if $n/p_i < 40$. Notice that (8.8) reduces to (8.7) when n goes to infinity (and assuming p_i is finite).

For models with over-dispersion, such as certain event count models, yet a third variant of Akaike's information criterion has been proposed, namely the quasi-AIC:

$$QAIC_i = -2\frac{\ell_i}{c} + 2p_i \tag{8.9}$$

where c is a variance inflation factor (see Burnham and Anderson 1998). This measure is based on quasi-likelihood theory.[6]

> **Example 8.2.** Imagine that there are three classical normal linear regression models that we seek to estimate: $A1 : y_i = \beta_0 + \beta_1 x1_i + \beta_2 x2_i + \beta_3 x3_i + \epsilon_i$, $A2 : y_i = \beta_0 + \beta_1 x1_i + \beta_2 x2_i + \beta_3 x3_i + \epsilon_i$, with $\sigma^2 = 1$, and $B : y_i = \beta_0 + \epsilon_i$. Notice that $A1$ has 5 estimated parameters (β_0, β_1, β_2, β_3, and σ^2), while $A2$ has four estimated parameters (σ^2 is fixed) and B has two estimated parameters (β_1, β_2, and β_3 are fixed). Table 8.1 shows the log-likelihood values as well as the AIC scores. We see that $A2$ has the smallest AIC-value, so this is the preferred model.

Our discussion of AIC so far suggests that we simply use it to select one best model. But as example 8.2 shows, this can lead to strange decisions. The AIC values of models $A1$ and $A2$ are very close, yet we would completely

[6] Quasi-likelihood estimation focuses on estimation problems where the precise distribution of the data is not known. All that is known is the relationship between the mean of the distribution and a set of covariates (the link function) and the relationship between the variance and the mean of the distribution (the variance function). The quasi-likelihood function captures those relationships and allows one to obtain parameter estimates.

Table 8.1: AIC for three Regression Models

Model	k	l	AIC	ΔAIC	$L(AIC)$	$w(AIC)$
A1	5	-1426.577	2863.154	1.884	.390	.281
A2	4	-1426.635	2861.270	0.000	1.000	.719
B	2	-2635.705	5275.410	2414.140	.000	.000

ignore $A1$. When AIC differences are small, accepting a single model can lead to a false sense of confidence. The AIC has another drawback as well: raw AIC scores cannot be used to compare the weight of the evidence of models $A1$ and $A2$ versus B.

To overcome these limitations Akaike suggested transforming AIC into a **Akaike weight**. This process starts by computing differences in AIC with respect to the best model:

$$\Delta_i(AIC) \quad = \quad AIC_i - \min AIC$$

Next, the relative likelihood of a model is computed using

$$\mathcal{L}(M_i|D) \propto \exp[-.5\Delta_i(AIC)]$$

(In this step we take advantage of the fact that the AIC is an unbiased estimator of -2 times the log-likelihood of a model.) Finally, the relative likelihoods are normalized to obtain Akaike weights

$$w_i(AIC) \quad = \quad \frac{\exp[-.5\Delta_i(AIC)]}{\sum_k \exp[-.5\Delta_k(AIC)]} \tag{8.10}$$

where $\sum_i w_i(AIC) = 1$. The Akaike weights can be interpreted as the probability that a model, M_i, is the best model (in the sense of minimizing the expected K-L information). As the example below shows, Akaike weights can also be used to evaluate pairs or groups of models.

> **Example 8.3.** If we want to compute the Akaike weights for the three regression models described in Example 8.2, then we proceed as shown in the last three columns of Table 8.1. This singles model $A2$ out as the best model. This model is $.719/.281 \approx 2.6$ times more likely than the next best fitting model, $A1$. The normalized probability

that model $A2$ is to be preferred over $A1$ is $.719/(.719 + .281) = .719$. Models in class A (those including covariates) are infinitely more likely than models in class B (those ignoring covariate information): $(.719 + .281)/.000 \rightarrow \infty$.

8.2.3 Bayesian Information Criterion

The Bayesian Information Criterion (BIC), also known as Schwarz's criterion, is an alternative information-based measure for model evaluation, which originates from the literature on Bayesian inference (see Schwarz 1978).[7] Specifically, BIC is an asymptotic approximation to a Bayesian model selection (BMS) analysis, which does not require the specification of a prior distribution for the parameters. It is computed as[8]

$$BIC_i \quad = \quad -2\ell_i + p_i \ln n \tag{8.11}$$

Notice that the BIC penalty term (the second term) is larger than the AIC penalty term if $n > e^2$, i.e. in samples greater than approximately 7. As a practical matter, this means that the BIC penalty term is always greater than the AIC penalty term, since sample sizes below 7 are extremely rare. We can transform the BIC into so-called Schwartz weights by performing the same operations that we performed to obtain AIC weights.

Example 8.4. Table 8.2 shows the BIC values for the three regression models discussed in Examples 8.2-8.3. Compared to Table 8.1, we see that the BIC favors model $A2$ even more strongly: this model is now 29 times more likely than the next most plausible model, $A1$. The reason for this difference between the BIC and AIC is that the BIC puts a greater penalty on the extra parameter that is included in $A1$.

[7] Compared to AIC, the BIC has two main advantages. First, it is not as liberal as AIC, which means that it is less likely to select overly complex models (the reason being that it takes the sampling variability of parameter estimates into account). Second, BIC is consistent whereas AIC is not. Thus, as n approaches infinity, the probability that the BIC recovers a true model approaches unity.

[8] In some cases, BIC is defined in terms of the LR chi-squared test statistic and is computed as $-LR + K \ln n$, where K is the number of covariates in the substantive model instead of the total number of parameters (for details see Long and Freese (2003).

Table 8.2: BIC for three Regression Models

Model	k	l	BIC	ΔBIC	$L(BIC)$	$w(BIC)$
A1	5	-1426.577	2887.693	6.792	.034	.033
A2	4	-1426.635	2880.901	0.000	1.000	.967
B	2	-2635.705	5285.226	2404.325	.000	.000

8.3 Assessing Model Fit in Stata

Stata provides a number of commands that help to assess model fit. By default, many applications report McFadden's pseudo-R^2. Running the `stats` option on the `lrtest` command will produce estimates of the AIC and BIC criteria, as was discussed in Chapter 6.1.5.

Long and Freese (2003) have developed the `fitstat` command as a part of their SPost Stata add-on. This command can be issued after running any of the following procedures: `clogit`, `cloglog`, `cnreg`, `gologit`, `intreg`, `logistic`, `logit`, `mlogit`, `nbreg`, `ocratio`, `ologit`, `oprobit`, `poisson`, `probit`, `regress`, `tobit`, `zinb`, and `zip`. These commands will be discussed in Part II of this report. The `fitstat` program computes R^2_{McF}, $\bar{R}^2_{McF}$, R^2_{LR}, Cragg and Uhler's R^2, AIC (which is equal to equation [8.7] divided by n), `AIC*n` (which is equal to equation [8.7]), BIC, `BIC'` (a LR chi-squared version of BIC), as well as several other useful statistics.[9]

[9]In regression analysis, all of the pseudo-R^2 measures reduce to the regular R^2. As such, `fitstat` reports the normal regression R^2 and adjusted R^2.

Part II

Applications of Maximum Likelihood

Chapter 9

Models for Binary Outcomes

Binary response variables are extremely common in political science. Students of voting behavior want to know if a citizen participated in an election or stayed home. Experts on legislatures are interested in modeling whether a representative voted yeah or nay on a bill. In comparative politics, scholars may want to know if a government fell prematurely or not. In international relations, students of inter-state conflict are interested in whether two countries went to war against each other or stayed at peace.

Contrary to what your introductory econometrics text may have told you, it is possible to analyze these kinds of binary responses using linear regression analysis. The resulting model is known as the linear probability model (LPM). As we shall see, however, this model has a number of drawbacks, which have caused econometricians and statisticians to look for alternatives. The logit and probit models are two particularly well-known alternatives, but there are others. In this chapter, I discuss how one can derive models for binary response variables, estimate them via MLE, and interpret the results.

9.1 The Linear Probability Model

9.1.1 Derivation and Estimation

The LPM and other models for binary responses can be motivated in a number of different ways. Here I focus on decision theoretic rationales, which

are common in econometrics.[1] Consider an individual i who faces a choice between two alternatives, θ or $\bar{\theta}$ (i.e. not θ). For example, θ could be voting in an election, a yeah vote for a bill, or going to war against another country.[2] We indicate the individual's choice with a binary indicator

$$y_i = \begin{cases} 0 & \text{if } \theta \text{ is not chosen} \\ 1 & \text{if } \theta \text{ is chosen} \end{cases}$$

Imagine that we ignored the admonition that the linear regression model should be used only for continuous response variables. In this case, we could model the binary response variable via the population regression function

$$y_i = \boldsymbol{x}_i\boldsymbol{\beta} + \epsilon_i$$

where $\boldsymbol{x}_i$ is a set of attributes of the decision maker. Under the standard assumption that $\mathcal{E}[\epsilon_i] = 0$, we know that

$$\mathcal{E}[y_i] = \boldsymbol{x}_i\boldsymbol{\beta}$$

Since y_i is binary, however, we also know that[3]

$$\begin{aligned} \mathcal{E}[y_i] &= \Pr[y_i = 1] \\ &= \pi_i \end{aligned}$$

Substituting this result into the regression model, we get

$$\pi_i = \boldsymbol{x}_i\boldsymbol{\beta} \tag{9.1}$$

Here π_i is the probability that i chooses θ (and $1 - \pi$ is the probability that he/she does not choose θ). In equation (9.1), this is a linear function of the covariates in $\boldsymbol{x}_i$. Thus, the model in (9.1) is known as the **linear probability model**.

[1] In biostatistics, the motivation for these models is quite different. One common rationale in this field is the dose-response rationale, which considers the probability of a particular response (e.g. being cured from an illness or death) from a specified dose of a medication or pathogen. Since binary response variables arise mostly in the context of choice in political science, the econometric rationale seems more helpful.

[2] The current usage of θ should not be confused with the usage in Part I, where θ denoted a parameter.

[3] A 0-1 binary variable y follows the Bernoulli distribution, $f(y) = \pi^y(1-\pi)^{1-y}$, where π is the probability of success (i.e. choosing θ). The expectation of this distribution is π.

The LPM has a number of advantages. First, it can be estimated using least squares estimation, a method that is known to have desirable small sample properties. By contrast, the logit and probit alternatives that will be discussed in the next section can be estimated only through MLE. As we have seen in Chapter 4, desirable properties for ML estimators are guaranteed only asymptotically. Second, since the LPM is a linear model, interpretation is straightforward. Let β_k be the coefficient associated with a predictor x_{ik}. In the LPM, the interpretation of this coefficient is literally that, for a unit increase in x_{ik}, π_i is expected to increase by β_k units. Since logit and probit are inherently non-linear models, their interpretation is considerably more complex.

Despite these advantages, the LPM is only used infrequently these days. The reason is that the model has a number of well-known drawbacks. In increasing order of severity, these drawbacks include: (1) the error distribution is not normal; (2) the LPM has built-in heteroskedasticity; (3) the model produces out-of-bounds predictions; and (4) the linear functional form is frequently invalid. Let me address each of these concerns in turn.

First, since y_i follows the Bernoulli distribution, it follows immediately that ϵ_i is not normally distributed. Instead, ϵ_i also follows a Bernoulli distribution: it takes on the value of $1 - \boldsymbol{x}_i\boldsymbol{\beta}$ with probability π_i and the value $-\boldsymbol{x}_i\boldsymbol{\beta}$ with probability $1-\pi_i$.[4] The reason why we would like the disturbances to be normally distributed is that it follows immediately that $\hat{\beta}_k$ follows a normal distribution (see e.g. Gujarati 2003). We take advantage of this fact for purposes of hypothesis testing. The non-normality of ϵ_i, however, is usually no great concern. If the sample size is sufficiently large, then we can rely on the central limit theorem to prove that the sampling distribution of $\hat{\beta}_k$ is asymptotically normal. Hypothesis testing can then proceed as usual. Thus, non-normality becomes a concern only in small samples.

Second, the LPM has built-in heteroskedasticity. Specifically, we can demonstrate that:[5]

$$V[\epsilon_i] \quad = \quad \boldsymbol{x}_i\boldsymbol{\beta}(1 - \boldsymbol{x}_i\boldsymbol{\beta})$$

We see that the variance of ϵ_i is a function of the predictors, which is a

[4]This follows immediately from the mathematical definition of the disturbance term, namely $\epsilon_i = y_i - \boldsymbol{x}_i\boldsymbol{\beta}$. Substituting the different values of y_i then yields the two distinct values that the disturbance can take on in the LPM.

[5]Proof: The variance of a Bernoulli-distributed variable is $\pi_i(1 - \pi_i)$. In the LPM, $\pi_i = \boldsymbol{x}_i\boldsymbol{\beta}$. Substitution of this result produces the desired equation.

classical form of heteroskedasticity.

The presence of heteroskedasticity in the LPM makes it an unwise choice to estimate this model using OLS. However, this is one of the rare instances where the nature of the heteroskedasticity is known precisely so that we can use weighted least squares (WLS) without any difficulty. Goldberger (1964) outlined a two-step procedure to deal with the problem.

1. First, estimate the LPM using OLS. Obtain the predicted values $\hat{y}_i = \boldsymbol{x}_i\hat{\boldsymbol{\beta}}$. Then compute weights

$$w_i \;=\; \hat{y}_i(1-\hat{y}_i)$$

2. Next, perform WLS using $\sqrt{w_i}$ as the weight variable. This is tantamount to performing OLS on

$$\frac{y_i}{\sqrt{w_i}} \;=\; \frac{1}{\sqrt{w_i}}\boldsymbol{x}_i\boldsymbol{\beta} + \frac{\epsilon_i}{\sqrt{w_i}}$$

   Notice that the new error term $\delta_i = w_i^{-.5}\epsilon_i$ is homoskedastic: $V[\delta_i] = (w_i^{-.5})^2V[\epsilon_i] = w_i^{-1}V[\epsilon_i] = [\boldsymbol{x}_i\boldsymbol{\beta}(1-\boldsymbol{x}_i\boldsymbol{\beta})]^{-1}\boldsymbol{x}_i\boldsymbol{\beta}(1-\boldsymbol{x}_i\boldsymbol{\beta}) = 1$.

Since we can perform WLS, the built-in heteroskedasticity in the LPM is but a minor nuisance. That is to say, it is a minor nuisance as long as the weights can be calculated, which requires that there can be no out-of-bounds predictions.

Since the LPM essentially predicts a probability, namely π_i, it has to be the case that $0 \leq \boldsymbol{x}_i\hat{\boldsymbol{\beta}} \leq 1$. Instances where $\boldsymbol{x}_i\hat{\boldsymbol{\beta}} < 0$ or $\boldsymbol{x}_i\hat{\boldsymbol{\beta}} > 1$ make no sense conceptually. We say that these are out-of-bounds predictions. Unfortunately, there is no guarantee that the least squares estimators will produce predictions within the probability bounds. The least squares estimation procedures treat y_i just as any response variable, namely something that is theoretically bounded between negative and positive infinity. As such, there is a more than negligible chance to find that $\hat{y}_i$ takes on values less than zero or greater than one.

The problem here is not just a conceptual one. With out-of-bounds predictions, WLS will also fail. The reason is that $w_i < 0$ whenever $\hat{y}_i < 0$ or $\hat{y}_i > 1$, and the square root of a negative number is undefined. Indeed, WLS already becomes problematic when the predictions are right at the bounds, since $w_i = 0$ in these instances and dividing by the square root of zero leads to undefined results.

What to do in this case? For purposes of WLS, Achen (1986) proposed replacing the out of bounds predictions with ones that are just inside the bounds. However, this seems highly arbitrary. Only slightly less arbitrary is to prevent out-of-bounds predictions in the first place by running a *constrained LPM* model. Here, the coefficients are estimated in such a way that all predicted values lie between 0 and 1, in recognition of the fact that we are predicting a probability. Unfortunately, the restricted LPM model is frequently beset with multicollinearity problems. It is not difficult to see why: once one has chosen one parameter estimate, the other estimates often follow from it and the restrictions that are being imposed. It is also common for this model to produce negative estimates of the variance. Thus, on the problem of out-of-bounds predictions there are really no good solutions.

Finally, the linear functional form of the LPM is in many cases questionable. This functional form implies that π_i changes at a constant rate, regardless of the starting point on a predictor. A simple example can illustrate why this implication is often implausible. Consider a consumer deciding whether to purchase an article that costs \$2. Among the factors that influence the decision is wealth. Imagine a person whose current wealth is \$0. If we increase this person's wealth by one unit, for example by giving him/her \$1, then the probability of purchasing the article will not increase by a noticeable amount because the person is still short \$1. Next consider a millionaire. If that person has not yet acquired the article it is surely not because of financial reasons. Hence, increasing this person's wealth by one unit should also not have much of an effect on acquisition of the article. Finally, consider someone whose current wealth is \$1. Here a unit increase in wealth could have a tremendous effect because the individual goes from being unable to afford the article to being able to purchase it. The point here is that changes in the probability are not constant across the predictor, as the LPM assumes. On the contrary, these changes are larger or smaller depending on what the initial value of the predictor is.

Indeed, the example suggests that an S-shaped functional form may be more reasonable. Here, initial increases in a covariate have relatively little impact on π_i (e.g. giving someone with nothing one dollar). However, as one moves along the scale of a covariate, the effect of unit increases become ever larger, at least up to a certain point (e.g. giving a person with one dollar another dollar so that they can acquire an article). Then, the changes become smaller again (e.g. giving a dollar to a millionaire). The LPM is ill-suited to accommodate these S-shaped changes in the effects of covariates

on π_i. This is one of the most important reasons why this model has been replaced by other models, most notably logit and probit.

9.1.2 Example

I illustrate the LPM model with an analysis of vote choice in the 2000 U.S. Presidential elections. The response variable is whether a respondent voted for Al Gore ($y = 1$) or George W. Bush ($y = 0$).[6] The covariates include age, gender (1 = male), race (1 = white), religion (1 = born again Christian), education (`educ`), household income (`hhinc`), partisanship (`pid`), a trait differential (`traitdif`), and an issue differential (`issuedif`). The trait differential is measured as the difference between trait evaluations of Gore and Bush. Higher values mean that the trait evaluations of Gore are more positive than those of Bush. The issue differential is measured as the difference in the issue distances between the respondent and Bush and Gore, respectively. Higher values mean that the respondent is further away on the issues from Bush than from Gore.

We begin by computing the OLS estimates of the LPM of vote choice. These estimates are obtained by running the following command:

```
regress vote pid male white bornagain age educ hhinc ///
traitdif issuedif
```

The resulting estimates are shown in the columns marked OLS in Table 9.1 These estimates are straightforward to interpret. For example, the coefficient on partisanship suggests that for a unit increase, which means moving from the Democratic to the Republican end of the party identification scale, the probability of voting for Gore decreases, on the average, by .1 points.

The OLS results are marred by the problem of built-in heteroskedasticity that plagues the LPM. Such heteroskedasticity means that the standard errors for the coefficients cannot be trusted, so that claims about the significance (or lack of significance) of predictors are suspect. To address this issue, we can implement Goldberger's (1964) method. Tamás Bartus has written the `linprob` package that accomplishes this task in Stata. The generic syntax for this package is as follows:

[6]Non-voters or respondents who voted for a candidate from a different party have been dropped from the sample.

Table 9.1: LPM of Vote Choice in 2000

	OLS		WLS	
Predictor	**Estimate**	**S.E.**	**Estimate**	**S.E.**
Partisanship	$-.100^{**}$	.015	$-.095^{**}$	.012
Male	$-.016$	.046	.022	.040
White	$-.110$	.082	$-.148^{*}$	.069
Born Again Christian	$-.141^{**}$	.049	$-.168^{**}$	.039
Age	$-.003^{+}$	.002	$-.004^{**}$	.001
Education	$-.008$	.017	$-.010$	.011
Household Income	$-.012^{+}$	.006	$-.019^{**}$	.005
Trait Differential	$.195^{**}$	.041	$.281^{**}$	.037
Issue Differential	.005	.042	.001	.033
Constant	1.216^{**}	.153	1.340^{**}	.124

Notes: $n = 142$. $^{**}p < .01$, $^{*}p < .05$ (two-tailed). 61 observations produced out-of-range predictions.

`linprob` *varlist*, [`gen(`*varname*`)`]

Here, the option `gen(varname)` generates an indicator variable that flags all out of bound predictions. The WLS results can be found in the last two columns of Table 9.1. We see that, relative to the OLS results, few things have changed. In the WLS solution, race is now significant and the significance levels of age and household income have improved. Other than this, the results are comparable. Note, however, the very large number of observations (61) that produced out-of-bound predictions. This sheds considerable doubt on the appropriateness of the LPM for this data.

9.2 Logit and Probit Analysis

As the discussion of the LPM makes clear, the ideal mapping of predictors to probabilities is one that avoids heteroskedasticity and out-of-bounds predictions while being accommodating to a sigmoid shaped relationship between $\boldsymbol{x}_i$ and π_i. Two mappings that comply with these requirements are

$$\pi_i \quad = \quad \Phi\left(\boldsymbol{x}_i\boldsymbol{\beta}\right) \tag{9.2}$$

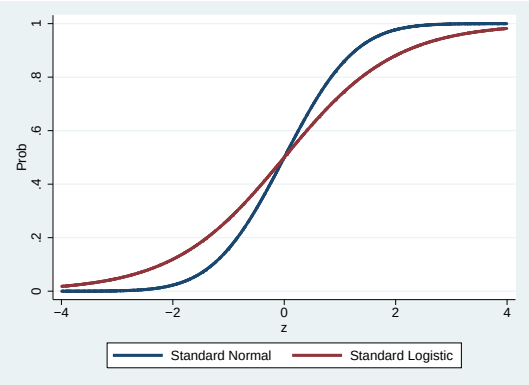

Figure 9.1: The Standard Logistic and Standard Normal CDFs

and

$$\begin{aligned} \pi_i &= \Lambda\left(\boldsymbol{x}_i\boldsymbol{\beta}\right) \\ &= \frac{\exp\left(\boldsymbol{x}_i\boldsymbol{\beta}\right)}{1+\exp\left(\boldsymbol{x}_i\boldsymbol{\beta}\right)} \end{aligned} \tag{9.3}$$

where $\Phi(.)$ denotes the standard normal CDF and $\Lambda(.)$ the standard logistic CDF. Equation (9.2) describes the well-known **probit model**, while equation (9.3) describes the equally well-known **logit model**.[7]

The probit and logit specifications are ideal for a number of reasons. First, both specifications are, by definition, homoskedastic. The variance of the probit error terms is fixed at 1, while the logit error term has a variance of $\pi^2/3$, where π is the mathematical constant pi. Second, both specifications avoid out-of-bounds predictions because they are based on CDFs, which are bounded between 0 and 1.[8] Finally, as Figure 9.1 illustrates, both specifications allow for an S-shaped relationship between the predictors and π_i. For

[7]The logit model has been credited to the American physicist and statistician Joseph Berkson who developed it in 1944. Because the logit model is based on the logistic distribution, it is also known as the logistic model. However, Berkson (1944) preferred the term logit due to its similarity to the term probit, which had been coined by the entomologist Chester Bliss (1934) ten years earlier. As we have seen, the term logit refers to the transformation $\ln[\pi/(1-\pi)]$ that we discussed in Chapter 4 and that linearizes the logit model. The probit model has been credited to Bliss, although its roots go back to Gustav Fechner who experimented with the model in the 19th century.

[8]Technically, the values of 0 and 1 are the lower and upper asymptotes, respectively, of the logit/probit CDFs.

these reasons, the logit and probit specifications are a useful and widely used alternative to the LPM.

9.2.1 Derivation

The discussion so far motivates logit and probit modeling in terms of practical advantages (homoskedasticity, proper predictions, and an appealing functional form). But what is the theoretical rationale for these models? That is, how can they be derived from a theoretical model of choice behavior? Here, I outline two theoretical frameworks: (1) latent regression models and (2) stochastic utility theory.[9]

Latent Regression Models

One way in which the logit and probit models can be derived is to think of the choice variable, y_i, as a manifestation of an unobserved or **latent variable**, y_i^*, which is continuous. This latent variable can be seen as an infinitely fine-grained measure of the net benefit/pleasure/utility that someone derives out of choosing alternative θ. We do not observe this utility; all we observe is the final choice that it produces. The mechanism linking y_i to y_i^* is:

$$y_i = \begin{cases} 1 & \text{if } y_i^* > 0 \\ 0 & \text{if } y_i^* \leq 0 \end{cases} \tag{9.4}$$

This mechanism means that a person chooses θ, so that $y_i = 1$, only if she derives some positive net utility out of the alternative.[10]

We can now model y_i^* as a function of attributes of the decision maker plus a constant, contained in $\boldsymbol{x}_i$, and a stochastic component (ϵ_i). The stochastic component captures unobserved aspects of the net utility of θ, which in this case is mostly measurement error or attributes of the decision maker that are unobserved (from the perspective of the modeler—see Manski 1997). Thus,

$$y_i^* = \boldsymbol{x}_i\boldsymbol{\beta} + \epsilon_i \tag{9.5}$$

[9]Much of the work on the micro-level foundations of the logit and probit models is due to the economist Daniel McFadden (1973).

[10]A more general formulation is to say that $y_i = 1$ if $y_i^* > k$, where k is some arbitrary constant. The restriction $k = 0$ that we impose here may seem arbitrary, but it is of little practical relevance since the model for y_i^* includes a constant that can accommodate deviations from $k = 0$ if the data suggest so.

Econometricians sometimes refer to $\boldsymbol{x}_i\boldsymbol{\beta}$ as the **index function**. Notice that the index function contains a constant, β_0.

We can now develop the latent regression model a little further. Combining equations (9.4) and (9.5), we have[11]

$$y_i = \begin{cases} 1 & \text{if } \epsilon_i > -\boldsymbol{x}_i\boldsymbol{\beta} \\ 0 & \text{if } \epsilon_i \leq -\boldsymbol{x}_i\boldsymbol{\beta} \end{cases} \tag{9.6}$$

Moreover, the stochastic nature of ϵ_i means that choice is probabilistic. That is, the outcome $y_i = 1$ is stochastic because one of the parts driving it, ϵ_i, is stochastic. Accordingly, choice is probabilistic and the choice probabilities are given by

$$\begin{aligned} \Pr[y_i = 1] &= \pi_i \\ &= \Pr[\epsilon_i > -\boldsymbol{x}_i\boldsymbol{\beta}] \end{aligned} \tag{9.7}$$

Similarly, $\Pr[y_i = 0] = 1 - \pi_i = \Pr[\epsilon_i \leq -\boldsymbol{x}_i\boldsymbol{\beta}]$.

Equation (9.7) is undefined unless we are willing to make some distributional assumptions. In the probit model, we assume that $\epsilon_i \sim N(0, 1)$, i.e. the stochastic component follows a standard normal distribution. Since the standard normal distribution is a symmetric distribution, it follows that $\Pr[\epsilon_i > -\boldsymbol{x}_i\boldsymbol{\beta}] = \Pr[\epsilon_i < \boldsymbol{x}_i\boldsymbol{\beta}] = F(\boldsymbol{x}_i\boldsymbol{\beta})$, where $F(.)$ is the CDF, in this case the standard normal CDF, $\Phi(.)$. Thus,

$$\pi_i = \Phi(\boldsymbol{x}_i\boldsymbol{\beta})$$

which is equation (9.2) that we discussed before. In the logit model, we assume that $\epsilon_i \sim L(0, \pi^2/3)$, i.e. the stochastic component follows a standard logistic distribution. This, too, is a symmetrical distribution so that $\Pr[\epsilon_i > -\boldsymbol{x}_i\boldsymbol{\beta}] = \Pr[\epsilon_i < \boldsymbol{x}_i\boldsymbol{\beta}] = F(\boldsymbol{x}_i\boldsymbol{\beta})$. Here $F(.) = \Lambda(.)$, which is the standard logistic CDF, so that

$$\pi_i = \Lambda(\boldsymbol{x}_i\boldsymbol{\beta})$$

This is equation (9.3) that we discussed before.

The latent regression approach shows that the derivation of the logit and probit models follows a clear logic. Even the distributional assumptions are

[11] *Proof:* From (9.4) we know that $y_i = 1$ if $y_i^* > 0$. Substituting $\boldsymbol{x}_i\boldsymbol{\beta} + \epsilon_i$ for y_i^*, this can be formulated as $y_i = 1$ if $\boldsymbol{x}_i\boldsymbol{\beta} + \epsilon_i > 0$. Subtracting the index function from both sides of the inequality sign, we get $\epsilon_i > -\boldsymbol{x}_i\boldsymbol{\beta}$.

not as arbitrary as they may seem at first glance. Specifically, assuming a symmetrical distribution for ϵ_i with mean 0 makes a great deal of sense if one believes that the stochastic component of the latent variable has no systematic tendency and if positive and negative displacements from $\boldsymbol{x}_i\boldsymbol{\beta}$ are equally likely.[12]

Stochastic Utility Theory

The logit and probit models can be derived also from the **stochastic utility model** (a.k.a. the random utility model), which finds its roots in both economics (e.g. Neuman and Morgenstern) and psychology (e.g. Thurstone).[13] Let y_i indicate the choice between two alternatives, θ and $\bar{\theta}$ such that $y_i = 1$ if θ is chosen and $y_i = 0$ if $\bar{\theta}$ is chosen. Further, let U^{θ} and $U^{\bar{\theta}}$ be the utilities of both alternatives. In the stochastic utility model, these utilities have both fixed and stochastic components:

$$\begin{aligned} U_i^{\theta} &= \boldsymbol{x}_i\boldsymbol{\beta}_{\theta} + \epsilon_{i,\theta} \\ U_i^{\bar{\theta}} &= \boldsymbol{x}_i\boldsymbol{\beta}_{\bar{\theta}} + \epsilon_{i,\bar{\theta}} \end{aligned} \tag{9.8}$$

Here, $\boldsymbol{x}_i$ is a vector of observed attributes of the decision maker, $\boldsymbol{\beta}_{\theta}$ and $\boldsymbol{\beta}_{\bar{\theta}}$ are the effects of those attributes on the utilities of alternatives θ and $\bar{\theta}$, respectively, and $\epsilon_{i,\theta}$ and $\epsilon_{i,\bar{\theta}}$ are random and unobserved components, including attributes of the decision maker that are hidden from the modeler (e.g. Manski 1997).

The stochastic utility model assumes that decision makers are utility maximizers. This means that $y_i = 1$ if $U_i^{\theta} > U_i^{\bar{\theta}}$. Since the utilities contain a

[12] There are other distributional assumptions that one can make, as we shall see in Chapter 9.3. Whether those assumptions are more attractive depends on the nature of the problem that one is analyzing.

[13] Also see Nakosteen and Zimmer (1980) for a derivation of logit and probit analysis from stochastic utility theory.

stochastic component, choice is again probabilistic. Thus,

$$\begin{aligned}
\Pr[y_i = 1] &= \pi_i \\
&= \Pr[U_i^{\theta} > U_i^{\bar{\theta}}] \\
&= \Pr[\boldsymbol{x}_i\boldsymbol{\beta}_{\theta} + \epsilon_{i,\theta} > \boldsymbol{x}_i\boldsymbol{\beta}_{\bar{\theta}} + \epsilon_{i,\bar{\theta}}] \\
&= \Pr[\boldsymbol{x}_i\boldsymbol{\beta}_{\theta} + \epsilon_{i,\theta} - \boldsymbol{x}_i\boldsymbol{\beta}_{\bar{\theta}} - \epsilon_{i,\bar{\theta}} > 0] \\
&= \Pr[\boldsymbol{x}_i(\boldsymbol{\beta}_{\theta} - \boldsymbol{\beta}_{\bar{\theta}}) + \epsilon_{i,\theta} - \epsilon_{i,\bar{\theta}} > 0] \\
&= \Pr[\boldsymbol{x}_i\boldsymbol{\beta} + \epsilon_i > 0] \\
&= \Pr[\epsilon_i > -\boldsymbol{x}_i\boldsymbol{\beta}]
\end{aligned} \tag{9.9}$$

where $\boldsymbol{\beta} = \boldsymbol{\beta}_{\theta} - \boldsymbol{\beta}_{\bar{\theta}}$ and $\epsilon_i = \epsilon_{i,\theta} - \epsilon_{i,\bar{\theta}}$. By imposing the appropriate distributional assumption on ϵ_i we obtain, once again, the logit and probit models.

Stochastic utility theory provides a powerful framework for understanding binary and other choices. We shall encounter this framework again in Chapter 10, where we shall consider choices between more than two alternatives.

The Problem of Model Identification

You may wonder why we need to fix the variance of the stochastic components. In the LPM, as in regression analysis more generally, the variance is an estimated parameter (see Chapter 3.3), so why is it not here? The reason for fixing the variance is that the underlying latent variable, y_i^*, has no scale. This is different from regression analysis, since the response variable is measured in that approach. With the scale of y_i^* undefined, the scale of $\boldsymbol{\beta}$ is also undefined. Thus, the parameters are under-identified, meaning that they could be changed by multiplying them with an arbitrary constant δ.[14] By fixing the variance of ϵ_i, we standardize the scale of y_i^* which, in turn, allows us to fix the scale of $\boldsymbol{\beta}$. Thus, a variance assumption concerning ϵ_i is a necessary identifying constraint without which estimation would fail.

There is nothing sacred about setting the variance of ϵ_i to 1, as is done in the probit model, or to $\pi^2/3$, as is done in the logit model. These particular constraints are arbitrary. This fact has two important implications. First, because the scale of $\boldsymbol{\beta}$ is tied to the particular variance assumption, one can expect logit and probit estimates to come out differently, a point we shall

[14] To see this, imagine that we replace y_i^* with $w_i^* = \delta y_i^*$. Since $y_i^* = \boldsymbol{x}_i\boldsymbol{\beta} + \epsilon_i$, it follows that $w_i^* = \delta(\boldsymbol{x}_i\boldsymbol{\beta} + \epsilon_i) = \boldsymbol{x}_i(\delta\boldsymbol{\beta}) + \delta\epsilon_i$. Hence, $\boldsymbol{\beta}_{w^*} = \delta\boldsymbol{\beta}_{y^*}$. Note that $V[w^*] = \delta^2 V[y^*]$.

discuss in greater detail in Chapter 9.2.3. Second, because the scale of $\boldsymbol{\beta}$ is fixed arbitrarily, it makes no sense to give a direct interpretation to the coefficients contained in this vector.

One might wonder wether the arbitrariness of the identifying constraint does not completely jeopardize the validity of the logit and probit models. The answer is negative because the identifying assumption has no impact on π_i, which is the quantity that interests us ultimately.[15] It is easy to illustrate this for the logit model. The standard logistic distribution may also be written as

$$F(\epsilon) = \frac{1}{\exp(-\epsilon)}$$

In this distribution, $V[\epsilon] = \pi^2/3$. Imagine that we want to transform ϵ so that it has a variance of 1, just as in the probit model. We can do this by dividing ϵ by σ, which is the square root of $V[\epsilon]$. Of course, if we transform ϵ this way, then we should transform the other elements of (9.5) as well. Thus,

$$\begin{aligned} \frac{y_i^*}{\sigma} &= \frac{\boldsymbol{x}_i\boldsymbol{\beta}}{\sigma} + \frac{\epsilon_i}{\sigma} \\ &= \frac{\boldsymbol{x}_i\boldsymbol{\beta}}{\sigma} + \nu_i \end{aligned}$$

Here ν follows the standardized logistic distribution, which is given by

$$F(\nu) = \frac{1}{\exp\left(-\frac{\pi}{\sqrt{3}}\nu\right)}$$

If we now substitute ϵ/σ for ν, we get

$$\begin{aligned} F(\nu) &= \frac{1}{\exp\left(-\frac{\pi}{\sqrt{3}}\frac{\epsilon\sqrt{3}}{\pi}\right)} \\ &= \frac{1}{\exp(-\epsilon)} \end{aligned}$$

Thus, changing the variance (and thus the identifying restriction) has no impact on the CDF. As a result, the probabilities remain unaffected, even if the scale of the parameters is changed by changing the variance of the stochastic component.

[15] That is, π_i is an estimable function, which means that it is invariant to any identifying constraints imposed on the parameters (e.g. Searle 1971).

9.2.2 Estimation

Likelihood and Log-Likelihood Functions

Unlike the LPM, estimation of the logit and probit models requires MLE. Assume that we have drawn a sample of n independent observations. On the response variable, these observations are a collection of 0s and 1s, where the 1s occur with probability π_i and the 0s with probability $1 - \pi_i$. The likelihood function is then given by

$$\begin{aligned}
\mathcal{L} &= \prod_{y_i=1} f(y_i = 1) \prod_{y_i=0} f(y_i = 0) \\
&= \prod_{y_i=1} \pi_i \prod_{y_i=0} (1 - \pi_i) \\
&= \prod_{y_i=1} F(\boldsymbol{x}_i\boldsymbol{\beta}) \prod_{y_i=0} [1 - F(\boldsymbol{x}_i\boldsymbol{\beta})]
\end{aligned}$$

This can be written more compactly as

$$\mathcal{L} = \prod_{i=1}^{n} [F(\boldsymbol{x}_i\boldsymbol{\beta})]^{y_i} [1 - F(\boldsymbol{x}_i\boldsymbol{\beta})]^{1-y_i}$$

where $F(.)$ is substituted for by either $\Phi(.)$, in the case of probit, or $\Lambda(.)$, in the case of logit. Keeping in mind that both of these distributions are symmetrical, we know that $1 - F(\boldsymbol{x}_i\boldsymbol{\beta}) = F(-\boldsymbol{x}_i\boldsymbol{\beta})$. This suggests yet another formulation of the likelihood function:

$$\mathcal{L} = \prod_{i=1}^{n} F(q_i\boldsymbol{x}_i\boldsymbol{\beta})$$

where $q_i = 2y_i - 1$. The corresponding log-likelihood function can be formulated as:

$$\begin{aligned}
\ell &= \sum_{i=1}^{n} \{y_i \ln[F(\boldsymbol{x}_i\boldsymbol{\beta})] + (1 - y_i) \ln[1 - F(\boldsymbol{x}_i\boldsymbol{\beta})]\} \\
&= \sum_{i=1}^{n} \ln[F(q_i\boldsymbol{x}_i\boldsymbol{\beta})]
\end{aligned}$$

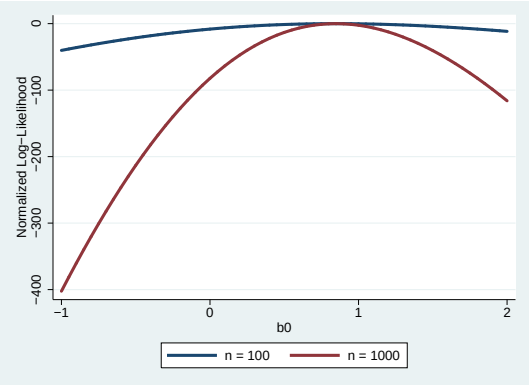

Figure 9.2: Logit Log-Likelihoods for a Constant-Only Model

Optimization

In order to optimize the log-likelihood function we follow the usual procedure of evaluating the first-order condition. The gradient is given by

$$\begin{aligned}\frac{\partial \ell}{\partial \boldsymbol{\beta}'} &= \sum_{i=1}^{n}\left[y_i\frac{f(\boldsymbol{x}_i\boldsymbol{\beta})}{F(\boldsymbol{x}_i\boldsymbol{\beta})} - (1-y_i)\frac{f(\boldsymbol{x}_i\boldsymbol{\beta})}{1-F(\boldsymbol{x}_i\boldsymbol{\beta})}\right]\boldsymbol{x}_i \\ &= \sum_{i=1}^{n} q_i\frac{f(q_i\boldsymbol{x}_i\boldsymbol{\beta})}{F(q_i\boldsymbol{x}_i\boldsymbol{\beta})}\boldsymbol{x}_i\end{aligned}$$

where $f(.)$ is the PDF, i.e. $\phi(.)$ in the case of probit and $\lambda(.)$ in the case of logit.

The likelihood equation sets these derivatives equal to zero. Since the derivatives are nonlinear functions, the solution to the likelihood equation has to be found by way of numerical methods. Generally, this does not pose complications since the logit and probit log-likelihood functions are typically well-behaved.[16] Specifically, the log-likelihood functions are single-peaked, which means that a unique maximum exists. Figure 9.2 illustrates the logit log-likelihood functions for different sample sizes. The figure clearly shows the single-peakedness of the log-likelihood function, as well as the relationship between curvature and sample size.

While it is generally easy to optimize the logit and probit log-likelihood functions, problems can occur when there is **perfect classification** or **sep-**

[16]Concavity problems may arise if the data contain extremely few 0s or 1s. In this case, it may be wise to look for a different probability distribution model.

aration. This happens when we can perfectly predict a score of one or zero from one of the covariates (e.g. every Republican voted for Bush in 2000). If we can perfectly predict a score of one, then $\pi_i = 1$. Likewise, if we can perfectly predict a score of zero, then $\pi_i = 0$. Such probabilities can be accommodated by the standard logistic and the standard normal distribution functions only by letting $|\boldsymbol{x}_i\boldsymbol{\beta}| \rightarrow \infty$, which, for finitely bounded covariates, means that $|\boldsymbol{\beta}| \rightarrow \infty$ (the standard errors will also tend to infinity). This will generally create numerical difficulties (e.g. potential singularity of the Hessian) in the algorithms that we discussed in Chapter 5. Statistical packages typically resolve the problem by dropping the offending predictor and observations. They then optimize the log-likelihood for the remaining observations and predictors.[17]

Logit versus Probit Estimates

Whether one optimizes a logit or probit log-likelihood does not matter much. As Figure 9.1 shows, the standard normal and standard logistic distribution functions look very similar, with the logistic distribution displaying slightly heavier tails. Although the theoretical development of the probit model preceded that of the logit model, for a while applied researchers preferred logit because of its better computational tractability. However, the development of powerful algorithms for computing normal probabilities and the availability of powerful computers have eliminated the historical advantage of logit. Nowadays, it is just as easy to run a probit as a logit analysis and the choice between them is much like flipping a coin.

This does not mean that logit and probit will give you exactly the same parameter estimates. Because these two models fix the variance at different values, the scale of the parameters is also different. Equating the variances of the standard logistic and normal distributions, we find $\beta_L \approx (\pi/\sqrt{3})\beta_P \Leftrightarrow \beta_L \approx 1.8\beta_P$, where β_L indicates the logit parameter and β_P the probit parameter.[18] Amemiya (1981) suggests that $\beta_L \approx 1.6\beta_P$ if other features of the standard logistic and normal distributions are also synchronized. Following the same logic, Long (1997) finds that $\beta_L \approx 1.7\beta_P$. Thus, the scale of the

[17] This is similar to concentrating the perfectly predicted observations out of the likelihood function (see Ruud 2000). For an alternative solution, which relies on a penalized log-likelihood function, see Zorn (2005).

[18] As Long (1997) suggests, this is probably the best approximation to use if you want to translate the coefficients reported in published research.

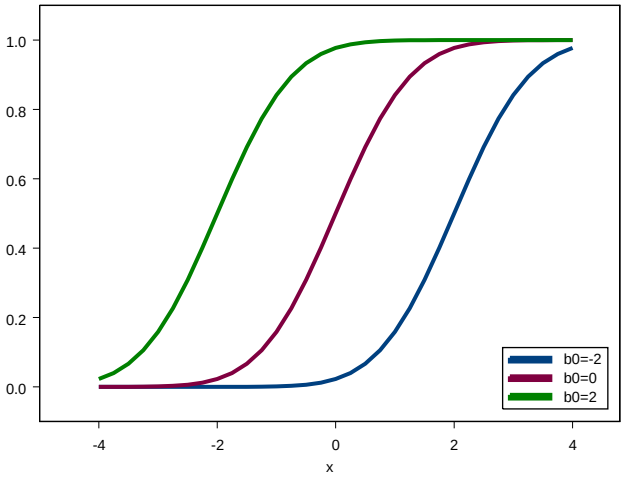

Figure 9.3: Role of the Constant in Logit and Probit

logit and probit parameter estimates is different, although it is possible to approximate one set of estimates by applying a proper transformation of the other set.

9.2.3 Interpretation

The sigmoidal relationship between the predictors and π_i means that the interpretation of logit and probit estimates is considerably more difficult than that of LPM estimates. Unlike the LPM model, the effect of predictors is not constant, which means that we have to rely on different interpretative devices than the standard phrase "for a unit increase in x_k, π is expected to change by β_k units." Before turning to these interpretative devices, it is useful to first say something more general about the meaning of logit and probit coefficients.

General Comments on the Meaning of Parameters

Consider the simple latent variable model $y_i^* = \beta_0 + \beta_1 x_i + \epsilon_i$. In this model, β_0 is a *shift parameter*. It shifts the entire probability curve to the left or to the right, as is illustrated in Figure 9.3. If $\beta_0 = 0$, then we attain $\pi_i = .5$

when $y_i^* = 0$. When $\beta_0 > 0$, then the curve shifts to the left, which means that we attain $\pi_i = .5$ when $y_i^* < 0$. This captures a world in which the net utility for an alternative can be low, yet decision makers are still inclined to choose that alternative. Finally, when $\beta_0 < 0$, then the curve shifts to the right, which means that we attain $\pi_i = .5$ when $y_i^* > 0$. This captures a situation in which decision makers are inclined to choose an alternative only if it gives them considerable net utility.

Turning our attention to β_1, the sign of this coefficient influences the *direction* of the relationship between π_i and x_i. As is illustrated in Figure 9.4, if $\beta_1 > 0$, then the probability of choosing an alternative increases with values of the predictor. On the other hand, if $\beta_1 < 0$, then choosing the alternative becomes less likely when the values of x_i increase. ($\beta_1 = 0$ captures the situation where the probability of choosing an alternative does not depend on x_i.)

In terms of its size, β_1 can be interpreted as a *stretch parameter*. As indicated in Figure 9.5, β_1 either stretches out or shrinks the probability curve, which influences how quickly changes in π_i occur. Specifically, the smaller β_1, the more stretched out the curve is, which means that changes in π_i occur more gradually. By contrast, the greater β_1, the more condensed the curve is, which means that changes in π_i occur rapidly. We also see that higher values of β_1 amplify the sigmoidal nature of the probability curve.

Interpretation in Terms of the Latent Variable

One way of interpreting logit and probit coefficients is to focus on the underlying latent variable, y_i^*. This interpretation has the advantage of being simple. Since y_i^* is a linear function of the predictors, we can say that it is expected to change by β_k units for a unit change in x_{ik}. On the downside, as an unobserved variable, y_i^* does not have an intrinsically meaningful scale. In fact, as we have seen, the scale units of y_i^* are dictated by our choice of error variance on ϵ_i. Because of the arbitrariness of the scale, interpretations in terms of the latent variable are rare.

Predicted Probabilities

Most of the interpretation of logit and probit coefficients occurs in terms of predicted probabilities or changes in those predicted probabilities, which will be discussed in the next two sub-sections. Predicted probabilities are

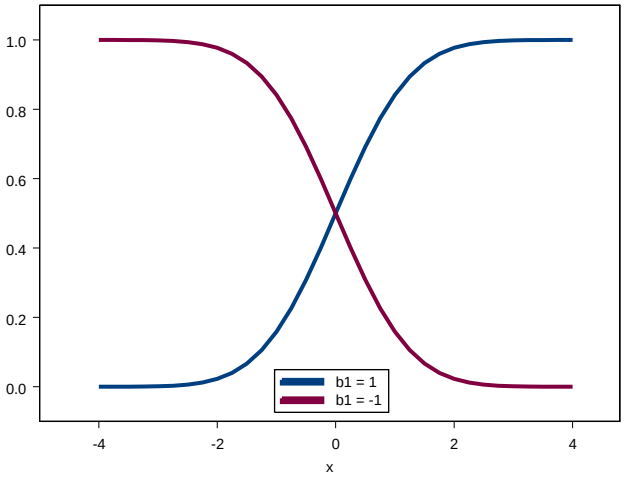

Figure 9.4: Impact of the Sign of β_1 in Logit and Probit

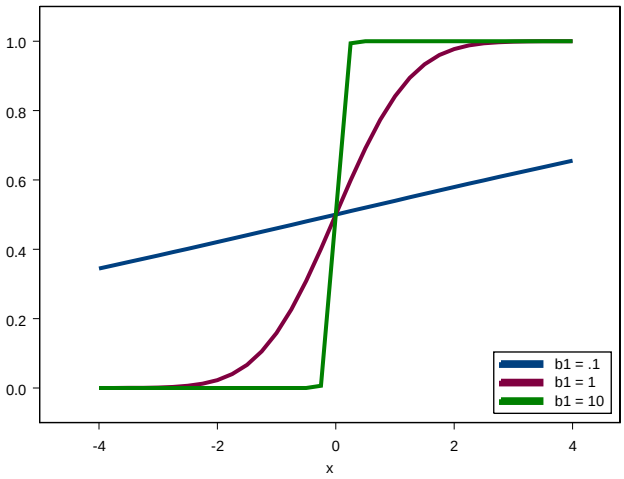

Figure 9.5: Impact of the Size of β_1 in Logit and Probit

computed as

$$\hat{\pi}_i \quad = \quad F(\boldsymbol{x}_i\hat{\boldsymbol{\beta}}) \tag{9.10}$$

where $F(.)$ is the standard normal or standard logistic distribution function. Equation (9.10) yields predicted probabilities for each of the sample units, based on that unit's values on the covariates. It is also possible to obtain a predicted probability by simulating a set of values on the covariates. For example, one might be interested in the probability of turnout of someone who scores average on all of the covariates. When one simulates covariates in this manner, one should always bear in mind that there may be no sample unit that fits the simulated profile. There are some risks associated with such out-of-sample predictions, namely that the prediction may not be realistic and that the parameter estimates might have come out differently if we really had included the unit that is average on all covariates. Nevertheless, predicted probabilities are an invaluable interpretative device, especially since they can be plotted to graphically present the effects of covariates (see the example below).

A drawback of (9.10) is that it produces a point estimate of the predicted probability and, as such, does not consider sampling variability. Ideally, then, one would like to compute a confidence interval for the predicted probabilities.[19] Xu and Long (2005) describe a number of procedures for computing confidence intervals, including endpoint transformations and the delta method.[20] The method of **endpoint transformations** works well when we are computing a confidence interval for a monotonic function of the linear predictions $\boldsymbol{x\beta}$, as is the case with the distribution functions of the probit and logit models (see Figure 9.1). In this case, one starts by computing a confidence interval for $\boldsymbol{x}\hat{\boldsymbol{\beta}}$, namely the symmetric interval $LB_{\boldsymbol{x\beta}} \leq \boldsymbol{x\beta} \leq UB_{\boldsymbol{x\beta}}$, where LB and UB denote the lower and upper bounds, respectively. One then transforms the endpoints or bounds of this confidence interval to obtain a confidence interval for $F(\boldsymbol{x\beta})$. Thus, $\{F(\boldsymbol{x\beta})_{LB}\} \leq F(\boldsymbol{x\beta} \leq \{F(\boldsymbol{x\beta})_{UB}\}$.

Some work is required in obtaining the lower and upper bounds for the linear prediction. First, from the properties of ML estimators we know that $\hat{\boldsymbol{\beta}} \sim N(\boldsymbol{\beta}, \mathbf{V}[\hat{\boldsymbol{\beta}}])$. Since $\boldsymbol{x}\hat{\boldsymbol{\beta}}$ is a linear function of $\hat{\boldsymbol{\beta}}$, we know that it,

[19] See Herron (1999) on the need to take into consideration uncertainty in post-estimation statistics in political analysis.

[20] A third procedure is bootstrapping. For a discussion of bootstrap confidence intervals see Chapter 7.4 in this report.

too, is asymptotically normally distributed: $\boldsymbol{x}\hat{\boldsymbol{\beta}} \sim N(\boldsymbol{x}\boldsymbol{\beta}, V[\boldsymbol{x}\hat{\boldsymbol{\beta}}])$.[21] Here, $V[\boldsymbol{x}\hat{\boldsymbol{\beta}}] = \boldsymbol{x}\mathbf{V}[\hat{\boldsymbol{\beta}}]\boldsymbol{x}'$. Thus, the $100(1-\alpha)\%$ confidence interval for $\boldsymbol{x}\boldsymbol{\beta}$ is given by

$$\boldsymbol{x}\hat{\boldsymbol{\beta}} - z_{\alpha/2}\sqrt{\boldsymbol{x}\mathbf{V}[\hat{\boldsymbol{\beta}}]\boldsymbol{x}'} \leq \boldsymbol{x}\boldsymbol{\beta} \leq \boldsymbol{x}\hat{\boldsymbol{\beta}} + z_{\alpha/2}\sqrt{\boldsymbol{x}\mathbf{V}[\hat{\boldsymbol{\beta}}]\boldsymbol{x}'}$$

where $1-\alpha$ is the confidence level and $-z_{\alpha/2}$ is the value of a standard normal variate such that $\Pr(z < -z_{\alpha/2}) = \alpha/2$. Using the method of endpoint transformations, the $100(1-\alpha)\%$ confidence interval for $F(\boldsymbol{x}\boldsymbol{\beta})$ is then given by

$$F\left\{\boldsymbol{x}\hat{\boldsymbol{\beta}} - z_{\alpha/2}\sqrt{\boldsymbol{x}\mathbf{V}[\hat{\boldsymbol{\beta}}]\boldsymbol{x}'}\right\} \leq F(\boldsymbol{x}\boldsymbol{\beta}) \leq F\left\{\boldsymbol{x}\hat{\boldsymbol{\beta}} + z_{\alpha/2}\sqrt{\boldsymbol{x}\mathbf{V}[\hat{\boldsymbol{\beta}}]\boldsymbol{x}'}\right\}$$

where $F(.) = \Lambda(.)$ for the logit model and $F(.) = \Phi(.)$ for the probit model. This interval requires that we specify a series of values for the covariates.

A different approach to obtaining confidence intervals, which is more generally applicable, is to rely on the **delta method**, which is described in greater detail in Section 9.5 of this chapter. Here we use a Taylor expansion to linearize the function $\hat{\pi} = F(\boldsymbol{x}\hat{\boldsymbol{\beta}})$, which yields $F(\boldsymbol{x}\boldsymbol{\beta}) + (\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})f(\boldsymbol{x}\boldsymbol{\beta})$. By this expansion, the asymptotic variance of $\hat{\pi}$ is given by

$$V[F(\boldsymbol{x}\hat{\boldsymbol{\beta}})] \;=\; \left\{f(\boldsymbol{x}\hat{\boldsymbol{\beta}})\right\}^2 \boldsymbol{x}\mathbf{V}[\hat{\boldsymbol{\beta}}]\boldsymbol{x}' \tag{9.11}$$

(see Greene 2003).[22] Relying again on the property that $F(\boldsymbol{x}\hat{\boldsymbol{\beta}})$ is normally distributed, the $100(1-\alpha)\%$ confidence interval for $\pi = F(\boldsymbol{x}\boldsymbol{\beta})$ is then given

[21] Here we rely on the Slutsky theorem, which states that if plim $\hat{\theta} = \theta$, then plim $g(\hat{\theta}) = g(\theta)$. From the asymptotic properties of ML estimators, we know that plim $\hat{\boldsymbol{\beta}} = \boldsymbol{\beta}$. Letting $g(\boldsymbol{\beta}) = \boldsymbol{x}\boldsymbol{\beta}$, it then follows that plim $\boldsymbol{x}\hat{\boldsymbol{\beta}} = \boldsymbol{x}\boldsymbol{\beta}$.

[22] *Proof:* Following the results from Section 9.5, the asymptotic variance of $F(\boldsymbol{x}\hat{\boldsymbol{\beta}})$ is given by

$$\left[\frac{\partial F(\boldsymbol{x}\hat{\boldsymbol{\beta}})}{\partial \hat{\boldsymbol{\beta}}'}\right] \mathbf{V}[\hat{\boldsymbol{\beta}}] \left[\frac{\partial F(\boldsymbol{x}\hat{\boldsymbol{\beta}})}{\partial \hat{\boldsymbol{\beta}}'}\right]',$$

Using the chain rule,

$$\frac{\partial F(\boldsymbol{x}\hat{\boldsymbol{\beta}})}{\partial \hat{\boldsymbol{\beta}}'} \;=\; \frac{dF(\hat{z})}{d\hat{z}}\frac{\partial \hat{z}}{\partial \hat{\boldsymbol{\beta}}'}$$

The first term on the right-hand side is equal to $f(\boldsymbol{x}\hat{\boldsymbol{\beta}})$, while (by convention) the second

by

$$F(\boldsymbol{x}\hat{\boldsymbol{\beta}}) - z_{\alpha/2} s.e._{F(\boldsymbol{x}\hat{\boldsymbol{\beta}})} \leq F(\boldsymbol{x}\boldsymbol{\beta}) \leq F(\boldsymbol{x}\hat{\boldsymbol{\beta}}) + z_{\alpha/2} s.e._{F(\boldsymbol{x}\hat{\boldsymbol{\beta}})}$$

where

$$s.e._{F(\boldsymbol{x}\hat{\boldsymbol{\beta}})} = \sqrt{\left\{f(\boldsymbol{x}\hat{\boldsymbol{\beta}})\right\}^2 \boldsymbol{x}\mathbf{V}[\hat{\boldsymbol{\beta}}]\boldsymbol{x}'}$$

As usual, this interval depends on the specification of the values of the covariates. You should also keep in mind that this is an asymptotic confidence interval. As such, it may not be wise to report the interval when the sample size is small.

Marginal Effects and Elasticities

Marginal Effects Econometricians often interpret logit and probit results in terms of marginal effects, which are also known as partial effects. In general terms, the marginal effect of a predictor is defined as the partial derivative of a prediction function with respect to that predictor. In the case of logit and probit, the prediction function is the predicted probability, π_i, so that the marginal effect is defined as $\partial\pi_i/\partial x_{ik}$.[23] Using the chain rule of differentiation, this can be computed as

$$\begin{aligned} \frac{\partial \pi_i}{\partial x_{ik}} &= \frac{\partial F(\boldsymbol{x}_i\boldsymbol{\beta})}{\partial x_{ik}} \\ &= \frac{dF(\boldsymbol{x}_i\boldsymbol{\beta})}{d\boldsymbol{x}_i\boldsymbol{\beta}} \frac{\partial \boldsymbol{x}_i\boldsymbol{\beta}}{\partial x_{ik}} \\ &= f(\boldsymbol{x}_i\boldsymbol{\beta})\beta_k \end{aligned} \tag{9.12}$$

term is equal to $\boldsymbol{x}$. Hence

$$\begin{aligned} \left[\frac{\partial F(\boldsymbol{x}\hat{\boldsymbol{\beta}})}{\partial \hat{\boldsymbol{\beta}}'}\right] \mathbf{V}[\hat{\boldsymbol{\beta}}] \left[\frac{\partial F(\boldsymbol{x}\hat{\boldsymbol{\beta}})}{\partial \hat{\boldsymbol{\beta}}'}\right]' &= f(\boldsymbol{x}\hat{\boldsymbol{\beta}})\boldsymbol{x}\mathbf{V}[\hat{\boldsymbol{\beta}}]\left[f(\boldsymbol{x}\hat{\boldsymbol{\beta}})\boldsymbol{x}\right]' \\ &= \left\{f(\boldsymbol{x}\hat{\boldsymbol{\beta}})\right\}^2 \boldsymbol{x}\mathbf{V}[\hat{\boldsymbol{\beta}}]\boldsymbol{x}' \end{aligned}$$

[23] I should note that marginal effects assume that x_k is continuous. When it is a dummy variable, then taking the derivative of π with respect to the predictor will not work. Many statistical packages will in this case resort to reporting discrete change in predicted probabilities.

Here $f(.)$ is the appropriate PDF, i.e. $\phi(.)$ in the case of probit and $\lambda(.)$ in the case of logit. In mathematical terms, the marginal effect is the slope of the probability curve relating x_{ik} to π_i, while holding all other predictors constant. This may be interpreted as the instantaneous change in the predicted probability, i.e. the change in π_i due to an infinitisimal change in the value of x_{ik}.[24] Note that the maximum marginal effects in logit and probit models occur at $\pi_i = .5$.[25]

The *sign* of the marginal effect is driven entirely by β_k, since $f(\boldsymbol{x}_i\boldsymbol{\beta}) > 0$ by virtue of the fact that $f(.)$ is a PDF. The *magnitude* of the marginal effect is driven by both β_k and $\boldsymbol{x}_i\boldsymbol{\beta}$, which includes x_{ik}.[26] There are two

[24]Econometricians sometimes treat $f(\boldsymbol{x}_i\boldsymbol{\beta})$ as a scale parameter, so that the marginal effect is equal to a scale parameter times a predictor's coefficient. Viewed in this light, marginal effects simply re-scale the original coefficients.

[25]*Proof:* Let $w_i = \boldsymbol{x}_i\boldsymbol{\beta}$. The marginal effect with respect to w_i is given by

$$\frac{\partial \pi_i}{\partial w_i} = f(w_i)$$

If we seek to maximize this effect, then we should take its first derivative, set it to 0, and solve for w_i:

$$\frac{\partial^2 \pi_i}{\partial w_i^2} = f'(w_i)$$

For the probit model, $f'(w_i) = -w_i\phi(w_i)$. This is equal to 0 when $w_i = 0$, which corresponds to $\pi_i = \Phi(0) = .5$. For the logit model, $f'(w_i) = [1 - 2\Lambda(w_i)]\Lambda(w_i)[1 - \Lambda(w_i)]$, which is equal to

$$-\frac{\exp(w_i)\left[\exp(w_i) - 1\right]}{\left[1 + \exp(w_i)\right]^3}$$

This is equal to 0 when $\exp(w_i) - 1 = 0$, which happens when $w_i = 0$. Again, this corresponds to $\pi_i = \Lambda(0) = .5$.

[26]However, ratios of marginal effects do not depend on $\boldsymbol{x}_i\boldsymbol{\beta}$. It is easy to see this. The marginal effect of x_k is defined as $f(\boldsymbol{x}_i\boldsymbol{\beta})\beta_k$ Likewise, the marginal effect of x_m is defined as $f(\boldsymbol{x}_i\boldsymbol{\beta})\beta_m$. If we now take the ratio of these marginal effects, we are left with the ratio of the parameters:

$$\begin{aligned}\frac{\frac{\partial \pi_i}{\partial x_{ik}}}{\frac{\partial \pi_i}{\partial x_{im}}} &= \frac{f(\boldsymbol{x}_i\boldsymbol{\beta})\beta_k}{f(\boldsymbol{x}_i\boldsymbol{\beta})\beta_m} \\ &= \frac{\beta_k}{\beta_m}\end{aligned}$$

The ratio of the marginal effects provides a nice way to compare the magnitude of the

main approaches to setting the values of $\boldsymbol{x}$:

1. **Marginal Effects at the Mean:** One approach is to set all of the predictors, including x_k, at their mean values, producing so-called marginal effects at the mean or MEMs:

$$MEM_k = f(\bar{\mathbf{x}}\boldsymbol{\beta})\beta_k$$

   This can be simplified considerably if the predictors have been centered about their means. In this case, the means of the centered predictors are all zero and $f(\bar{\mathbf{x}}\boldsymbol{\beta}) = f(\beta_0)$ (see Anderson and Newell 2003).

2. **Average Marginal Effects:** Another approach is to use the actual values of the sample units on the predictors, which produces separate marginal effects estimates for each unit. These estimates are than averaged to produce an average marginal effect or AME:

$$AME_k = \frac{1}{n}\sum_{i=1}^{n} f(\mathbf{x}_i\boldsymbol{\beta})\beta_k$$

Although MEMs are commonly computed by statistical software packages, they have an important drawback: there may not be a single sample unit that is average on all of the predictors. Thus MEMs may represent imaginary units, a problem that the AME avoids by relying on extant values on the predictors.[27]

Regardless of whether one computes the MEM or AME, it should be kept in mind that the resulting statistics are estimates of the marginal effects in the population. As such, it would be helpful to obtain a measure of the sampling variability of the marginal effects. An asymptotic estimate of this variability can be obtained once more by using the delta method. Let $\hat{\boldsymbol{\gamma}} = f(\boldsymbol{x}\hat{\boldsymbol{\beta}})\hat{\boldsymbol{\beta}}$ be the vector of marginal effects, then the asymptotic variance is given by

$$\mathbf{V}[\hat{\gamma}] = \left[\frac{\partial\hat{\boldsymbol{\gamma}}}{\partial\hat{\boldsymbol{\beta}}'}\right]\mathbf{V}[\hat{\boldsymbol{\beta}}]\left[\frac{\partial\hat{\boldsymbol{\gamma}}}{\partial\hat{\boldsymbol{\beta}}'}\right]' \tag{9.13}$$

effects of covariates.

[27] The choice between MEM and AME is not without consequence, as these procedures may produce different estimates of the effects of predictors. Bartus (2005) approximates their relative difference as

$$\frac{AME_k - MEM_k}{MEM_k} \approx .5\frac{f"(\bar{\mathbf{x}}\boldsymbol{\beta})}{f(\bar{\mathbf{x}}\boldsymbol{\beta})}V[\mathbf{x}\boldsymbol{\beta}]$$

where[28]

$$\left[\frac{\partial\hat{\boldsymbol{\gamma}}}{\partial\hat{\boldsymbol{\beta}}'}\right] = f(\boldsymbol{x}\hat{\boldsymbol{\beta}})\mathbf{I} + \left(\frac{df(\boldsymbol{x}\hat{\boldsymbol{\beta}})}{d\boldsymbol{x}\hat{\boldsymbol{\beta}}}\right)\hat{\boldsymbol{\beta}}\boldsymbol{x}$$

For the probit model, $df(z)/dz = -z\phi(z)$. Substitution and simplification then yields

$$\mathbf{V}[\hat{\boldsymbol{\gamma}}] = \left[\phi(\boldsymbol{x}\hat{\boldsymbol{\beta}})\right]^2 \left[\mathbf{I} - \boldsymbol{x}\hat{\boldsymbol{\beta}}\hat{\boldsymbol{\beta}}\boldsymbol{x}\right]\mathbf{V}[\hat{\boldsymbol{\beta}}]\left[\mathbf{I} - \boldsymbol{x}\hat{\boldsymbol{\beta}}\hat{\boldsymbol{\beta}}\boldsymbol{x}\right]'$$

For the logit model, $\lambda = \Lambda(1-\Lambda)$ and (applying the chain rule) $d\lambda(z)/dz = (1-2\Lambda(z))(d\Lambda(z)/dz)$. Substitution and simplification then yields

$$\begin{aligned}\mathbf{V}[\hat{\boldsymbol{\gamma}}] = {} & \left\{\Lambda(\boldsymbol{x}\hat{\boldsymbol{\beta}})\left[1-\Lambda(\boldsymbol{x}\hat{\boldsymbol{\beta}})\right]\right\}^2 \times \\ & \left\{\mathbf{I} + \left[1-2\Lambda(\boldsymbol{x}\hat{\boldsymbol{\beta}})\right]\hat{\boldsymbol{\beta}}\boldsymbol{x}\right\}\mathbf{V}[\hat{\boldsymbol{\beta}}]\left\{\mathbf{I} + \left[1-2\Lambda(\boldsymbol{x}\hat{\boldsymbol{\beta}})\right]\hat{\boldsymbol{\beta}}\boldsymbol{x}\right\}'\end{aligned}$$

Computation of these variances again requires that one selects a particular set of values, i.e. profile, for $\boldsymbol{x}$.

With the variance of the marginal effects defined, it is now possible to compute asymptotic confidence intervals on the marginal effects. Taking advantage of the fact that a function of a maximum likelihood function is itself normally distributed (see Section 9.5), the $100(1-\alpha)\%$ confidence interval for the marginal effect of a predictor x_k is given by

$$\hat{\gamma}_k - z_{\alpha/2}s.e._{\hat{\gamma}_k} \leq \gamma_k \leq \hat{\gamma}_k + z_{\alpha/2}s.e._{\hat{\gamma}_k}$$

[28] *Proof:* The derivation of this result depends on a number of rules and conventions of matrix differentiation. First, by virtue of the product rule,

$$\frac{\partial f(\boldsymbol{x}\hat{\boldsymbol{\beta}})\hat{\boldsymbol{\beta}}}{\partial\hat{\boldsymbol{\beta}}'} = \hat{\boldsymbol{\beta}}\frac{\partial f(\boldsymbol{x}\hat{\boldsymbol{\beta}})}{\partial\hat{\boldsymbol{\beta}}'} + f(\boldsymbol{x}\hat{\boldsymbol{\beta}})\frac{\partial\hat{\boldsymbol{\beta}}}{\partial\hat{\boldsymbol{\beta}}'}$$

The derivative $\partial\hat{\boldsymbol{\beta}}/\partial\hat{\boldsymbol{\beta}}'$ is equal to an identity matrix. To obtain $\partial f(\boldsymbol{x}\hat{\boldsymbol{\beta}})/\partial\hat{\boldsymbol{\beta}}'$ we can apply the chain rule. Define $\hat{z} = \boldsymbol{x}\hat{\boldsymbol{\beta}}$, then

$$\frac{\partial f(\boldsymbol{x}\hat{\boldsymbol{\beta}})}{\partial\hat{\boldsymbol{\beta}}'} = \frac{df(\hat{z})}{d\hat{z}}\frac{\partial\hat{z}}{\partial\hat{\boldsymbol{\beta}}'}$$

The second term on the right-hand side yields $\boldsymbol{x}$. Combining the different results then produces the desired derivative.

Here $\hat{\gamma}_k = f(\boldsymbol{x}\hat{\boldsymbol{\beta}})\hat{\beta}_k$ is the estimated marginal effect of x_k and $s.e._{\hat{\gamma}_k}$ is the square root of the diagonal element of $\mathbf{V}[\hat{\boldsymbol{\gamma}}]$ that corresponds to $\hat{\gamma}_k$.

Special care should be taken when computing the marginal effects for models including polynomial and interaction terms. Consider, for example, the latent variable model $y_i^* = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \epsilon_i$. When computing the marginal effect with respect to x_i it should not be forgotten that this covariate appears both in linear and quadratic terms. The correct marginal effect in this case is

$$\frac{\partial \pi_i}{\partial x_{ik}} = f(.)(\beta_1 + 2\beta_2 x_i)$$

where $f(.) = f(\beta_0 + \beta_1 x_i + \beta_2 x_i^2)$. As another example, consider the latent variable model $y_i^* = \beta_0 + \beta_1 x_i + \beta_2 z_i + \beta_{12} x_i z_i$. Proper computation of the marginal effect of the interaction term $x_i z_i$ now requires that we differentiate with respect to both x_i and z_i. This yields:

$$\frac{\partial^2 \pi_i}{\partial x_i \partial z_i} = f(.)\beta_{12} + f'(.)(\beta_1 + \beta_{12} z_i)(\beta_2 + \beta_{12} x_i)$$

where $f(.) = f(\beta_0 + \beta_1 x_i + \beta_2 z_i + \beta_{12} x_i z_i)$ and $f'(.) = f'(\beta_0 + \beta_1 x_i + \beta_2 z_i + \beta_{12} x_i z_i)$ is the first derivative of the PDF.[29] Notice that this result implies that there can be a marginal effect of the interaction even if $\beta_3 = 0$. This stands in stark contrast with linear regression analysis where $\beta_3 = 0$ would imply the absence of an interaction. Moreover, the sign of the interaction effect cannot be read directly from the sign of β_3, as the other terms of the marginal effect may run in an opposite direction.

Elasticities Elasticities are closely associated with marginal effects, but offer a different interpretation of a predictor's effect. Specifically, we can define the elasticity, κ, of the kth predictor as

$$\begin{aligned} \kappa_k &= \frac{\partial \pi_i}{\partial x_{ik}} \frac{x_{ik}}{\pi_i} \\ &= \frac{\partial \ln(\pi_i)}{\partial \ln(x_{ik})} \end{aligned} \tag{9.14}$$

[29] For an extensive discussion of marginal effects of interaction terms in logit and probit models see Ai and Norton (2003) and Norton, Wang, and Ai (2004). These papers also discuss the derivation of the asymptotic variance of the marginal effects.

The elasticity is thus the product of the marginal effect of x_k and the ratio of this predictor over the predicted probability.[30] Elasticities have a straightforward and useful interpretation: they give the percentage change in π in response to a one percent change in the predictor. In light of this interpretation, it makes sense to compute elasticities only for continuous predictors.

In computing elasticities, it is again essential to set the predictors to a particular value. As with marginal effects, elasticities can be computed by setting all of the predictors to their means or by using the actual values of the predictors for each sample unit and then averaging the unit-specific elasticities. A variation on the latter procedure is to weight each individual elasticity by the predicted probability before aggregation (see Hensher and Johnson 1981).

Discrete Change in Predicted Probabilities

A different approach to interpretation is to compute the discrete change in predicted probabilities.[31] Here, we do not consider the instantaneous change in predicted probability, as we do with marginal effects, but rather the change in predicted probabilities when a predictor changes from one discrete value to another.[32] Because the simulated changes in the predictor are discrete, this method works both for continuous and discrete covariates.

Discrete change in the predicted probabilities is defined as the change in π in response to a change of δ units in the predictor of interest, while holding all other covariates, $\boldsymbol{x}$, constant:

$$\frac{\Delta\pi}{\Delta x_k} = \Pr(y = 1|\boldsymbol{x}, x_k + \delta) - \Pr(y = 1|\boldsymbol{x}, x_k) \tag{9.15}$$

It is conventional to set all of the remaining covariates to their means, so that the discrete change is computed as $\Pr(y = 1|\bar{\boldsymbol{x}}, x_k + \delta) - \Pr(y = 1|\bar{\boldsymbol{x}}, x_k)$. However, other choices are surely possible and may be more appropriate. For

[30] More generally, elasticities are defined as $(\partial y/\partial x)(x/y) = (\partial y/y)/(\partial x/x)$, where y is some prediction function.

[31] Discrete changes in π are sometimes also labeled as "first differences" in predicted probabilities.

[32] As a result, discrete change in probabilities generally does not yield the same results as marginal effects. Only if the discrete change in the predictor is made very small should we expect the discrete change in predicted probability to resemble the marginal effect. In relatively linear portions of the probability curve the two measures may also appear similar.

example, when some of the covariates are dummy variables, it may be useful to set their values equal to the mode, while ordinal scales can be set equal to their medians. You are free to set the reference values of the remaining covariates as long as you use the same values in both terms on the left-hand side of (9.15) and as long as you document your choice of reference values.

Just as you have a choice in selecting the values of the other covariates, you have a choice in defining δ. Common choices for the change in x_k include the following.

1. A **unit change** relative to $\bar{x}_k$, so that $\delta = 1$ and $\Delta\pi/\Delta x_k = \Pr(y = 1|\bar{\boldsymbol{x}}, \bar{x}_k + 1) - \Pr(y = 1|\bar{\boldsymbol{x}}, \bar{x}_k)$. This is tantamount to setting all covariates to their mean values initially and then increasing only x_k by one unit.

2. A **centered unit change** relative to $\bar{x}_k$, so that x_k moves from $\bar{x}_k - .5$ to $\bar{x}_k + .5$, as proposed by Kaufman (1996). The discrete change in predicted probability is then $\Delta\pi/\Delta x_k = \Pr(y = 1|\bar{\boldsymbol{x}}, \bar{x}_k + .5) - \Pr(y = 1|\bar{\boldsymbol{x}}, \bar{x}_k - .5)$.

3. A **standard deviation change** relative to $\bar{x}_k$, so that x_k moves from $\bar{x}_k - .5s_k$ to $\bar{x}_k + .5s_k$, where s_k is the sample standard deviation of x_k. The discrete change is then defined as $\Delta\pi/\Delta x_k = \Pr(y = 1|\bar{\boldsymbol{x}}, \bar{x}_k + .5s_k) - \Pr(y = 1|\bar{\boldsymbol{x}}, \bar{x}_k - .5s_k)$.

4. It is also possible to compute the **maximum possible change**. Here we let x_k move from the minimum to the maximum in the sample, so that δ is equal to the range and the discrete change is defined as $\Delta\pi/\Delta x_k = \Pr[y = 1|\bar{\boldsymbol{x}}, \max(x_k)] - \Pr[y = 1|\bar{\boldsymbol{x}}, \min(x_k)]$. This manner of computing discrete change is quite common in political analysis, although it is sometimes criticized for creating an exaggerated view of the effects of predictors because very few cases may fall at the extremes.

5. For dummy predictors, computing the maximum possible change is tantamount to computing a **change from 0 to 1**. Thus, $\Delta\pi/\Delta x_k = \Pr(= 1|\bar{\boldsymbol{x}}, x_k = 1) - \Pr(y = 1|\bar{\boldsymbol{x}}, x_k = 0)$. This computation is not subject to the same criticism as maximum possible change. On the contrary, this is actually one of the best ways to characterize the effect of dummy predictors.

6. **Percentile change** provides another method of operationalizing discrete change. Here, we let x_k move from the pth percentile to the $100-p$th percentile, e.g. from the 10th to the 90th percentile. Letting $x_{k,p}$ and $x_{k,100-p}$ denote the values of x_k that correspond to the pth and $100-p$th percentiles, respectively, discrete change is defined as $\Delta\pi/\Delta x_k = \Pr(y=1|\bar{\boldsymbol{x}}, x_{k,100-p}) - \Pr(y=1|\bar{\boldsymbol{x}}, x_{k,p})$. Especially when p is not chosen too small, the percentile change method can help to overcome the criticism of maximum possible change. After all, we would know that $p\%$ of the cases would score lower and $p\%$ would score higher than the values that are used to demarcate the discrete change in x_k.

The different variants of discrete change all produce point estimates of $\Delta\pi/\Delta x_k$. It is often useful to complement these point estimates with interval estimates. To obtain a confidence interval for the discrete change we must begin by computing the asymptotic variance of $\Delta\hat{\pi}/\Delta x_k$, which can be done using the delta method (see Xu and Long 2005). Let $\boldsymbol{x}_a$ be one set of values for the predictors and let $\boldsymbol{x}_b$ be another set of values. For example, $\boldsymbol{x}_a$ could set x_k to the maximum and all other predictors to the mean, while $\boldsymbol{x}_b$ could set x_k to the minimum while still keeping all other predictors at the mean. We are interested in the variance of $F(\boldsymbol{x}_a\hat{\boldsymbol{\beta}}) - F(\boldsymbol{x}_b\hat{\boldsymbol{\beta}})$. To this effect, we begin by defining the partial derivative of this quantity with respect to $\hat{\boldsymbol{\beta}}'$:[33]

$$\frac{\partial F(\boldsymbol{x}_a\hat{\boldsymbol{\beta}}) - F(\boldsymbol{x}_b\hat{\boldsymbol{\beta}})}{\partial \hat{\boldsymbol{\beta}}'} = f(\boldsymbol{x}_a\hat{\boldsymbol{\beta}})\boldsymbol{x}_a - f(\boldsymbol{x}_b\hat{\boldsymbol{\beta}})\boldsymbol{x}_b = \boldsymbol{q}$$

The variance of $F(\boldsymbol{x}_a\hat{\boldsymbol{\beta}}) - F(\boldsymbol{x}_b\hat{\boldsymbol{\beta}})$ is given by

$$\begin{aligned} V[F(\boldsymbol{x}_a\hat{\boldsymbol{\beta}}) - F(\boldsymbol{x}_b\hat{\boldsymbol{\beta}})] &= \boldsymbol{q}\mathbf{V}[\hat{\boldsymbol{\beta}}]\boldsymbol{q}' \\ &= f(\boldsymbol{x}_a\hat{\boldsymbol{\beta}})^2\boldsymbol{x}_a\mathbf{V}[\hat{\boldsymbol{\beta}}]\boldsymbol{x}'_a + f(\boldsymbol{x}_b\hat{\boldsymbol{\beta}})^2\boldsymbol{x}_b\mathbf{V}[\hat{\boldsymbol{\beta}}]\boldsymbol{x}'_b - \\ &\quad 2f(\boldsymbol{x}_a\hat{\boldsymbol{\beta}})f(\boldsymbol{x}_b\hat{\boldsymbol{\beta}})\boldsymbol{x}_a\mathbf{V}[\hat{\boldsymbol{\beta}}]\boldsymbol{x}'_b \end{aligned}$$

[33] *Proof:* The proof relies on the property that the derivative of a difference of two functions is equal to the difference between the derivatives. Thus,

$$\begin{aligned} \frac{\partial F(\boldsymbol{x}_a\hat{\boldsymbol{\beta}}) - F(\boldsymbol{x}_b\hat{\boldsymbol{\beta}})}{\partial \hat{\boldsymbol{\beta}}'} &= \frac{\partial F(\boldsymbol{x}_a\hat{\boldsymbol{\beta}})}{\partial \hat{\boldsymbol{\beta}}'} - \frac{\partial F(\boldsymbol{x}_b\hat{\boldsymbol{\beta}})}{\partial \hat{\boldsymbol{\beta}}'} \\ &= \frac{dF(\hat{z}_a)}{d\hat{z}_a}\frac{\partial \hat{z}_a}{\partial \hat{\boldsymbol{\beta}}'} - \frac{dF(\hat{z}_b)}{d\hat{z}_b}\frac{\partial \hat{z}_b}{\partial \hat{\boldsymbol{\beta}}'} \\ &= f(\hat{z}_a)\boldsymbol{x}_a - f(\hat{z}_b)\boldsymbol{x}_b \end{aligned}$$

where $\hat{z}_a = \boldsymbol{x}_a\hat{\boldsymbol{\beta}}$ and $\hat{z}_b = \boldsymbol{x}_b\hat{\boldsymbol{\beta}}$.

Since $F(\boldsymbol{x}_a\hat{\boldsymbol{\beta}}) - F(\boldsymbol{x}_b\hat{\boldsymbol{\beta}})$ is normally distributed, a confidence interval for the discrete change is given by

$$\hat{\Delta} - z_{\alpha/2} s.e._{\hat{\Delta}} \leq \frac{\Delta\pi}{\Delta x_k} \leq \hat{\Delta} + z_{\alpha/2} s.e._{\hat{\Delta}}$$

where $\hat{\Delta} = F(\boldsymbol{x}_a\hat{\boldsymbol{\beta}}) - F(\boldsymbol{x}_b\hat{\boldsymbol{\beta}})$ and $s.e._{\hat{\Delta}} = \sqrt{V[F(\boldsymbol{x}_a\hat{\boldsymbol{\beta}}) - F(\boldsymbol{x}_b\hat{\boldsymbol{\beta}})]}$. Obviously, this confidence interval depends on the specific values placed in $\boldsymbol{x}_a$ and $\boldsymbol{x}_b$.

As was the case with marginal effects, special care should be taken in correctly specifying the discrete change of polynomial and interaction terms. Consider again the latent variable model $y_i^* = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \epsilon_i$. If we let x move from $\bar{x} - .5$ to $\bar{x} + .5$, then the discrete change is given by $F[\beta_0 + \beta_1(\bar{x} + .5) + \beta_2(\bar{x} + .5)^2] - F[\beta_0 + \beta_1(\bar{x} - .5) + \beta_2(\bar{x} - .5)^2]$. The change thus has to be introduced in both the linear and quadratic terms.

To see what happens with the discrete change on interactions, let us consider two dummy predictors, x and z, in the latent variable model $y_i^* = \beta_0 + \beta_1 x_i + \beta_2 z_i + \beta_{12} x_i z_i + \epsilon_i$. The discrete change term for the interaction is[34]

$$\frac{\Delta^2\pi}{\Delta x \Delta z} = F(\beta_0 + \beta_1 + \beta_2 + \beta_{12}) - F(\beta_0 + \beta_1) - F(\beta_0 + \beta_2) + F(\beta_0)$$

We see that the discrete change can be non-zero even if $\beta_3 = 0$ and that the overall direction of the change depends on more than the sign of β_3 alone.

[34] *Proof:* Let us begin by defining $\Delta\pi/\Delta x$. This is equal to

$$\begin{aligned}
\frac{\Delta\pi}{\Delta x} &= \Pr(y = 1|x = 1, z) - \Pr(y = 1|x = 0, z) \\
&= F(\beta_0 + \beta_1 \times 1 + \beta_2 z + \beta_{12} \times 1 \times z) - F(\beta_0 + \beta_1 \times 0 + \beta_2 z + \beta_{12} \times 0 \times z) \\
&= F(\beta_0 + \beta_1 + \beta_2 z + \beta_{12} z) - F(\beta_0 + \beta_2 z)
\end{aligned}$$

Now we take the discrete change in $\Delta\pi/\Delta x$ with respect to Δz:

$$\begin{aligned}
\frac{\Delta^2\pi}{\Delta x \Delta z} &= \frac{F(\beta_0 + \beta_1 + \beta_2 z + \beta_{12} z) - F(\beta_0 + \beta_2 z)}{\Delta z} \\
&= F(\beta_0 + \beta_1 + \beta_2 \times 1 + \beta_{12} \times 1) - F(\beta_0 + \beta_2 \times 1) - \\
&\quad [F(\beta_0 + \beta_1 + \beta_2 \times 0 + \beta_{12} \times 0) - F(\beta_0 + \beta_2 \times 0)] \\
&= F(\beta_0 + \beta_1 + \beta_2 + \beta_{12}) - F(\beta_0 + \beta_2) - [F(\beta_0 + \beta_1) - F(\beta_0)]
\end{aligned}$$

By rearranging terms we obtain the desired formula.

This is reminiscent of the discussion of the marginal effects of interactions in the previous section.[35]

Odds Ratios

For the logit model, yet another method of interpretation is available: the odds ratio. In general, the odds are defined as the ratio of the probabilities of scoring 1 and 0 on the response variable, conditional on a set of covariates. Thus,

$$\Omega(\boldsymbol{x}) = \frac{\Pr(y=1|\boldsymbol{x})}{\Pr(y=0|\boldsymbol{x})} = \frac{\Pr(y=1|\boldsymbol{x})}{1-\Pr(y=1|\boldsymbol{x})} \tag{9.16}$$

In the logit model, it is easily demonstrated that[36]

$$\Omega(\boldsymbol{x}) = \exp(\boldsymbol{x}\boldsymbol{\beta}) \tag{9.17}$$

As the name suggests, the odds ratio is the ratio of two odds. These odds come about by changing the value of one of the covariates. To determine our thoughts, imagine that the value of the kth covariate is changed from x_k to $x_k + \delta$. Writing the latent regression model as $y_i^* = \beta_k x_k + \boldsymbol{x}_i\boldsymbol{\beta} + \epsilon_i$, where $\boldsymbol{x}_i$ contains all of the remaining predictors plus a constant term, the odds at x_k are

$$\Omega(x_k, \boldsymbol{x}_i) = \exp(\beta_k x_k + \boldsymbol{x}_i\boldsymbol{\beta}) = \exp(\beta_k x_k)\exp(\boldsymbol{x}_i\boldsymbol{\beta})$$

[35]For a more detailed discussion of discrete change involving dummy interactions see Ai and Norton (2003), Norton, Wang, and Ai (2004), and Mitchell and Chen (2005). Ai and Norton (2003) and Norton, Wang, and Ai (2004) also show formulas for the standard errors.

[36]*Proof:* In the logit model, $\Pr(y=1|\boldsymbol{x}) = \exp(\boldsymbol{x}\boldsymbol{\beta})/[1+\exp(\boldsymbol{x}\boldsymbol{\beta})]$. Moreover,

$$\begin{aligned} 1-\Pr(y=1|\boldsymbol{x}) &= 1 - \frac{\exp(\boldsymbol{x}\boldsymbol{\beta})}{1+\exp(\boldsymbol{x}\boldsymbol{\beta})} = \frac{1+\exp(\boldsymbol{x}\boldsymbol{\beta})-\exp(\boldsymbol{x}\boldsymbol{\beta})}{1+\exp(\boldsymbol{x}\boldsymbol{\beta})} \\ &= \frac{1}{1+\exp(\boldsymbol{x}\boldsymbol{\beta})} \end{aligned}$$

Consequently,

$$\begin{aligned} \Omega(\boldsymbol{x}) &= \frac{\frac{\exp(\boldsymbol{x}\boldsymbol{\beta})}{1+\exp(\boldsymbol{x}\boldsymbol{\beta})}}{\frac{1}{1+\exp(\boldsymbol{x}\boldsymbol{\beta})}} = \frac{\exp(\boldsymbol{x}\boldsymbol{\beta})}{1+\exp(\boldsymbol{x}\boldsymbol{\beta})}[1+\exp(\boldsymbol{x}\boldsymbol{\beta})] \\ &= \exp(\boldsymbol{x}\boldsymbol{\beta}) \end{aligned}$$

The odds at $x_k + \delta$ are

$$\Omega(x_k + \delta, \boldsymbol{x}_i) \;=\; \exp[\beta_k(x_k + \delta) + \boldsymbol{x}_i\boldsymbol{\beta}] = \exp(\beta_k x_k)\exp(\beta_k\delta)\exp(\boldsymbol{x}_i\boldsymbol{\beta})$$

The odds ratio is then defined as

$$\begin{aligned}\frac{\Omega(x_k + \delta, \boldsymbol{x}_i)}{\Omega(x_k, \boldsymbol{x}_i)} &= \frac{\exp(\beta_k x_k)\exp(\beta_k\delta)\exp(\boldsymbol{x}_i\boldsymbol{\beta})}{\exp(\beta_k x_k)\exp(\boldsymbol{x}_i\boldsymbol{\beta})} \\ &= \exp(\beta_k\delta) \qquad (9.18)\end{aligned}$$

The interpretation of (9.18) is that, for a change of δ in x_k, the odds are expected to change by a factor of $\exp(\beta_k\delta)$.

Compared to the other interpretative devices that we have discussed, the nice aspect of the odds ratio is that the other covariates are irrelevant. No matter how we set those covariates, the odds ratio for x_k is always equal to $\exp(\beta_k\delta)$. Moreover, the odds ratio can be easily computed from the coefficients, especially when $\delta = 1$.[37] On the downside, some people have a difficult time wrapping their brain around the concept of odds.

Two transformations of the odds ratio are useful. First, the **log-odds ratio** is simply the natural logarithm of the odds ratio: $\ln[\exp(\beta_k\delta)] = \beta_k\delta$. As this equation shows, the log-odds ratio is a linear function and, as such, interpretation is straightforward: for an increase of δ units in x_k the log-odds are expected to increase by $\beta_k\delta$, where the log-odds are defined as $\ln[\pi/(1-\pi)]$, where π is the probability of scoring 1 on the response variable.[38]

A second useful transformation is to compute the percentage change in the odds:

$$100 \times \frac{\Omega(x_k + \delta, \boldsymbol{x}_i) - \Omega(x_k, \boldsymbol{x}_i)}{\Omega(x_k, \boldsymbol{x}_i)} \;=\; 100 \times [\exp(\beta_k\delta) - 1]$$

This can be interpreted as the percentage change in the odds for a δ unit change in x_k.[39]

[37] In this case, the odds ratio becomes $\exp(\beta_k)$. Long (1997) refers to this as the *factor change*.

[38] As we have seen in Chapter 4, $\ln[\pi/(1-\pi)]$ is also known as the logit. In the logit model, $\ln[\pi/(1-\pi)] = \ln[\exp(\boldsymbol{x}_i\boldsymbol{\beta})] = \boldsymbol{x}_i\boldsymbol{\beta}$. The fact that the logit is a linear function of the covariates is what gives the logit model its name.

[39] We obtain this formula by recognizing that $\Omega(x_k + \delta, \boldsymbol{x}_i) - \Omega(x_k, \boldsymbol{x}_i) = \exp(\beta_k\delta)\exp(\beta_k x_k)\exp(\boldsymbol{x}_i\boldsymbol{\beta}) - \exp(\beta_k x_k)\exp(\boldsymbol{x}_i\boldsymbol{\beta}) = \exp(\beta_k x_k)\exp(\boldsymbol{x}_i\boldsymbol{\beta})[\exp(\beta_k\delta) - 1]$. Division of this expression by $\Omega(x_k, \boldsymbol{x}_i) = \exp(\beta_k x_k)\exp(\boldsymbol{x}_i\boldsymbol{\beta})$ yields $\exp(\beta_k\delta) - 1$. All that is left to do then is to multiply by 100.

Table 9.2: Correct Prediction in Logit and Probit Models

	y		
$\hat{y}$	0	1	Total
0	n_{00}	n_{01}	$n_{0.}$
1	n_{10}	n_{11}	$n_{1.}$
Total	$n_{.0}$	$n_{.1}$	n

9.2.4 Model Fit

Correct Predictions and ROC

One measure of how well logit and probit models fit the data is to consider the number of correct predictions or classifications that they generate. The approach here is to predict an individual's score on y based on his or her predicted probability. A common cutoff for the predicted probability is .5 so that

$$\begin{aligned} \hat{y} &= 1 \quad \text{if } \hat{\pi} > .5 \\ \hat{y} &= 0 \quad \text{if } \hat{\pi} \leq .5 \end{aligned}$$

where $\hat{y}$ is the predicted score on the response variable. We can then compare the predictions with the actual scores on the response variable. Whenever there is agreement between $\hat{y}$ and y, there is a correct prediction. The higher the percentage of correctly classified observations, the better the model fit is.

The mechanics of computing the percentage of correct predictions can be illustrated using Table 9.2. In this table, the column variable indicates the actual value of the response variable, while the row variable indicates the predicted value. The quantities n_{00} and n_{11} represent the number of cases where the predictions correspond to the actual value of the response variable. These are, then, the correctly classified cases. The percentage of correctly classified (CC) cases is given by

$$CC \quad = \quad 100 \times \frac{n_{00} + n_{11}}{n}$$

In judging CC, it is important to consider how good our predictions would have been in the absence of the model. If we were to ignore all knowledge

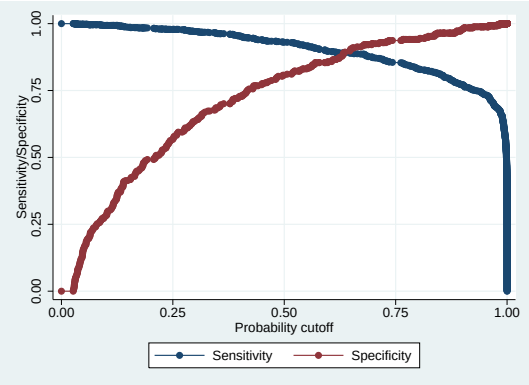

Figure 9.6: Sensitivity, Specificity, and Probability Cutoffs

about the covariates, then the best prediction of the response variable would be the mode. (This is the best prediction in the sense of minimizing the number of misclassified cases.) If CC is much higher than the percentage of cases in the modal category, then our model clearly seems to add to our ability to predict the response variable. On the other hand, if CC is in the vicinity of the percentage of cases in the modal category, then we should question how much the model adds, even when CC seems reasonably high.

The literature on **receiver-operating-characteristics** or ROCs adds two other measures of classification, known as sensitivity and specificity.[40] Sensitivity is the conditional probability of $\hat{y} = 1$ given that $y = 1$. It is operationalized as $100 \times (n_{11}/n_{.1})$. Specificity is the conditional probability of $\hat{y} = 0$ given that $y = 0$. It is operationalized as $100 \times (n_{00}/n_{.0})$. These measures give an indication of the likelihood of correct "positives" and "negatives" in model predictions.

The traditional cutoff of $\hat{\pi} = .5$ is quite arbitrary. In the ROC literature, it is customary to manipulate the cutoff over the interval $[0, 1]$ and to plot sensitivity and specificity against the cutoff values, as is done in Figure 9.6. We see that sensitivity and specificity move in opposite directions as the probability cutoff is increased. This is because an increase in the cutoff classifies fewer and fewer sample units as 1, while more and more units are classified as 0. Inevitably. the conditional probability of correctly classifying cases that actually scored 1 drops precipitously when the cutoff increases, whereas the conditional probability of correctly classifying cases that actually

[40] See Green and Swets (1974) for a general discussion of ROC methods.

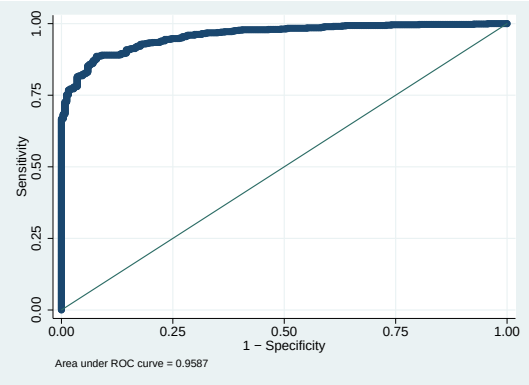

Figure 9.7: ROC Curve

scored 0 increases.[41]

A different approach to manipulating the cutoffs is to graph the ROC curve. This is a graph of sensitivity versus 1 - specificity for different probability cutoffs. Here sensitivity measures the incidence of correct positives (i.e. predictions that the dependent variable takes on the value 1), while 1 - specificity captures the incidence of false positives. Figure 9.7 gives an example of such a graph. The curve starts at the origin, which corresponds to a probability cutoff of 1. It ends at coordinate (1,1), which corresponds to a probability cutoff of 0. The more predictive power the model has, the more bowed the ROC curve is. As a measure of this predictive power, the area under the ROC curve is calculated. When the model has no predictive power, this area is .5, corresponding to the area below the 45 degree reference line that is shown in Figure 9.7. When the model fits the data perfectly, then the area under the ROC curve is 1. In this example, the area is computed as .96, which means that we have a model that fits the data quite nicely.

[41]It is enlightening to consider what happens at the extreme values of the probability cutoff. First, consider a cutoff of 0. This means that all units are classified as 1. Because of this, $n_{11} = n_{.1}$ and sensitivity is equal to 1. By the same token, $n_{00} = 0$ because no one is predicted to score 0, so that specificity is equal to 0. At the other extreme, a probability cutoff of 1 means that no sample unit is predicted to score 0 on the response variable. Now $n_{11} = 0$, which means that sensitivity is 0. On the other hand, all units that actually scored 0 were also classified as such: $n_{00} = n_{.0}$ so that specificity is 1.

Pseudo-R^2

A large number of pseudo-R^2 measures has been defined for the binary logit and probit models.[42] Several Monte-Carlo simulations have been performed on these measures (see Hagle and Mitchell 1992; Langer 2000; Veall and Zimmerman 1992, 1993, 1994; and Windmeijer 1995). Based on the simulation results, two measures stand out, namely those developed by McKelvey and Zavoina (1975) and (with a correction) Aldrich and Nelson (1984).

McKelvey & Zavoina Pseudo-R^2 The pseudo-R^2 measure proposed by McKelvey and Zavoina (1975) is based on the latent response variable y^*. The measure takes as the explained variance the variance in the predicted values $\hat{y}^*$. It then compares this to the total variation in y^*, just as one would do in a linear regression R^2. Thus,

$$R^2_{M\&Z} = \frac{\hat{V}[\hat{y}^*]}{\hat{v}[y^*]} = \frac{\hat{V}[\hat{y}^*]}{\hat{V}[\hat{y}^*] + V[\epsilon]} \tag{9.19}$$

Here $\hat{V}[\hat{y}^*] = \hat{\boldsymbol{\beta}}'\mathbf{V}[\boldsymbol{x}]\hat{\boldsymbol{\beta}}$.[43] By assumption, $V[\epsilon] = 1$ in the probit model and $V[\epsilon] = \pi^2/3$ in the logit model. The theoretical range is from 0 to 1. The upper limit of 1 is reached when the predictions of y^* are perfect. In this case, $\hat{y}^* = y^*$ and $V[\hat{y}^*] = V[y^*]$. The lower limit of 0 arises when all of the regression coefficients are zero with the exception of $\hat{\beta}_0$. In this case $V[\hat{y}^*] = \hat{\beta}_0^2 \times \sigma_{00}^2 = \hat{\beta}_0^2 \times 0 = 0$, where σ_{00}^2 is the variation in the constant, which is zero.

Corrected Aldrich & Nelson Pseudo-R^2 As we discussed in Chapter 8, the pseudo-R^2 measure proposed by Aldrich and Nelson (1984) is based on the likelihood ratio test statistic. Expanding on (8.4), we have for the

[42] Apart from the measures described in Chapter 8, the range of pseudo-R^2 measures for logit and probit includes measures proposed by Efron (1978), Goldberger (1973), Lave (1970), Morrison (1972), and Neter and Maynes (1970). A review of these measures can be found in Veall and Zimmermann (1996).

[43] It may appear this is at odds with the earlier result that $V[\boldsymbol{x}\hat{\boldsymbol{\beta}}] = \boldsymbol{x}\mathbf{V}[\hat{\boldsymbol{\beta}}]\boldsymbol{x}'$. However, this is not the case. Earlier we were trying to obtain the variability of the predictions across samples. Now we are interested in obtaining the variation in the predictions in our particular sample. These are two different goals and they produce different estimators.

probit model

$$\begin{aligned} R^2_{A\&N-P} &= \frac{LR}{LR+n} \\ &= \frac{2(\ell_1-\ell_0)}{2(\ell_1-\ell_0)+n} \end{aligned} \tag{9.20}$$

where ℓ_1 is the log-likelihood of a model with covariates and ℓ_0 is the log-likelihood of a constant-only model (i.e. a model without covariates). For the logit model, the sample size is re-scaled by a factor of $\pi^2/3$, in order to accommodate the different variance of this model. Thus,

$$R^2_{A\&N-L} = \frac{2(\ell_1-\ell_0)}{2(\ell_1-\ell_0)+\frac{\pi^2}{3}n}$$

Here, the log-likelihood of the constant-only model is given by[44]

$$\ell_0 = n_0 \ln\left(\frac{n_0}{n}\right) + n_1 \ln\left(\frac{n_1}{n}\right)$$

The lower-bound on $R^2_{A\&N-P}$ and $R^2_{A\&N-L}$ is 0, which occurs when $\ell_1 = \ell_0$, i.e. when the covariates do not contribute to the fit of the model. The theoretical upper-bound of $R^2_{A\&N-P}$ is equal to $-2\ell_0/(-2\ell_0+n)$, which arises when $\mathcal{L}_1 = 1$ or $\ell_1 = 0$. For the logit model, this theoretical upper-bound is $-2\ell_0/(-2\ell_0+3.29n)$, where 3.29 is the approximate value of $\pi^2/3$. This is the theoretical upper-bound because $\mathcal{L}_1 = 1$ only when the model is saturated, i.e. we have as many parameters as observations. In practice, this is never the case with logit and probit models.

The discussion so far suggests that the upper-bound of the Aldrich-Nelson pseudo-R^2 measures can be quite far removed from 1, which is the upper bound of the familiar regression R^2. In order to standardize the range of $R^2_{A\&N-P}$ and $R^2_{A\&N-L}$, Veall and Zimmermann (1992) have proposed the

[44] In the constant only model, β_0 is adjusted so as to accommodate the proportions of 0s and 1s in the data. The proportion of 1s is the estimate of π, while the proportion of 0s is the estimate of $1-\pi$. Letting n_1 and n_0 denote the numbers of cases that score 1 and 0, respectively, the estimate of π is equal to n_1/n, while the estimate of $1-\pi$ is equal to n_0/n. Since n_1/n occurs n_1 times in the data, the contribution to the log-likelihood function is $n_1 \times (n_1/n)$. Similarly, since n_0/n occurs n_0 times in the data, the contribution to ℓ is equal to $n_0 \times (n_0/n)$. Adding these two components gives the formula for ℓ.

following corrections:

$$R^2_{V\&Z-P} = \frac{\frac{2(\ell_1-\ell_0)}{2(\ell_1-\ell_0)+n}}{\frac{-2\ell_0}{-2\ell_0+n}} = \frac{2(\ell_1-\ell_0)}{2(\ell_1-\ell_0)+n}\frac{-2\ell_0+n}{-2\ell_0}$$

$$R^2_{V\&Z-L} = \frac{\frac{2(\ell_1-\ell_0)}{2(\ell_1-\ell_0)+3.29n}}{\frac{-2\ell_0}{-2\ell_0+3.29n}} = \frac{2(\ell_1-\ell_0)}{2(\ell_1-\ell_0)+3.29n}\frac{-2\ell_0+3.29n}{-2\ell_0}$$

These measures have a theoretical range of 0 to 1. In simulation studies, they performed rather well, in the sense of approximating the major properties of the linear regression R^2 (see Hagle and Mitchell 1992; Langer 2000; Veall and Zimmerman 1992, 1993, 1994; and Windmeijer 1995).

9.2.5 Residuals*

Residuals are a useful way to detect problems with model fit and influential data points. Where the residual is large, the model fits the observation poorly. Observations with large residuals are known as outliers. Such observations may also exert a large influence on the parameter estimates, in which case the observations are influential data points. However, it is important to keep in mind that not every outlier is an influential data point. Much depends, as well, on how atypical the observation is with respect to the covariates.

In linear regression analysis, the residual is defined as the discrepancy between the observed and predicted values: $e_i = y_i - \hat{y}_i$. In logit and probit models, $\hat{y}_i = \hat{\pi}_i$ so that the residual may be defined as $y_i - \hat{\pi}_i$. However, when defined in this manner, the residuals are not homoskedastic, since $V[y_i - \pi_i] = \pi_i(1-\pi_i)$.[45] One solution is to compute the **Pearson residual**:

$$r_i = \frac{y_i - \hat{\pi}_i}{\sqrt{\hat{\pi}_i(1-\hat{\pi}_i)}}$$

The sum of the squared Pearson residuals is the Pearson χ^2 fit statistic, which is defined as $\chi^2 = \sum_{i=1}^{n} r_i^2$. This can be considered a measure of the overall fit of the model in as far as it aggregates the prediction failures of the model.

[45] *Proof:* By definition, $V[y_i - \pi_i] = (0-\pi_i)^2(1-\pi_i) + (1-\pi_i)^2\pi_i$, where $(0-\pi_i)^2$ and $(1-\pi_i)^2$ are the squared deviations from the mean of $y_i - \pi_i$, which is π_i. Expansion gives the result that $V[y_i - \pi_i] = \pi_i(1-\pi_i)$.

While the Pearson residual is useful, Pregibon (1981) demonstrated that it is not quite homoskedastic, as one might have expected. He proposed using a **standardized Pearson residual**, which is given by

$$r_i^* = \frac{r_i}{\sqrt{1-h_{ii}}}$$

where

$$h_{ii} = \hat{\pi}_i(1-\hat{\pi}_i)\boldsymbol{x}_i\mathbf{V}[\hat{\boldsymbol{\beta}}]\boldsymbol{x}'_i$$

is the hat value. This residual has the property that $V[r_i^*] = 1$ and, as such, it is homoskedastic. Large values of the standardized Pearson residual indicate a lack of fit, although Hosmer and Lemeshow (1989) warn that it is impossible to provide a hard and fast rule about the cutoff value beyond which one would consider data points to be outliers.

As noted previously, data points that are outliers are not necessarily influential. In linear regression analysis, influence is ascertained in a number of different ways, including changes in the estimated coefficients and changes in fit. Pregibon (1981) showed that the change in coefficients can be approximated via

$$\Delta\hat{\boldsymbol{\beta}}_i = \frac{r_i^2 h_{ii}}{(1-h_{ii})^2}$$

where $\Delta\hat{\boldsymbol{\beta}}_j$ is the standardized difference between $\hat{\boldsymbol{\beta}}$ and $\hat{\boldsymbol{\beta}}_{-i}$, where the first term represents the ML estimates in the entire sample and the second term represents the ML estimates after the ith observation has been removed. This is essentially the equivalent of Cook's (1977) distance in linear regression analysis. The change in model fit is given by

$$\Delta\chi_i^2 = \frac{r_i^2}{1-h_{ii}}$$

This is the change in the Pearson χ^2 statistic when we compare the fit in the entire sample versus the fit in the sub-sample that comes about by dropping the ith observation.

9.2.6 Example

Estimation Results

Let us return to the voting behavior example from Chapter 9.1. Table 9.3 shows the logit and probit results from this example.[46] These results were obtained by running the following commands:

```
logit vote pid male white bornagain age educ hhinc traitdif
issuedif

probit vote pid male white bornagain age educ hhinc traitdif
issuedif
```

On the average, the logit coefficients are about 1.74 times larger than the probit coefficients, as should be expected given the different scaling of the parameters. Other than this, the models produce similar results, as should also be expected. Thus, all predictors have effects of the same sign in the logit and probit specifications and the same predictors are statistically significant.

At the final iteration, the value of the log-likelihood function is -15.067 for the logit model and -14.801 for the probit model. (Both models require only 9 iterations to converge.) In both models, we can soundly reject the null hypothesis that none of the predictors matter for vote choice, as is evidenced by the large values of the Wald χ^2 test statistics. The model fit is excellent. To obtain McKelvey and Zavoina's (1975) pseudo-R^2 I ran the

```
fitstat
```

command developed by J. Scott Long and Jeremy Freese. I computed Veall and Zimmermann's (1992) pseudo-R^2 measure by hand, although `fitstat` provides all of the necessary ingredients, including ℓ_0 which, in this case, is -98.300.

As another indicator of the fit, we can consider the percentage of correct classifications. This can be obtained by issuing the

```
estat class
```

[46] The example does not show the confidence intervals. By default, Stata reports 95% Wald confidence intervals. For the logit model, likelihood-based confidence intervals can be obtained using the `logprof` command developed by Mark Pearce.

Table 9.3: Logit and Probit Models of Vote Choice in 2000

	Logit		Probit	
Predictor	**Estimate**	**S.E.**	**Estimate**	**S.E.**
Partisanship	$-.997^{**}$	.350	$-.581^{**}$	.203
Male	.229	1.101	.153	.638
White	$-.236$	1.462	$-.117$	.856
Born Again Christian	-2.835^{*}	1.286	-1.594^{*}	.720
Age	$-.045$	.035	$-.025$	.020
Education	.197	.483	.116	.279
Household Income	$-.232$	.148	$-.136$	.088
Trait Differential	5.730^{**}	1.967	3.273^{**}	1.093
Issue Differential	1.667	1.809	.980	1.060
Constant	6.446^{*}	3.279	3.665^{*}	1.859
ℓ	-15.067		-14.801	
Wald χ^2	166.470		167.000	
p	.000		.000	
$R^2_{M\&Z}$	.964		.967	
$R_{V\&Z}$	.887		.931	

Notes: $n = 142$. $^{**}p < .01$, $^{*}p < .05$ (two-tailed). In the logit model, 1 failure and 2 successes are completely determined. In the probit model, 26 failures and 20 successes are completely determined. Models estimated using Stata's `logit` and `probit` commands.

command after estimating a logit or probit model. The percentage of correct predictions is 93.7% for both models, which can be considered very high.[47]

Interpretation

Considering the results, we can say impressionistically that the probability of a Gore vote decreases as partisanship moves toward the Republican end of the scale and as people indicate that they are born-again Christians. The likelihood of a Gore vote increases as trait judgments of him are more favorable than those for Bush. However, beyond these rather vague statements, nothing can be said about the effects based on the parameter estimates alone. To provide further interpretation of the results we have to rely on the interpretation methods described earlier. Here, I shall illustrate those methods by focusing on the interpretation of the effects of the trait differential and born-again Christianity.

Marginal Effects and Elasticities To obtain marginal effects at the mean in Stata, one should issue the

```
mfx [compute], [eyex]
```

command. Issuing `mfx` by itself will compute marginal effects at the mean for continuous predictors and discrete changes in the predicted probability for dummy predictors. Adding the `eyex` option produces elasticities. The top panel of Table 9.4 shows the marginal effect at the mean of the trait differential, including standard errors (computed via the delta method), test statistics, and elasticities. We observe a statistically significant marginal positive effect for the trait differential. The size of this effect is considerable: the instantaneous change in predicted probability is 1.234 in the logit model and 1.186 in the probit model. Looking at the elasticities, we observe that a 1% increase in the trait differential, which corresponds to .06 units, produces a .083% increase in the probability of voting for Gore in the logit model and a .076% increase in the probit model.

Average marginal effects can be obtained by running Tamár Bartus' `margeff` command. The syntax for this command is

[47] The ROC curve also suggests an excellent fit. Stata's `lroc` command, which is issued after the `logit` and `probit` commands, reveals that the area under the ROC curve is .99 for the logit and probit models. This is very close to the upper-bound of 1.0.

Table 9.4: Marginal Effect of the Trait Differential on Vote Choice

(a) Marginal Effect at the Mean					
Method	**Effect**	**S.E.**	**z**	**p**	**Elasticity**
Logit	1.234	.438	2.820	.000	.083
Probit	1.186	.392	3.030	.000	.076

(b) Average Marginal Effect					
Method	**Effect**	**S.E.**	**z**	**p**	**Elasticity**
Logit	.189	.041	4.650	.000	NA
Probit	.186	.039	4.800	.000	NA

Notes: Marginal effects at the mean were computed using Stata's `mfx` command. Average marginal effects were computed using the `margeff` command.

```
margeff [compute], [dummies(varlist)]
```

The second panel in Table 9.3 shows these marginal effects, which are quite a bit different from the marginal effects at the mean but still statistically significant and positive.

Predicted Probabilities and Discrete Change in Predicted Probabilities Another way of interpreting the results is by considering predicted probabilities. These can be obtained using a number of the commands in the Stata add-on `SPost`, written by J. Scott Long and Jeremy Freese. First, it is possible to obtain a table of predicted probabilities using the `prtab` command:

```
prtab rowvar [colvar] [supercolvar], [by(superrowvar)]
[rest(stat)]
```

The table can display predicted probabilities based on up to four covariates; the minimum is to specify one covariate (the *rowvar*). The option `rest(stat)` allows one to specify values for the remaining covariates. For example, in our analysis we might be interested in predicted probabilities by partisanship and born-again Christianity, while holding everything else constant. We can obtain these predicted probabilities by issuing:

Table 9.5: Predicted Probability of Voting for Gore by Partisanship and Religious Beliefs

	Born-Again Christian?	
Pid	**No**	**Yes**
Strong Democrat	.970	.615
Weak Democrat	.904	.386
Leaning Democrat	.765	.192
Independent	.557	.073
Leaning Republican	.331	.021
Weak Republican	.154	.005
Strong Republican	.055	.001

Notes: Predicted probabilities were computed using the `prtab` command while holding all other predictors at their mean. Predicted probabilities are based on probit estimation results.

```
prtab pid bornagain, rest(mean)
```

The resulting predicted probabilities are shown in Table 9.4 for the probit model. The table reveals a much faster drop-off in the likelihood of voting for Gore among born-again Christians than among other respondents. While strong Democrats have a better than .5 predicted probability of voting for Gore, regardless of their religious beliefs, weak Democrats who are born-again Christians are already more likely to vote for Bush than for Gore. This suggests a rather powerful effect of born-again Christianity.

Another useful command in the `SPost` suite is the `prgen` command, which allows one to compute and graph predicted probabilities.[48] The relationship between partisanship and the probability of voting for Gore, stratified by religious beliefs, can be plotted using the following command sequence:

```
prgen pid, from(0) to(6) gen(pidnot) x(bornagain=0) rest(mean)
n(7)
```

[48]In logit analysis, predicted probabilities can also be graphed using Mitchell and Chen's (2005) tools for visualizing binary logit models (`vibl`).

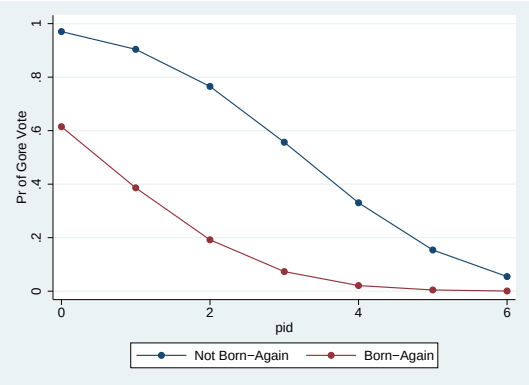

Figure 9.8: Predicted Probability of Voting for Gore by Partisanship and Religious Beliefs

```
prgen pid, from(0) to(6) gen(pidba) x(bornagain=1) rest(mean)
n(7)

label var pidnop1 "Not Born-Again"

label var pidbap1 "Born-Again"

graph twoway connected pidnop1 pidbap1 pidbax, ytitle("Pr of
Gore Vote") xtitle("pid")
```

Here `pidnop1` contains the predicted probabilities of a Gore vote for the different levels of partisanship for respondents who are not born-again Christians; `pidbap1` contains similar predicted probabilities but now for born-again Christians. Figure 9.8 shows the graph of the predicted probability, clearly revealing the differential drop-offs in the likelihood of a Gore vote among born-again and other respondents.

A third useful command in the `SPost` suite is the `prvalue` command, which allows one to compute predicted probabilities given certain values of the covariates. The general syntax for the command is

```
prvalue, [x(variable1=value1 [...]) rest(stats) delta save dif
lev(#)]
```

The `save` option preserves the predicted probability for subsequent comparisons, while the `dif` option draws the comparison between the current value of the predicted probability and that stored using the `save` option. The

`delta` option causes Stata to compute confidence intervals on the predicted probabilities using the delta method. The confidence level of these intervals is controlled by `lev(#)`. For example, one can issue the following sequence after running a probit analysis:

```
prvalue, x(bornagain=0) rest(mean) delta save lev(99)

prvalue, x(bornagain=1) rest(mean) delta dif lev(99)
```

The first command causes Stata to compute the predicted probability of a Gore vote and the 99% confidence interval for respondents who are not born-again Christians. This predicted probability is .571, although it has a rather wide confidence interval that runs from .202 to .941. The second command causes Stata to calculate the predicted probability of a Gore vote for born-again Christians. It also computes the difference between this predicted probability and that for the previous group, as well as the 99% confidence interval on the difference. For born-again Christians, the predicted probability of a Gore vote is only .079. Compared to the earlier predicted probability, this is a change of -.493 points. The confidence interval on this difference runs from -.928 to -.057. Since this interval does not include 0, we can say that the difference in predicted probabilities due to born-again Christianity is statistically significant at the .01 level.

Finally, Long and Freese's `prchange` command computes discrete change in the predicted probabilities. By just typing

```
prchange
```

Stata will produce a number of the discrete change measures that we have discussed, including maximum possible change, a change from 0 to 1, a centered unit change, and a standard deviation change. In computing these measures, the default is to keep the remaining predictors at their mean values. Table 9.6 reports these measures for partisanship, the trait differential, and born-again Christianity. One should keep in mind, that only maximum possible change and 0-1 change are reasonable for dummy predictors; these measures produce identical estimates of the discrete change. Looking at the discrete changes, we see sizable effects for all predictors. For example, moving PID from the minimum to the maximum produces a drop in the predicted probability of a Gore vote of .884 points. Moving the trait differential from one standard deviation below to one standard deviation above the mean produces a boost in the predicted probability of a Gore vote of .860 points. And

Table 9.6: Discrete Change in the Predicted Probability of a Gore Vote

Predictor	Min → Max	0 → 1	± .5	± .5 SD
PID	−.884	−.144	−.208	−.464
Trait Differential	1.000	.691	.866	.860
Born-Again Christian	−.493	−.493	NA	NA

Notes: Discrete changes in predicted probabilities were computed using the `prchange` command while holding all other predictors at their mean. These changes are based on probit estimation results.

compared to others, born-again Christians are almost half a point less likely to vote for Gore, holding all else equal.

The Odds Ratio For the logit model, the odds ratio forms an alternative basis for interpretation. Long and Freese's `SPost` suite contains a useful command that will compute odds ratios and factor/percentage changes in those ratios. The syntax of this command is as follows:

```
listcoef varlist, [factor|percent] [help]
```

The `factor` option shows the factor changes in the odds for a unit increase in the predictor, whereas the `percent` option shows the percentage changes in the odds. The `help` option adds annotation to the output so that it can be read more easily.

Table 9.7 shows the factor and percentage changes in the odds for PID, the trait differential, and born-again Christianity. The factor changes for PID and born-again Christian are less than 1 because the corresponding logit coefficients are negative. The factor change for the trait differential is enormous because of the very large coefficient on this predictor. The percentage changes indicate that for each increase in PID, the odds of voting for Gore drop by 63.1%. Compared to others, the odds of voting for Gore are 94.1% lower for born-again Christians. An increase by one unit in the trait differential, which is an increase in favor of Gore, boosts the odds of a Gore vote by over 30,000%.

Table 9.7: Odds Ratios for the Gore Vote

Predictor	Factor	Percent
PID	.369	−63.100
Trait Differential	307.845	30684.500
Born-Again Christian	.059	−94.100

Notes: Factor and percentage changes in the odds were computed using the `listcoef` command. These changes are based on logit estimation results.

9.2.7 Heteroskedastic Logit and Probit

Derivation, Estimation, and Interpretation

By assumption, logit and probit models have homoskedastic stochastic components. But what happens when this assumption is wrong? What if the stochastic elements of choice are heteroskedastic? In this case, we have a problem that is far more serious than would be the case in linear regression analysis. In linear regression, the OLS estimator is still unbiased in the presence of heteroskedastic errors. It is no longer the *best* linear unbiased estimator and the estimated standard errors will be biased, but the regression coefficients themselves can be trusted. This is not true in logit and probit models. As we have seen, the variance assumption in these models plays a critical role in setting the scale of the coefficients. We can transform that scale by setting the variance differently, as is the case when we compare logit and probit. As long as we apply the same transformation to all sample units, then we can still compare predicted probabilities and marginal effects across those units. But if the appropriate variance is different for some units than other units, then imposing a common variance and setting the scale of the coefficients accordingly will introduce bias. This bias does not disappear asymptotically, so that the ML estimators will be inconsistent, as well as asymptotically inefficient. In addition, one also has the problem that the estimated standard errors will be inconsistent.

There is a way around the imposition of homoskedasticity and that is to run a heteroskedastic logi/probit analysis. Heteroskedastic logit and probit are patterned after Harvey's (1976) heteroskedastic regression model. Here,

I focus on the heteroskedastic probit model, since it is implemented through Stata's `hetprob` command.

Derivation of the heteroskedastic probit model begins by specifying the following latent variable model:

$$\begin{aligned} y_i^* &= \boldsymbol{x}_i\boldsymbol{\beta} + \epsilon_i \\ \epsilon_i &\sim N(0, \sigma_i^2) \end{aligned}$$

Compared to the standard probit model, the difference here lies in the variance assumption for ϵ_i: in probit, this variance is fixed to 1 for all units, but here it is left free to vary across units. We model the variance as a function of a set of predictors, analogous to Harvey (1976):

$$\sigma_i^2 = \left[\exp\left(\boldsymbol{z}_i\boldsymbol{\gamma}\right)\right]^2$$

or, equivalently, $\sigma_i = \exp\left(\boldsymbol{z}_i\boldsymbol{\gamma}\right)$ or $\ln\sigma_i = \boldsymbol{z}_i\boldsymbol{\gamma}$. Here $\boldsymbol{z}_i$ may or may not contain the same predictors as $\boldsymbol{x}_i$. Importantly, $\boldsymbol{z}_i$ does *not* contain a constant, which means that if none of the predictors in the variance model have an effect, then $\sigma_i^2 = \exp(0) = 1$, the same as in the standard probit model. This suggests a test of whether a heteroskedastic model is necessary, namely a test of $H_0 : \boldsymbol{\gamma} = \mathbf{0}$.

It is possible to transform the heteroskedastic probit specification into a homoskedastic model by dividing all terms of the latent variable model by σ_i. This produces

$$\begin{aligned} \frac{y_i^*}{\sigma_i} &= \frac{\boldsymbol{x}_i\boldsymbol{\beta}}{\sigma_i} + \frac{\epsilon_i}{\sigma_i} \\ \frac{y_i^*}{\exp\left(\boldsymbol{z}_i\boldsymbol{\gamma}\right)} &= \frac{\boldsymbol{x}_i\boldsymbol{\beta}}{\exp\left(\boldsymbol{z}_i\boldsymbol{\gamma}\right)} + \frac{\epsilon_i}{\exp\left(\boldsymbol{z}_i\boldsymbol{\gamma}\right)} \\ \frac{y_i^*}{\exp\left(\boldsymbol{z}_i\boldsymbol{\gamma}\right)} &= \frac{\boldsymbol{x}_i\boldsymbol{\beta}}{\exp\left(\boldsymbol{z}_i\boldsymbol{\gamma}\right)} + \delta_i \end{aligned}$$

In this model, $\delta_i \sim N(0, 1)$ so that the errors are homoskedastic again.[49]

We can now state the mechanism linking the observed response to the underlying latent variable. In the ordinary probit model, this mechanism is $\pi_i = \Pr(y_i = 1) = \Pr(y_i^* > 0) = \Pr(\epsilon_i < \boldsymbol{x}_i\boldsymbol{\beta})$. In terms of the transformed

[49] *Proof:* $V[\delta_i] = \mathcal{E}[\delta_i^2] = \left[\exp\left(\boldsymbol{z}_i\boldsymbol{\gamma}\right)\right]^{-1} V[\epsilon_i] = \left[\exp\left(\boldsymbol{z}_i\boldsymbol{\gamma}\right)\right]^{-1} \exp\left(\boldsymbol{z}_i\boldsymbol{\gamma}\right) = 1.$

latent variable this becomes

$$\begin{aligned}\pi_i &= \Pr\left(\frac{y_i^*}{\exp(\boldsymbol{z}_i\boldsymbol{\gamma})} > 0\right) \\ &= \Pr\left(\delta_i < \frac{\boldsymbol{x}_i\boldsymbol{\beta}}{\exp(\boldsymbol{z}_i\boldsymbol{\gamma})}\right) \\ &= \Phi\left(\frac{\boldsymbol{x}_i\boldsymbol{\beta}}{\exp(\boldsymbol{z}_i\boldsymbol{\gamma})}\right) \end{aligned} \tag{9.21}$$

The log-likelihood function for the heteroskedastic probit model is given by

$$\ell = \sum_{i=1}^{n}\left\{y_i \ln\Phi\left(\frac{\boldsymbol{x}_i\boldsymbol{\beta}}{\exp(\boldsymbol{z}_i\boldsymbol{\gamma})}\right) + (1-y_i)\ln\Phi\left(\frac{\boldsymbol{x}_i\boldsymbol{\beta}}{\exp(\boldsymbol{z}_i\boldsymbol{\gamma})}\right)\right\}$$

This is a difficult log-likelihood function to optimize and convergence problems are not uncommon. The first-order conditions are given by

$$\begin{aligned}\frac{\partial\ell}{\partial\boldsymbol{\beta}'} &= \sum_{i=1}^{n}\left[\frac{\phi_i(y_i-\Phi_i)}{\Phi_i(1-\Phi_i)}\right]\exp(\boldsymbol{z}_i\boldsymbol{\gamma})\,\boldsymbol{x}_i = \mathbf{0} \\ \frac{\partial\ell}{\partial\boldsymbol{\gamma}'} &= \sum_{i=1}^{n}\left[\frac{\phi_i(y_i-\Phi_i)}{\Phi_i(1-\Phi_i)}\right]\exp(\boldsymbol{z}_i\boldsymbol{\gamma})\,\boldsymbol{z}_i\,(-\boldsymbol{x}_i\boldsymbol{\beta}) = \mathbf{0}\end{aligned}$$

where $\phi_i = \phi\left[\boldsymbol{x}_i\boldsymbol{\beta}/\exp(\boldsymbol{z}_i\boldsymbol{\gamma})\right]$ and $\Phi_i = \Phi\left[\boldsymbol{x}_i\boldsymbol{\beta}/\exp(\boldsymbol{z}_i\boldsymbol{\gamma})\right]$.

Interpretation is also more complicated than in the standard probit model, at least when $\boldsymbol{z}_i$ and $\boldsymbol{x}_i$ share elements in common. Let w_{ik} be a predictor that could be included in $\boldsymbol{x}_i$, in $\boldsymbol{z}_i$, or in both, then the marginal effect of this predictor is given by

$$\frac{\partial\pi_i}{\partial w_{ik}} = \phi\left[\frac{\boldsymbol{x}_i\boldsymbol{\beta}}{\exp(\boldsymbol{z}_i\boldsymbol{\gamma})}\right]\frac{\beta_k - (\boldsymbol{x}_i\boldsymbol{\beta})\,\gamma_k}{\exp(\boldsymbol{z}_i\boldsymbol{\gamma})}$$

When w_{ik} appears in only $\boldsymbol{x}_i$, then $\gamma_k = 0$ and the formula for the marginal effect is adjusted accordingly. When w_{ik} appears in only $\boldsymbol{z}_i$, then $\beta_k = 0$ and the formula for the marginal effect is adjusted accordingly. When w_{ik} appears in both sets of covariates, then the marginal effect may deviate dramatically from what is suggested by either $\hat{\beta}_k$ or $\hat{\gamma}_k$. In general, the presence of $\exp(\boldsymbol{z}_i\boldsymbol{\gamma})$ can create important differences between the estimated coefficients and the marginal effects, for example in terms of significance levels. Thus, extreme care should be taken with the interpretation.

Table 9.8: Heteroskedastic Probit Model of Vote Choice in 2000

Predictor	Estimate	S.E.
Model for π		
Partisanship	$-.060^{+}$	.035
Male	$-.000$	.053
White	$-.185$	.140
Born Again Christian	$-.123$	.082
Age	$-.002$	.002
Education	.013	.022
Household Income	$-.011$	.009
Trait Differential	$.396^{+}$	.210
Constant	.430	.292
Variance Model		
Strength of Partisanship	$-.591^{**}$	.230
Education	$-.185^{+}$	.112
ℓ	-58.825	
LR test/p	17.790	.000

Notes: $n = 348$. $^{**}p < .01$, $^{+}p < .10$ (two-tailed). Estimates obtained using Stata's `hetprob` command. The likelihood ratio tests $H_0 : \boldsymbol{\gamma} = \mathbf{0}$.

Example

It is possible to add an uncertainty component to the 2000 presidential vote model that we explored earlier. This component incorporates the idea that vote choice may be more predictable for certain individuals than others. For instance, the vote choice of strong partisans should be more predictable than that of weak partisans or independents. Similarly, if we treat education as a proxy for information, then it stands to reason that more highly educated citizens are more predictable in their vote choices than less educated citizens.

To add the uncertainty component, we run Stata's `hetprob` command, whose syntax is:

```
hetprob depvar [ indepvars ], het(varlist) [ options ]
```

In our case, the dependent variable is vote choice (1=Gore, 0=Bush). Our vector $\boldsymbol{x}_i$ consists of partisanship, male, white, born-again Christian, age,

Table 9.9: Marginal Effects for Heteroskedastic Probit

Predictor	Effect	S.E.
Partisanship	$-.183^{**}$	.041
Male	$-.001$	.163
White	$-.370^{**}$	.149
Born Again Christian	$-.359^{*}$	.166
Age	$-.005$	.005
Education	.036	.066
Household Income	$-.033^{+}$	.018
Trait Differential	1.215^{**}	.280
Strength of Partisanship	$-.012$	.046

Notes: $^{**}p < .01$, $^{+}p < .10$ (two-tailed). Marginal effects are computed at the mean using the `mfx` command. Standard errors are computed using the delta method.

education, household income, and the trait differential describe earlier.[50] Our vector $\boldsymbol{z}_i$ consists of partisan strength (the absolute value of the difference between partisanship and the midpoint on the party identification scale) and education. The estimates are shown in Table 9.8 and the marginal effects at the mean, obtained by running `mfx`, are shown in Table 9.9. Note that Stata estimates $\ln \sigma_i^2$ rather than σ_i^2.[51] Negative values for the effects of the predictors in $\boldsymbol{z}_i$ mean that the variance (uncertainty) is reduced as the values of the predictors increase, whereas positive effects imply that the variance is increased.

Table 9.8 includes the result of a LR test of the hypothesis $H_0 : \boldsymbol{\gamma} = \mathbf{0}$. The value of the LR test statistic is 17.79, leading to a sound rejection of the hypothesis. This implies that the standard probit specification with $\sigma^2 = 1$ is inappropriate for the data at hand. Considering the marginal effects, we observe significant marginal effects for partisanship, white, born-

[50] The issue differential was dropped from the model because inclusion of this predictor reduced the sample size dramatically, leading to convergence problems for the heteroskedastic probit model.

[51] From a computational perspective, it is better to estimate $\ln \sigma_i^2$ since it is not bounded, whereas σ_i^2 is bounded to be non-negative. Thus issues of out-of-bound estimates or estimation at the boundary of parameter space do not arise with $\ln \sigma_i^2$.

again Christian, income, and the trait differential. Note that we obtain these effects, even though several of these predictors were not statistically significant in Table 9.8.

Some Cautionary Remarks

Heteroskedastic logit and probit models have been popularized in political science through the work of Alvarez and Brehm (1995, 2002). While interesting things may be learnt from these models, it is important to remember that they tinker with a fundamental identifying assumption of logit and probit analysis. There is a certain risk involved in this, as Keele and Park (2005) have demonstrated in a recent Monte Carlo simulation analysis of heteroskedastic probit and logit. Among other things, Keele and Park find that the estimate of $\boldsymbol{\beta}$ tends to be severely biased unless the sample size is large ($n \geq 500$). In our example, the sample size was considerably smaller than 500 and one may therefore question the estimates presented in Table 9.8. Keele and Park also found that small sample bias in the estimate of $\boldsymbol{\gamma}$ is less severe, although still problematic. They also find that the estimated standard errors tend to be too large and the coverage probabilities too low, even in reasonably large samples. All of these problems are compounded in the face of measurement error and/or model mis=specification. The lesson here is that heteroskedastic logit and probit models should be used with extreme care. One should be reasonably convinced that one has a correctly specified model and one should have a reasonably large sample before one can place great confidence in the results from these models.

9.3 Alternatives to Logit and Probit

Logit and probit are the best-known of the binary regression models. However, useful alternatives are available. These may be better capable of fitting skewed distributions of y, fitting models where the maximum marginal effect occurs for $\pi \neq .5$, or addressing complex causality. The gompit, scobit, power logit, rare events logit, and Boolean logit/probit models fall into this category.

9.3.1 The Gompit Model

Model and Estimation

The gompit model is one of the alternatives to logit and probit. The model can be motivated using the latent variable framework that was used to motivate the logit and probit models. Let y be a binary indicator of whether an alternative θ was chosen. We postulate that a decision maker chooses θ when he or she derives positive utility from this alternative, i.e. when $y_i^* > 0$. We further postulate that $y_i^* = \boldsymbol{x}_i\boldsymbol{\beta} + \epsilon_i$, where $\boldsymbol{x}_i$ is a vector of attributes of the decision maker and ϵ_i is a stochastic component. Thus, $\Pr(y_i^* > 0) = \Pr(\boldsymbol{x}_i\boldsymbol{\beta} + \epsilon_i > 0) = \Pr(\epsilon_i > -\boldsymbol{x}_i\boldsymbol{\beta})$. This is precisely the same derivation that we used for the logit and probit models.

Where things become different is in the error distribution that we postulate for ϵ_i. The gompit model assumes the standard Type-I extreme value distribution, which is also known as the standard Gumbel distribution:[52]

$$G(\epsilon) \;=\; \exp\{-\exp(-\epsilon)\}$$

Figure 9.9 shows the standard extreme value distribution and compares it to the standard logistic distribution. Compared to the standard logistic distribution, the standard extreme value distribution has a slightly higher mean (.57722 versus 0) and a positive skew (skewness coefficient is 1.139547). The presence of a degree of skewness allows the Gompit model to better accommodate a skewed distribution in y.[53]

With the distributional assumption in place, the probability of choosing θ can now be computed:

$$\begin{aligned}
\pi_i &= \Pr(\epsilon_i > -\boldsymbol{x}_i\boldsymbol{\beta}) \\
&= 1 - \Pr(\epsilon_i < -\boldsymbol{x}_i\boldsymbol{\beta}) \\
&= 1 - G(-\boldsymbol{x}_i\boldsymbol{\beta}) \\
&= 1 - \exp\{-\exp(\boldsymbol{x}_i\boldsymbol{\beta})\}
\end{aligned}$$

[52]The distribution is also referred to as the Gompertz distribution, which explains the name gompit model. Note that there are three types of extreme value distributions. In addition to the Gumbel distribution, there are Frechet (Type-II extreme value) and Weibull (Type-III extreme value) distributions.

[53]If it is a negative skew that needs to be accommodated, then the distribution may be defined as $\exp\{-\exp(\epsilon)\}$; this distribution has a mean of -.57722 and a skewness coefficient of -1.139547.

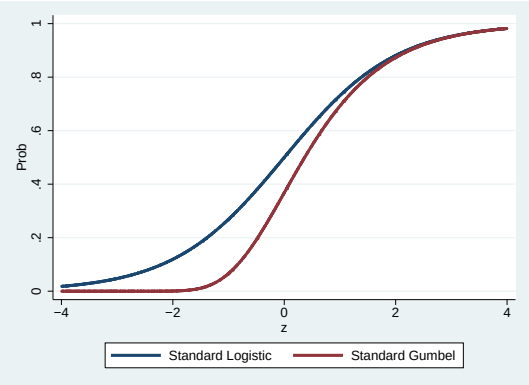

Figure 9.9: Standard Gumbel Distribution

This is the gompit model.

Estimation of the gompit model is relatively straightforward. The data consist of 1s that occur with probability π_i and 0s that occur with probability $1-\pi_i$. Substituting $1-G(-\boldsymbol{x}_i\boldsymbol{\beta})$ for π_i and $G(-\boldsymbol{x}_i\boldsymbol{\beta})$ for $1-\pi_i$, the likelihood function can be written as

$$\mathcal{L} = \prod_{i=1}^{n} [1 - G(-\boldsymbol{x}_i\boldsymbol{\beta})]^{y_i} [G(-\boldsymbol{x}_i\boldsymbol{\beta})]^{1-y_i}$$

The log-likelihood function is

$$\ell = \sum_{i=1}^{n} \{y_i \ln [1 - G(-\boldsymbol{x}_i\boldsymbol{\beta})] + (1 - y_i) \ln [G(-\boldsymbol{x}_i\boldsymbol{\beta})]\}$$

This is not a complex log-likelihood function so that estimation generally poses few problems.

The maximum marginal effect of the gompit model arises at $\pi_i = .632121$.[54]

[54] *Proof:* Let $w_i = \boldsymbol{x}_i\boldsymbol{\beta}$. The marginal effect with respect to w_i is given by:

$$\frac{\partial \pi_i}{\partial w_i} = \exp [- \exp (w_i) + w_i]$$

Optimization of this marginal effect requires that we take the first derivative and set it equal to zero:

$$\frac{\partial^2 \pi_i}{\partial w_i^2} = \exp [- \exp (w_i) + w_i] [\exp (w_i) - 1] = 0$$

Thus, the gompit model is quite useful if the goal is to fit a model where the maximum marginal effect does not arise at $\pi_i = .5$.

Example

As an example of a gompit model, we analyze the incidence of military coups in the post-World War II period, using Banks' *Handbook of Social and Political Indicators*. Military coups are extremely rare: out of 3,493 country-year cases that we analyze here, only 148 (less than 5%) experienced military coups. The severe skewness in this distribution may make logit and probit analysis a poor choice. Instead, we fit a gompit model with three predictors: regime type (1 = civilian regime; 0 = otherwise), per capita GDP (logged), and size of the military.

The gompit model can be estimated in Stata by issuing the `cloglog` command, which stands for complementary log-log regression. The syntax is

```
cloglog depvar [ indepvars ] [ , options ]
```

When applied to the Banks data, we obtain the estimates shown in Table 9.10. For the sake of comparison, I have also included the logit estimates of the coefficients, which, in this case, tell a very similar story. To shed further light on the gompit estimates, they can be converted into predicted probabilities or marginal effects using the same methods that were discussed earlier.

9.3.2 The Scobit Model

Model and Estimation

Nagler (1994) proposed the Burr-II distribution for the stochastic component in the latent variable model.[55] This distribution contains an **ancillary parameter** that controls the degree of skewness in the error distribution, giving a great deal of flexibility.[56] The resulting model is known as the

This equation holds true when $\exp(w_i) = 1$, which occurs when $w_i = 0$. Substitution of $w_i = 0$ into the formula for π_i yields .632121.

[55] Nagler (1994) calls this the Burr-10 distribution because it is the 10th equation in the list of a dozen distributions that Burr (1942) proposed but, as Achen (2002) notes, it is conventionally referred to as the Burr-II or Burr Type-II distribution.

[56] In general, an ancillary parameter is a feature of a probability function that is constant across observation units i.

Table 9.10: Gompit Model of Military Coups

	Logit		Gompit	
Predictor	**B**	**S.E.**	**B**	**S.E.**
Civilian Regime	-2.475^{**}	.207	-2.366^{**}	.202
Size of the Military	$-.001$	.001	$-.001$	.001
Log Per Capita GDP	-1.004^{**}	.239	$-.915^{**}$	.223
Constant	1.040^{+}	.575	.705	.526

Notes: $n = 3493$. $^{**}p < .01$, $^{+}p < .10$ (two-tailed). Table entries are maximum likelihood logit and gompit estimates with cluster-corrected estimated standard errors (clustering by country).

scobit (skewed logit) model and is closely related to methods developed by Aranda-Ordaz (1981) and Prentice (1976).

Derivation of the scobit model follows the by now familiar pattern. Let y_i denote a binary choice variable that takes on the value 1 if alternative θ is chosen. This variable is related to an underlying latent utility, which is modeled as $y_i^* = \boldsymbol{x}_i\boldsymbol{\beta} + \epsilon_i$. The link between y_i and y_i^* is probabilistic, due to the stochastic nature of ϵ_i. Correspondingly, $\pi_i = \Pr(y_i = 1) = \Pr(y_i^* > 0) = \Pr(\boldsymbol{x}_i\boldsymbol{\beta} + \epsilon_i > 0) = \Pr(\epsilon_i > -\boldsymbol{x}_i\boldsymbol{\beta})$.

This is where the distributional assumption comes into play. The scobit model assumes

$$B(\epsilon) \;=\; [1 + \exp(-\epsilon)]^{-\alpha}$$

where $\alpha > 0$ is an ancillary parameter controlling the degree of skewness. Figure 9.10 shows different examples of the Burr-II distribution, clearly illustrating the different degrees of skewness. In general, $0 < \alpha < 1$ implies negative skewness, $\alpha = 1$ implies no skewness, and $\alpha > 1$ implies positive skewness.

With the distributional assumption in place, it is now easy to work out the choice probability:

$$\begin{aligned}
\pi_i &= \Pr(\epsilon_i > -\boldsymbol{x}_i\boldsymbol{\beta}) \\
&= 1 - \Pr(\epsilon_i < -\boldsymbol{x}_i\boldsymbol{\beta}) \\
&= 1 - B(-\boldsymbol{x}_i\boldsymbol{\beta}) \\
&= 1 - [1 + \exp(\boldsymbol{x}_i\boldsymbol{\beta})]^{-\alpha}
\end{aligned}$$

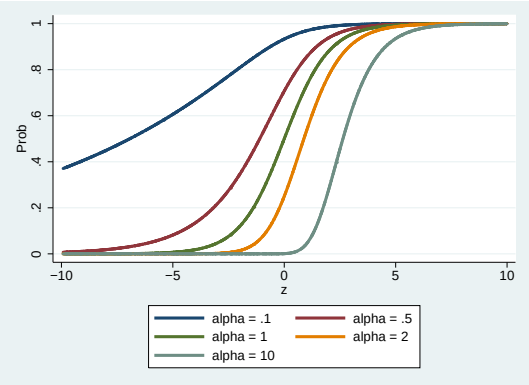

Figure 9.10: Burr-II Distribution

Note that when $\alpha = 1$, this expression is identical to the logit model.[57]

The likelihood function for the scobit model comes about by recognizing that the data consist of 1s that occur with probability π_i and 0s that occur with probability $1 - \pi_i$. Thus,

$$\mathcal{L} = \prod_{i=1}^{n} \left[1 - B(-\boldsymbol{x}_i\boldsymbol{\beta})\right]^{y_i} \left[B(-\boldsymbol{x}_i\boldsymbol{\beta})\right]^{1-y_i}$$

The log-likelihood function is

$$\ell = \sum_{i=1}^{n} \left\{y_i \ln\left[1 - B(-\boldsymbol{x}_i\boldsymbol{\beta})\right] + (1 - y_i) \ln\left[B(-\boldsymbol{x}_i\boldsymbol{\beta})\right]\right\}$$

This is a complex log-likelihood function and convergence problems are quite common. Another problem is that α can be highly collinear with the elements of $\boldsymbol{\beta}$, in particular the constant. This may produce inflated standard errors.

[57] *Proof:* When $\alpha = 1$ then $\pi_i = 1 - \left[1 + \exp\left(\boldsymbol{x}_i\boldsymbol{\beta}\right)\right]^{-1}$. This may also be written as

$$\begin{aligned} \pi_i &= \frac{1 + \exp\left(\boldsymbol{x}_i\boldsymbol{\beta}\right)}{1 + \exp\left(\boldsymbol{x}_i\boldsymbol{\beta}\right)} - \frac{1}{1 + \exp\left(\boldsymbol{x}_i\boldsymbol{\beta}\right)} \\ &= \frac{\exp\left(\boldsymbol{x}_i\boldsymbol{\beta}\right)}{1 + \exp\left(\boldsymbol{x}_i\boldsymbol{\beta}\right)} \\ &= \Lambda\left(\boldsymbol{x}_i\boldsymbol{\beta}\right) \end{aligned}$$

For this reason, Nagler (1994) suggests testing the elements of $\boldsymbol{\beta}$ using a likelihood ratio approach rather than a Wald test approach.

Despite the fact that the scobit model is much more difficult to estimate than a logit, probit, or gompit model, it stands out through its remarkable flexibility. Since α is an estimated parameter, one does not have to make any a priori assumptions about the degree of skewness that has to be assumed. Correspondingly, there is flexibility in where the maximum marginal effect occurs. It can be demonstrated that this maximum occurs at[58]

$$\pi_i \quad = \quad 1 - \left[\frac{\alpha}{\alpha + 1}\right]^{\alpha}$$

If $\alpha < 1$, then the maximum marginal effect occurs at $0 < \pi_i < .5$. If $\alpha = 1$, then the maximum occurs at $\pi_i = .5$, the same value as for the logit model. Finally, if $\alpha > 1$, then the maximum occurs for $.5 < \pi_i < 1 - e^{-1}$. Here $1 - e^{-1} \approx .632$ is the upper asymptote on π_i.

The presence of this upper asymptote is one limitation of the scobit model. If one expects the point of maximal impact to be greater than .632, then one should fit a **power logit** model instead. In analogy to the logit model, Achen (2002) specifies the power logit model as follows:

$$\pi_i \quad = \quad \left[1 + \exp\left(-\boldsymbol{x}_i\boldsymbol{\beta}\right)\right]^{-\alpha}$$

and shows that the marginal effect is optimized at

$$\pi_i \quad = \quad \left[\frac{\alpha}{\alpha + 1}\right]^{\alpha}$$

where $e^{-1} < \pi_i < 1$. Estimation of this model creates problems similar to the scobit model.

[58] *Proof:* As usual, let $w_i = \boldsymbol{x}_i\boldsymbol{\beta}$. Then the marginal effect with respect to w_i is given by

$$\frac{d\pi_i}{dw_i} \quad = \quad \frac{\alpha \exp(w_i)}{\left[1 + \exp(w_i)\right]^{1+\alpha}}$$

We seek to optimize this marginal effect. Thus we take the derivative of the marginal effect with respect to w_i and set the result to 0. This produces

$$\frac{d^2\pi_i}{dw_i^2} \quad = \quad -\frac{\alpha \exp(w_i)\left[\alpha \exp(w_i) - 1\right]}{\left[1 + \exp(w_i)\right]^{2+\alpha}} = 0$$

The solution to this equation is $w_i = -\ln \alpha$. Substitution of this result in the formula for π_i yields the formula for the maximum marginal effect.

Example

Returning to the military coup data, we can fit a scobit model by using Stata's `scobit` command:

```
scobit depvar indepvars [, options]
```

Given the difficult scobit log-likelihood, I routinely include `dif` (difficult) as one of the options. Table 9.11 shows the scobit estimates, which tell a similar story to the logit and gompit estimates reported earlier. You should note, however, that the standard errors are quite a bit higher than those for the gompit model, a consequence of the high degree of collinearity between the scobit estimates (the average absolute correlation between the estimates is .762). We also see that $\ln(\alpha)$ is statistically significant at the .01 level, resulting in an estimate of α of .101.[59] The nature of this estimate is such that we apparently need to introduce considerable negative skewness to accommodate the distribution of y_i, which should not come as a surprise in light of the preponderance of 0s on this variable.

9.3.3 The Rare Events Logit Model

Model and Estimation

King and Zeng (2001) have proposed a rare events logit model for analyzing a binary response variable with a preponderance of 0s.[60] The general idea is to select all of the 1s and a sample of the 0s, assuming 0s are preponderant. A logit analysis is then performed on this data.

There are several reasons to opt for this approach. First, the 0s are much less informative about the parameters than the 1s. Second, having too many 0s in a regular logit model can produce a downward bias in the predicted probabilities. Finally, by sampling only a subset of the 0s more attention can be paid to adequate measurement of the covariates. The potential payoff

[59]For estimation purposes it is better to parameterize the scobit model in terms of $\ln(\alpha)$ since it is not bounded, whereas α is bounded to be strictly positive. Remember that the scobit model reduces to a logit model when $\alpha = 1$. In terms of $\ln(\alpha)$, this means that a test of $H_0 : \ln(\alpha) = 0$ provides the mechanism for telling whether the scobit model deviates significantly from the logit model.

[60]The model applies also to cases with a preponderance of 1s but I'll develop it here for a skew in the direction of 0s, a situation that arises frequently especially in dyadic studies of conflict between nations.

Table 9.11: Scobit and Rare Events Logit Models of Military Coups

	Scobit		RE Logit	
Predictor	**B**	**S.E.**	**B**	**S.E.**
Civilian Regime	-3.543^{**}	1.157	-2.390^{**}	.267
Size of the Military	$-.001$	.001	$-.001$	.001
Log Per Capita GDP	-1.605^{**}	.598	$-.880^{**}$	.332
Constant	5.933^{+}	3.359	.326	.824
$\ln(\alpha)$	-2.293^{**}	.888		
α	.101			

Notes: $n = 3493$ for scobit and $n = 477$ for the rare events logit. $^{**}p < .01$, $^{+}p < .10$ (two-tailed). Table entries are maximum likelihood scobit and rare events (RE) logit estimates with cluster-corrected estimated standard errors (clustering by country). The rare events logit was based on a sample of 10% of 0s and 100% of 1s. The weight is .0314.

of better measures is generally far greater than the payoff of covering all instances of 0s.

The general approach of the rare events logit is to engage in choice-based sampling (also known as endogenous stratified sampling or the case-control design). In this approach, we select on the dependent variable by covering all instances of $y = 1$ (the "cases") and a random selection of instances of $y = 0$ (the "controls"). One can adjust the sampling fraction on the 0s so as to maximize attention on measurement. Once all of the data have been collected, a logit analysis is run on the choice-based sample, which will be characterized by far fewer 0s than a traditional approach would be.

Of course, the logit estimates will have to be adjusted. There are two strategies for doing so. First, the *prior correction* adjusts the estimate of β_0. This is the only problematic estimate under choice-based sampling; the remaining estimates are consistent. The correction on $\hat{\beta}_0$ is as follows:

$$\hat{\beta}_0 - \ln\left[\frac{1-\tau}{\tau}\frac{\bar{y}}{1-\bar{y}}\right]$$

Here τ is the population fraction of 1s, whereas $\bar{y}$ is the fraction of 1s in the sample. In an ordinary logit estimation under simple random sampling, one can expect $\bar{y} = \tau$ and the correction disappears. However, in the rare

events logit the fraction of 1s in the sample is larger than τ, which produces a downward adjustment on the constant.

A second strategy is *weighting*, which introduces weights into the logit log-likelihood function. Let $w_1 = \tau/\bar{y}$ and let $w_0 = (1-\tau)/(1-\bar{y})$, then the weighted exogenous sampling ML estimator can be obtained by optimizing

$$\ell \;=\; w_1 \sum_{y_i=1} \ln(\pi_i) + w_0 \sum_{y_i=0} \ln(1-\pi_i)$$

King and Zeng (2001) express a preference for weighting, since it works better when the functional form winds up being misspecified.

Choice-based sampling is not without risk. In general, sampling on the dependent variable is a risky business but with proper corrections of the estimators, the approach is legitimate. However, if selection on y is accompanied by differential selection on covariates, then serious biases may emerge. King and Zeng (2001) use the example of selecting all people with liver cancer in the local hospital ($y = 1$) and a random selection of the population without the disease ($y = 0$). The problem is that the first group is also more inclined to seek health care, or else they would not be in the hospital. This differential inclination becomes an omitted variable that can throw off the estimates for the covariates in the model.

Example

The last two columns in Table 9.11 show the rare events logit estimates for military coups. These were obtained using the `relogit` command developed by Michael Tomz, Gary King, and Langche Zeng. The syntax of this command is

`relogit` *depvar* [*indepvars*] [`, wc(#) pc(#)`]

where `ws(#)` specifies the weight and `pc(#)` the prior correction. As you can see from the table, the results from the rare events logit are consistent with those from the other analyses that we performed on the military coup data.

9.3.4 Boolean Logit and Probit*

Model and Estimation

A common complication in statistical modeling is how one should deal with causal complexity. Key questions that arise in this context are: (1) to what

extent does the effect of a predictor depend on the presence of other covariates in the model and (2) what combination of predictors produces an outcome? Ragin (1987) offers one approach to answering these questions that is based on fuzzy sets. Recently, Braumoeller (2003, 2004) introduced an alternative approach in the form of Boolean logit and probit models.

Braumoeller's approach is particularly good at handling two varieties of causal complexity, namely "multiple conjunctural causation" and substitutability. In the context of a pair of predictors, x_1 and x_2, multiple conjunctural causation means that both x_1 *and* x_2 cause y. Substitutability means that x_1 *or* x_2 causes y. Combinations of these two processes are also possible.

The model for multiple conjunctural causations centers on the intersection of two or more predictors. For a pair of predictors, this implies that one models the probability that each relevant factor will occur. One then obtains the probability of the intersection of these factors by relying on the multiplication rule for independent events: $\Pr(x_1 \cap x_2) = \Pr(x_1) \times \Pr(x_2)$. In general, this approach produces the following likelihood function:

$$\mathcal{L} = \prod_{i=1}^{n} \left\{ \prod_{j=1}^{J} F(\mathbf{x}_{ij}\boldsymbol{\beta}_j) \right\}^{y_i} \left\{ 1 - \prod_{j=1}^{J} F(\mathbf{x}_{ij}\boldsymbol{\beta}_j) \right\}^{1-y_i}$$

Here J refers to the number of causal pathways to y. This can be a difficult likelihood function to optimize.

As a hypothetical example, consider the act of counter-arguing a counter-attitudinal message (y). Several theories in political psychology suggest that two factors or causal paths combine to produce counter-arguing: (1) a strong prior attitude (c_1), which produces the motivation to counter-argue, and (2) political knowledge (c_2), which drives the ability to counter-argue. Thus,

$$\Pr(y = 1) = \Pr(c_1 = 1) \times \Pr(c_2 = 1)$$

Modeling c_1 and c_2 as a function of covariates then yields

$$\Pr(y = 1) = F(\boldsymbol{x}_1\boldsymbol{\beta}_1) \times F(\boldsymbol{x}_2\boldsymbol{\beta}_2)$$

The probability that no counter-arguing will occur is then $\Pr(y = 0) = 1 - \Pr(y = 1) = 1 - F(\boldsymbol{x}_1\boldsymbol{\beta}_1) \times F(\boldsymbol{x}_2\boldsymbol{\beta}_2)$. This recovers all of the elements of the likelihood function.

The model for substitutability centers on the union of two or more predictors. For a pair of predictors, this means that one starts by modeling the probability that each factor will occur. One than obtains the probability of the union by relying on the additive rule of probability: $\Pr(x_1 \cup x_2) = \Pr(x_1) + \Pr(x_2) - \Pr(x_1 \cap x_2) = 1 - [1 - \Pr(x_1)][1 - \Pr(x_2)]$. In general, this approach produces the following likelihood:

$$\mathcal{L} \;=\; \prod_{i=1}^{n} \left\{ 1 - \prod_{j=1}^{J} \left[1 - F(\mathbf{x}_{ij}\boldsymbol{\beta}_j) \right]^{y_i} \right\} \left\{ \prod_{j=1}^{J} \left[1 - F(\mathbf{x}_{ij}\boldsymbol{\beta}_j) \right]^{1-y_i} \right\}$$

Here J again refers to the number of causal pathways. This likelihood function, too, is complex and convergence problems may arise.

As another hypothetical example, consider attitude importance (y). Work by Krosnick and others suggests that attitude importance can flow from any of three factors: (1) interests (c_1), (2) values (c_2), or (3) identity (c_3). Thus

$$\Pr(y = 1) \;=\; 1 - [1 - \Pr(c_1 = 1)] \times [1 - \Pr(c_2 = 1)] \times [1 - \Pr(c_3 = 1)]$$

Modeling $c_1 \cdots c_3$ as a function of covariates then yields

$$\Pr(y = 1) \;=\; 1 - [1 - F(\boldsymbol{x}_1\boldsymbol{\beta}_1)] \times [1 - F(\boldsymbol{x}_2\boldsymbol{\beta}_2)] \times [1 - F(\boldsymbol{x}_3\boldsymbol{\beta}_3)]$$

This gives the probability that the attitude is personally important to the individual. The probability that the attitude is not important is then given by $\Pr(y = 0) = 1 - \Pr(y = 1) = [1 - F(\boldsymbol{x}_1\boldsymbol{\beta}_1)] \times [1 - F(\boldsymbol{x}_2\boldsymbol{\beta}_2)] \times [1 - F(\boldsymbol{x}_3\boldsymbol{\beta}_3)]$. Thus, all elements of the likelihood function have been recovered.

Example

Braumoeller (2004) describes the Stata program `mlboolean` that will perform Boolean logit and probit analysis. The generic syntax is

```
mlboolean link n (calculus) (depvar) (indepvars_1) ···
(indepvars_n) [ , options ]
```

Here `link` is the link function, which is either logit or probit. The number of causal paths is set by `n`≤ 5. The `calculus` argument allows the user to specify the causal logic, for example multiple conjectural causation or substitutability. The dependent variable is listed in parentheses as are

the covariates involved in the causal paths. The n sets of covariates that are specified may partially overlap, as would be the case if one covariate is involved in multiple causal paths.[61]

To illustrate Boolean logit let us revisit the military coup data. Imagine that our theory states that military coups are less likely when there is a civilian regime *and* one of the following is true: (1) the size of the military is small *or* (2) the logged per capita GDP is high. The following syntax would allow us to estimate this model:

```
mlboolean logit 3 (aand(borc)) (dicoup) (civilian)
(military_size) (log_pc_gdp)
```

In the calculus, `a` refers to the first covariate that is listed and `b` and `c` to the second and third covariates. Thus, `(aand(borc))` means that an understanding of the incidence of military coups requires that we know the first listed covariate (in our case regime type) and either the second listed covariate (size of the military) or the third listed covariate (logged per capita GDP).

Table 9.12 shows the estimates. The results lend support to the idea that regime type and per capita GDP jointly explain coups, but not regime type and size of the military. This is due to the insignificant coefficient on size of the military.

9.4 Bivariate and Multivariate Probit*

Imagine that we seek to model two binary choices simultaneously. We could run separate binary regression models but, to the extent that the outcomes are correlated, efficiency gains may be had from modeling the outcomes simultaneously. That is to say, we would run a model that allows the stochastic components of both outcomes to be correlated. The bivariate normal distribution produces a natural framework for doing so. The resulting model is known as the bivariate probit model. In principle, the basic idea of the bivariate probit model can be extended to multiple binary outcomes by drawing on the multivariate normal distribution. This model is known as the multivariate probit model and is considerably more difficult to estimate.

[61] The options mainly include methods for choosing an algorithm and controlling convergence. At this time, `mlboolean` does not allow for robust estimation or cluster-corrected standard errors as options.

Table 9.12: Boolean Logit Model of Military Coups

Predictor	B	S.E.
Path 1:		
Civilian Regime	-2.499^{**}	.192
Constant	-1.161^{**}	.181
Path 2:		
Size of the Military	$-.120$	.156
Constant	.015	1.525
Path 3:		
Log Per Capita GDP	-4.654^{*}	1.914
Constant	13.216^{*}	5.984

Notes: $n = 3493$. $^{**}p < .01$, $^{*}p < .05$ (two-tailed). Table entries are maximum likelihood Boolean logit estimates with estimated standard errors. 95.8% of the cases are correctly predicted by the model.

The Bivariate Probit Model

Model and Estimation The bivariate probit model was first proposed by Ashford and Sowden (1970). Derivation of the model begins by considering a choice problem involving two alternatives, θ and ψ, that are not mutually exclusive. Let

$$y_{i1} = \begin{cases} 1 & \text{if } \theta \text{ is chosen} \\ 0 & \text{otherwise} \end{cases}$$

$$y_{i2} = \begin{cases} 1 & \text{if } \psi \text{ is chosen} \\ 0 & \text{otherwise} \end{cases}$$

Underlying the choices is a latent variable

$$y_{ij}^{*} = \boldsymbol{x}_{ij}\boldsymbol{\beta}_j + \epsilon_{ij}$$

for $j = 1, 2$. We assume that $y_{ij} = 1$ if $y_{ij}^{*} > 0$. Therefore,

$$\begin{aligned} \Pr(y_{ij} = 1) &= \Pr(y_{ij}^{*} > 0) \\ &= \Pr(\boldsymbol{x}_{ij}\boldsymbol{\beta}_j + \epsilon_{ij} > 0) \\ &= \Pr(\epsilon_{ij} > -\boldsymbol{x}_{ij}\boldsymbol{\beta}_j) \end{aligned}$$

To solve this we need to impose a distributional assumption.

It is reasonable to adopt the bivariate normal distribution, since this distribution allows for correlated errors and is still relatively straightforward. Here we assume $\mu_j = \mathcal{E}[\epsilon_{ij}] = 0$ and $\sigma_j^2 = V[\epsilon_{ij}] = 1$, for $j = 1, 2$. The distribution also contains the correlation between ϵ_{i1} and ϵ_{i2}, which is given by ρ. Thus, the bivariate normal PDF is given by

$$\phi_2(\epsilon_{i1}, \epsilon_{i2}|\rho) = \frac{1}{2\pi\sqrt{1-\rho^2}} \exp\left\{-\frac{\epsilon_{i1}^2 + \epsilon_{I2}^2 - 2\rho\epsilon_{i1}\epsilon_{i2}}{2(1-\rho^2)}\right\}$$

With the distributional assumption in place, we can now work out the choice probabilities. First, consider π_{i00}, which is the probability of not choosing θ or ψ:

$$\begin{aligned}
\pi_{i00} &= \Pr(y_{i1} = 0 \cap y_{i2} = 0) \\
&= \Pr(y_{i1}^* \leq 0 \cap y_{i2}^* \leq 0) \\
&= \Pr(\boldsymbol{x}_{i1}\boldsymbol{\beta}_1 + \epsilon_{i1} \leq 0 \cap \boldsymbol{x}_{i2}\boldsymbol{\beta}_2 + \epsilon_{i2} \leq 0) \\
&= \Pr(\epsilon_{i1} \leq -\boldsymbol{x}_{i1}\boldsymbol{\beta}_1 \cap \epsilon_{i2} \leq -\boldsymbol{x}_{i2}\boldsymbol{\beta}_2) \\
&= \int_{-\infty}^{-\boldsymbol{x}_{i1}\boldsymbol{\beta}_1} \int_{-\infty}^{-\boldsymbol{x}_{i2}\boldsymbol{\beta}_2} \phi_2(\epsilon_{i1}, \epsilon_{i2}|\rho) d\epsilon_{i1} d\epsilon_{i2} \\
&= \Phi_2(-\boldsymbol{x}_{i1}\boldsymbol{\beta}_1, -\boldsymbol{x}_{i2}\boldsymbol{\beta}_2, \rho)
\end{aligned}$$

where Φ_2 is the bivariate normal CDF. Next, consider π_{i10}, which is the probability of choosing θ but not ψ:

$$\begin{aligned}
\pi_{i10} &= \Pr(y_{i1} = 1 \cap y_{i2} = 0) \\
&= \Pr(y_{i1}^* > 0 \cap y_{i2}^* \leq 0) \\
&= \Pr(\boldsymbol{x}_{i1}\boldsymbol{\beta}_1 + \epsilon_{i1} > 0 \cap \boldsymbol{x}_{i2}\boldsymbol{\beta}_2 + \epsilon_{i2} \leq 0) \\
&= \Pr(\epsilon_{i1} > -\boldsymbol{x}_{i1}\boldsymbol{\beta}_1 \cap \epsilon_{i2} \leq -\boldsymbol{x}_{i2}\boldsymbol{\beta}_2) \\
&= \Pr(\epsilon_{i1} < \boldsymbol{x}_{i1}\boldsymbol{\beta}_1 \cap \epsilon_{i2} \leq -\boldsymbol{x}_{i2}\boldsymbol{\beta}_2) \\
&= \int_{-\infty}^{\boldsymbol{x}_{i1}\boldsymbol{\beta}_1} \int_{-\infty}^{-\boldsymbol{x}_{i2}\boldsymbol{\beta}_2} \phi_2(\epsilon_{i1}, \epsilon_{i2}|\rho) d\epsilon_{i1} d\epsilon_{i2} \\
&= \Phi_2(\boldsymbol{x}_{i1}\boldsymbol{\beta}_1, -\boldsymbol{x}_{i2}\boldsymbol{\beta}_2, -\rho)
\end{aligned}$$

Notice that the sign on the correlation has changed due to the opposite signs on the conditions for y_{i1}^* and y_{i2}^*. Similarly, π_{i01}, which is the probability of

choosing ψ but not θ, is given by:

$$\begin{aligned}
\pi_{i01} &= \Pr(y_{i1} = 0 \cap y_{i2} = 1) \\
&= \Pr(y_{i1}^* \leq 0 \cap y_{i2}^* > 0) \\
&= \Pr(\boldsymbol{x}_{i1}\boldsymbol{\beta}_1 + \epsilon_{i1} \leq 0 \cap \boldsymbol{x}_{i2}\boldsymbol{\beta}_2 + \epsilon_{i2} > 0) \\
&= \Pr(\epsilon_{i1} \leq -\boldsymbol{x}_{i1}\boldsymbol{\beta}_1 \cap \epsilon_{i2} > -\boldsymbol{x}_{i2}\boldsymbol{\beta}_2) \\
&= \Pr(\epsilon_{i1} \leq -\boldsymbol{x}_{i1}\boldsymbol{\beta}_1 \cap \epsilon_{i2} < \boldsymbol{x}_{i2}\boldsymbol{\beta}_2) \\
&= \int_{-\infty}^{-\boldsymbol{x}_{i1}\boldsymbol{\beta}_1} \int_{-\infty}^{\boldsymbol{x}_{i2}\boldsymbol{\beta}_2} \phi_2(\epsilon_{i1}, \epsilon_{i2}|\rho) d\epsilon_{i1} d\epsilon_{i2} \\
&= \Phi_2(-\boldsymbol{x}_{i1}\boldsymbol{\beta}_1, \boldsymbol{x}_{i2}\boldsymbol{\beta}_2, -\rho)
\end{aligned}$$

Finally, π_{i11}—the probability of choosing both θ and ψ—is given by:

$$\begin{aligned}
\pi_{i11} &= \Pr(y_{i1} = 1 \cap y_{i2} = 1) \\
&= \Pr(y_{i1}^* > 0 \cap y_{i2}^* > 0) \\
&= \Pr(\boldsymbol{x}_{i1}\boldsymbol{\beta}_1 + \epsilon_{i1} > 0 \cap \boldsymbol{x}_{i2}\boldsymbol{\beta}_2 + \epsilon_{i2} > 0) \\
&= \Pr(\epsilon_{i1} > -\boldsymbol{x}_{i1}\boldsymbol{\beta}_1 \cap \epsilon_{i2} > -\boldsymbol{x}_{i2}\boldsymbol{\beta}_2) \\
&= \Pr(\epsilon_{i1} < \boldsymbol{x}_{i1}\boldsymbol{\beta}_1 \cap \epsilon_{i2} < \boldsymbol{x}_{i2}\boldsymbol{\beta}_2) \\
&= \int_{-\infty}^{\boldsymbol{x}_{i1}\boldsymbol{\beta}_1} \int_{-\infty}^{\boldsymbol{x}_{i2}\boldsymbol{\beta}_2} \phi_2(\epsilon_{i1}, \epsilon_{i2}|\rho) d\epsilon_{i1} d\epsilon_{i2} \\
&= \Phi_2(\boldsymbol{x}_{i1}\boldsymbol{\beta}_1, \boldsymbol{x}_{i2}\boldsymbol{\beta}_2, \rho)
\end{aligned}$$

In general, the choice probabilities are given by

$$\Pr(y_{i1} \cap y_{i2}) = \Phi_2(q_{i1}\boldsymbol{x}_{i1}\boldsymbol{\beta}_1, q_{i2}\boldsymbol{x}_{i2}\boldsymbol{\beta}_2, q_{i1}q_{i2}\rho) \tag{9.22}$$

where $q_{ij} = 2y_{ij} - 1$.

Estimation of the bivariate probit model proceeds through MLE. Our data consists of observation pairs (y_{i1}, y_{i2}), each occurring with probability π_{ijk}, where $j, k \in \{0, 1\}$. Thus, the likelihood function is given by

$$\begin{aligned}
\mathcal{L} &= \prod_{y_{i1}=0 \cap y_{i2}=0} \pi_{i00} \prod_{y_{i1}=1 \cap y_{i2}=0} \pi_{i10} \prod_{y_{i1}=0 \cap y_{i2}=1} \pi_{i01} \prod_{y_{i1}=1 \cap y_{i2}=1} \pi_{i11} \\
&= \prod_i \pi_{i00}^{(1-y_{i1})(1-y_{i2})} \pi_{i10}^{y_{i1}(1-y_{i2})} \pi_{i01}^{(1-y_{i1})y_{i2}} \pi_{i11}^{y_{i1}y_{i2}} \\
&= \prod_i \Phi_2(q_{i1}\boldsymbol{x}_{i1}\boldsymbol{\beta}_1, q_{i2}\boldsymbol{x}_{i2}\boldsymbol{\beta}_2, q_{i1}q_{i2}\rho)
\end{aligned}$$

The log-likelihood function is

$$\ell \;=\; \sum_i \ln \Phi_2(q_{i1}\boldsymbol{x}_{i1}\boldsymbol{\beta}_1, q_{i2}\boldsymbol{x}_{i2}\boldsymbol{\beta}_2, q_{i1}q_{i2}\rho)$$

While this log-likelihood function looks complicated it is relatively straightforward. The reason is that the gradient is a function of the univariate standard normal distribution, which is easy to work with. Hence, estimation of the bivariate probit model will generally produce few complications outside of the usual problems such as separation.

Interpretation Predicted probabilities can be obtained by evaluating (9.22) at selected values of one of the predictors, while holding all other covariates constant. To obtain marginal effects we proceed as follows. Let $\boldsymbol{x}_i = \boldsymbol{x}_{i1} \cup \boldsymbol{x}_{i2}$, i.e. $\boldsymbol{x}_i$ includes the predictors relevant to each of the outcomes as well as predictors that are relevant to both outcomes. Further, let $\boldsymbol{x}_{ij}\boldsymbol{\beta}_j = \boldsymbol{x}_i\boldsymbol{\gamma}_j$. Finally, let

$$\begin{aligned} g_1 &= \phi(q_{i1}\boldsymbol{x}_{i1}\boldsymbol{\beta}_1)\Phi\left(\frac{q_{i2}\boldsymbol{x}_{i2}\boldsymbol{\beta}_2 - q_{i1}^2 q_{i2}\rho\boldsymbol{x}_{i1}\boldsymbol{\beta}_1}{\sqrt{1-(q_{i1}q_{i2}\rho)^2}}\right) \\ g_2 &= \phi(q_{i2}\boldsymbol{x}_{i2}\boldsymbol{\beta}_2)\Phi\left(\frac{q_{i1}\boldsymbol{x}_{i1}\boldsymbol{\beta}_1 - q_{i1} q_{i2}^2\rho\boldsymbol{x}_{i2}\boldsymbol{\beta}_2}{\sqrt{1-(q_{i1}q_{i2}\rho)^2}}\right) \end{aligned}$$

Then the marginal effects are given by:

$$\frac{\partial \Phi_2}{\partial \boldsymbol{x}} \;=\; g_1\boldsymbol{\gamma}_1 + g_2\boldsymbol{\gamma}_2$$

This is the marginal effect aggregated over both response variables. Standard errors for these effects can be computed via the delta method.

Hypothesis Testing We would only run a bivariate probit if we believe $\rho \neq 0$. It is thus useful to test $H_0 : \rho = 0$. This can be done using a LR test approach. Under H_0,

$$\Pr(y_{i1} \cap y_{i2}) \;=\; \phi(q_{i1}\boldsymbol{x}_{i1}\boldsymbol{\beta}_1)\phi(q_{i2}\boldsymbol{x}_{i2}\boldsymbol{\beta}_2)$$

and

$$\ell_0 \;=\; \sum_i \left[\ln \phi(q_{i1}\boldsymbol{x}_{i1}\boldsymbol{\beta}_1) + \ln \phi(q_{i2}\boldsymbol{x}_{i2}\boldsymbol{\beta}_2)\right]$$

This is known as the *comparison log-likelihood*. We then compute

$$LR \;=\; 2(\ell_1 - \ell_0) \overset{asy}{\sim} \chi^2_1$$

where ℓ_1 is the log-likelihood function from the bivariate probit model. Under H_0, $\ell_1 = \ell_0$ and $LR = 0$. However, if H_0 is incorrect, then $\ell_1 > \ell_0$. If the scaled difference between the bivariate probit and comparison log-likelihoods is statistically significant, then we reject H_0.

It is also possible to perform a Wald test on H_0. Some programs will report this test statistic if the estimation was performed using robust or cluster-corrected standard errors. Here the estimate of ρ is divided by the estimated standard error and referred to a χ^2_1 distribution.

Example Imagine we would want to model the incidence of military coups and of political assassinations as two aspects of political violence in a country. Since it may be the case that the unobserved forces that drive coups may also be driving assassinations, a bivariate probit model seems appropriate. Stata allows one to estimate such a model using the `biprobit` command, whose syntax is[62]

```
biprobit depvar1 depvar2 [ indepvars ] [ , robust
cluster(varname) ]
```

Table 9.13 shows the results from the bivariate probit, using regime type (1 = civilian regime; 0 = otherwise), logged per capita GDP, and size of the military as predictors. The table also shows the marginal effects at the mean, which were obtained using the `mfx` command. We see that there is a significant positive correlation between the errors on military coups and political assassinations. The likelihood of a military coup decreases in civilian as opposed to other types of regimes. It also decreases as a function of per capita GDP. The probability that political assassinations occur is lower in civilian than in other regimes. Per capita GDP does not influence the incidence of assassinations. Size of the military is inconsequential for both response variables. Aggregating over both outcomes, we observe significant negative marginal effects for both regime type and per capita GDP.

[62] Stata estimates the hyperbolic tangent of ρ (see Chapter 5.8.4), which is not bounded. It then converts this estimate into an estimate of ρ.

Table 9.13: Bivariate Probit of Military Coups and Political Assassinations

	Coups		Assassinations		
Predictor	**B**	**S.E.**	**B**	**S.E.**	**MEM**
Civilian	-1.170^{**}	.096	$-.249^{*}$	.123	$-.033^{**}$
Military Size	$-.001$	.000	.000	.000	$-.000$
GDP	$-.536^{**}$	.116	.029	.089	$-.006^{**}$
Constant	.444	.290	-1.207^{**}	.235	
ρ	$.306^{**}$	.062			

Notes: $n = 3405$. $^{**}p < .01$, $^{*}p < .05$ (two-tailed). Table entries are maximum likelihood bivariate probit estimates with cluster-corrected estimated standard errors (clustering by country). Comparison log-pseudo-likelihood = -1550.776. Wald test on $H_0 : \rho = 0$ is 21.569, $p < .01$.

The Multivariate Probit Model

The bivariate probit model can be extended to more than two response variables (e.g. Chib and Greenberg 1998). Here we impose a multivariate normal distribution with correlation matrix

$$\boldsymbol{\Sigma} = \begin{bmatrix} 1 & \rho_{12} & \cdots & \rho_{1J} \\ \rho_{12} & 1 & \cdots & \rho_{2J} \\ \vdots & \vdots & \ddots & \vdots \\ \rho_{1J} & \rho_{2J} & \cdots & 1 \end{bmatrix}$$

where J is the number of response variables that is being modeled. The choice probabilities are given by

$$\Pr(y_{i1} \cap y_{i2} \cap \cdots \cap y_{iJ}) = \int_{A_{i1}} \cdots \int_{A_{iJ}} \phi_J(\epsilon_{i1}, \cdots, \epsilon_{iJ} | \boldsymbol{\Sigma}) d\epsilon_{i1} \cdots d\epsilon_{iJ}$$

where ϕ_J is the J-variate normal PDF and

$$A_{ij} = \begin{cases} \left(-\infty, \boldsymbol{x}_{ij}\boldsymbol{\beta}_j\right) & \text{if } y_{ij} = 1 \\ \left[\boldsymbol{x}_{ij}\boldsymbol{\beta}_j, \infty\right) & \text{if } y_{ij} = 0 \end{cases}$$

for $j = 1, \cdots, J$.

Estimation of the multivariate probit model is extremely complex. The likelihood function is given by

$$\begin{aligned}\mathcal{L} &= \prod_{i=1}^{n} \Pr(y_{i1} \cap \cdots \cap y_{iJ}) \\ &= \prod_{i=1}^{n} \int_{A_{i1}} \cdots \int_{A_{iJ}} \phi_J(\epsilon_{i1}, \cdots, \epsilon_{iJ}|\mathbf{\Sigma}) d\epsilon_{i1} \cdots d\epsilon_{iJ}\end{aligned}$$

Evaluation of the multivariate integrals is the problem here. The gradient contains a $J-1$-variate integral, which poses great computational difficulties. Chib and Greenberg (1998) discuss MCMC and MCEM (Monte Carlo EM) estimation of the multivariate probit model. The integrals could also be approximated using adaptive quadrature, although this tends to be extremely slow. Cappellari and Jenkins (2003) propose maximum simulated likelihood estimation, a procedure that is implemented in their Stata routine `mvprobit`. For a detailed discussion of the adaptive quadrature and maximum simulated likelihood solutions, the reader is referred to Chapter 11.

9.5 Appendix: The Delta Method*

When computing marginal effects and other useful statistics, we find ourselves working with functions of estimators. To draw inferences about such functions we need to know their limiting distributions. The delta method allows us to derive an approximate probability distribution of a function of an estimator based on knowledge of the limiting distribution of that estimator. Here we concentrate on cases in which the estimator itself is asymptotically normally distributed, as is the case for ML estimators. Then the limiting distribution of the function of the estimator is also normal and the key is simply to discover what the mean and variance of that distribution are.

For a single estimator and corresponding function, the following theorem presents the critical result about the limiting distribution.

> **Theorem:** If $\sqrt{n}(z_n - \mu) \xrightarrow{d} N(0, \sigma^2)$ and if $g(z_n)$ is a continuous function not involving n, then
>
> $$\sqrt{n}[g(z_n) - g(\mu)] \xrightarrow{d} N(0, [g'(\mu)]^2\sigma^2)$$

This result is based on the linear Taylor approximation about μ: $g(z_n) = g(\mu)+g'(\mu)(z_n-\mu)$.[63] Applying the rules for the variance of linear composites we then obtain $V[g(z_n)] = [g'(\mu)]^2 V[z_n] = [g'(\mu)]^2\sigma^2$.

The delta method can be extended to a set of functions and is then formulated as follows.

> **Theorem:** Let $\boldsymbol{z}_n$ be a sequence of K random variables such that $\sqrt{n}(\boldsymbol{z}_n - \boldsymbol{\mu}) \xrightarrow{d} N(\mathbf{0}, \boldsymbol{\Sigma})$. Let $\mathbf{c}(\boldsymbol{z}_n)$ be a set of J continuous functions of $\boldsymbol{z}_n$, which do not involve n. Then
>
> $$\sqrt{n}(\mathbf{c}(\boldsymbol{z}_n - \mathbf{c}(\boldsymbol{\mu})) \xrightarrow{d} N[\mathbf{0}, \mathbf{C}(\boldsymbol{\mu})\boldsymbol{\Sigma}\mathbf{C}(\boldsymbol{\mu})']$$

where $\mathbf{C}(\boldsymbol{\mu}) = \partial\mathbf{c}(\boldsymbol{\mu})/\partial\boldsymbol{\mu}'$ is a $J \times K$ matrix where the jth row contains the first partial derivatives of jth function with respect to $\boldsymbol{\mu}'$.

One should keep in mind that there is nothing sacred about using the linear Taylor approximation. This is done mostly for convenience but, in principle, the delta method could be expanded to higher-order Taylor series expansions. One may opt for these more complex approximations if one is worried about the accuracy of the linear approximation.

[63] In general, the Taylor series expansion of a function f in the vicinity of a is given by

$$\sum_{n=0}^{\infty} \frac{f^n(a)}{n!}(x-a)^n$$

where f^n is the n derivative of f.

Chapter 10

Models for Ordinal Outcomes

Ordinal response variables are extremely common in political analysis. In survey research, many response scales are ordinal in nature. This is true, for example, of the agree-disagree items that are widely used to measure political attitudes, beliefs, and values. In the field of international relations, many a research paper has been based on the Polity democracy and autocracy scores. While some have treated these scores as interval scales, they truly are ordinal scales. The same can be said of Freedom House scores and of most measures of corruption and the rule of law that have found their way into comparative politics. Thus, it is probably safe to say that ordinal measures dominate in political research.

How should one analyze these measures? There remains a strong current in political science that treats ordinal scales as if they were continuous. Advocates of this approach rely on the linear regression model to model ordinal response variables, thus imposing a linear effect on the covariates. As widespread as this approach is, it is also extremely risky. Unless the ordinal scale has many categories, it is easy to obtain misleading results (McKelvey and Zavoina 1975; Winship and Mare 1984).

It is not difficult to find the reasons why the linear regression model is inappropriate for ordinal outcomes. First, it is inherently suspect to assume normality, as the classical normal linear regression model does, when the dependent variable is ordinal in nature. Second, it is well-known that the association between an ordinal outcome and a predictor may be underestimated by traditional measures of association, such as Pearson's product-moment correlation, that underlie the regression model. This is especially true if the response scale has few categories and the distribution over those categories is

skewed (e.g. Jöreskog 1990). Finally, the linear regression model may impose the incorrect functional form on the relationship between the response variable and the covariates. In the linear regression model, the effect of a unit increase in a covariate is constant along the y scale. This makes sense when the differences on this scale may be interpreted as identical, as is the case with continuous dependent variables. However, on an ordinal response scale, the differences between adjacent scale values need not to represent identical distances. As such, there is no reason to expect that a unit increase in a covariate has a constant effect on the response variable. Imposing this as a constraint could easily produce biased parameter estimates.

There is, then, considerable peril in analyzing ordinal response variables, especially those with few categories, as if they are continuous. Instead of analyzing such variables via linear regression analysis, we should consider alternative models. The ordered logit and probit models are useful starting points for this purpose.

10.1 Ordered Logit and Probit Analysis

10.1.1 A Motivating Example

Imagine a survey respondent who is faced with the following well-known survey item:

> Generally speaking, do you consider yourself a Democrat, a Republican, or an independent?

We can think of the three response options as being rank-ordered with respect to the property "Republican-ness," i.e. $D < I < R$. It is the respondent's task to choose a particular response to the survey question. We indicate his/her response by

$$y_i = \begin{cases} 1 & \text{Democrat} \\ 2 & \text{Independent} \\ 3 & \text{Republican} \end{cases}$$

We can think of the 3-point response scale as a crude reflection of an underlying continuous latent variable that taps an individual's "Republican-ness" with infinite precision. We shall label this latent variable as y_i^*, just as we did for the binary regression models. The latent response scale ranges

from $-\infty$ to ∞. Low values on y_i^* indicate a low level of Republican affiliation, while high values indicate a high level of this property. As before, we model y_i^* as a function of systematic and stochastic components, i.e.

$$y_i^* = \boldsymbol{x}_i\boldsymbol{\beta} + \epsilon_i$$

where $\boldsymbol{x}_i$ is a vector of respondent attributes (e.g. income, race, gender). This vector does *not* include a constant. The stochastic term, ϵ_i, captures unobserved factors that make an individual more or less Republican. These may be things of which the respondent is him or her self unaware, or things that remain hidden from the modeler.

We now need a mechanism that connects y_i^* to y_i. Imagine a set of two **cut points** or **thresholds** on y_i^*. The first cut point is τ_1 and it indicates the level of y_i^* that is required to embrace at least the second response option ("independent"). That is, respondents whose level of Republican affiliation falls short of τ_1 are expected to embrace the first response option, while those respondents whose level is at least τ_1 will, at a minimum, indicate that they are independents. The second cut point, $\tau_2 > \tau_1$, indicates the level of y_i^* that is required to embrace the third response option. That is, only respondents whose level of Republican identification is at least τ_2 will respond that they are Republicans; everyone else is expected to indicate that they are independents or, if y_i^* falls short even of τ_1, Democrats. Thus,

$$y_i = \begin{cases} 1 & \text{if } -\infty < y_i^* < \tau_1 \\ 2 & \text{if } \tau_1 \leq y_i^* < \tau_2 \\ 3 & \text{if } \tau_2 \leq y_i^* < \infty \end{cases}$$

Substituting the model for y_i^* this may also be written as

$$y_i = \begin{cases} 1 & \text{if } -\infty < \boldsymbol{x}_i\boldsymbol{\beta} + \epsilon_i < \tau_1 \\ 2 & \text{if } \tau_1 \leq \boldsymbol{x}_i\boldsymbol{\beta} + \epsilon_i < \tau_2 \\ 3 & \text{if } \tau_2 \leq \boldsymbol{x}_i\boldsymbol{\beta} + \epsilon_i < \infty \end{cases}$$

or, subtracting $\boldsymbol{x}_i\boldsymbol{\beta}$ from both sides of the inequality signs,

$$y_i = \begin{cases} 1 & \text{if } -\infty < \epsilon_i < \tau_1 - \boldsymbol{x}_i\boldsymbol{\beta} \\ 2 & \text{if } \tau_1 - \boldsymbol{x}_i\boldsymbol{\beta} \leq \epsilon_i < \tau_2 - \boldsymbol{x}_i\boldsymbol{\beta} \\ 3 & \text{if } \tau_2 - \boldsymbol{x}_i\boldsymbol{\beta} \leq \epsilon_i < \infty \end{cases}$$

The presence of the stochastic term ϵ_i in the mechanism linking y_i to y_i^* means that the mechanism of necessity works probabilistically. Accordingly,

$\Pr(y_i = 1) = \Pr(-\infty < \epsilon_i < \tau_1 - \boldsymbol{x}_i\boldsymbol{\beta})$, $\Pr(y_i = 2) = \Pr(\tau_1 - \boldsymbol{x}_i\boldsymbol{\beta} \leq \epsilon_i < \tau_2 - \boldsymbol{x}_i\boldsymbol{\beta})$, and $\Pr(y_i = 3) = \Pr(\tau_2 - \boldsymbol{x}_i\boldsymbol{\beta} \leq \epsilon_i < \infty)$. To work out these probabilities, we need to make a distributional assumption for ϵ_i. A convenient starting point is to impose a symmetric probability distribution such as the standard logistic or standard normal distribution. Doing so produces the following probability for responding "Democrat:"

$$\begin{aligned}
\Pr(y_i = 1) &= \Pr((-\infty < \epsilon_i < \tau_1 - \boldsymbol{x}_i\boldsymbol{\beta}) \\
&= \int_{-\infty}^{\tau_1 - \boldsymbol{x}_i\boldsymbol{\beta}} f(\epsilon_i) d\epsilon_i \\
&= F(\tau_1 - \boldsymbol{x}_i\boldsymbol{\beta})
\end{aligned}$$

Here $f(.)$ is the PDF, while $F(.)$ is the CDF. Similarly the probabilities of responding "independent" and "Republican" are given by:

$$\begin{aligned}
\Pr(y_i = 2) &= \Pr(\tau_1 - \boldsymbol{x}_i\boldsymbol{\beta} \leq \epsilon_i < \tau_2 - \boldsymbol{x}_i\boldsymbol{\beta}) \\
&= \int_{-\infty}^{\tau_2 - \boldsymbol{x}_i\boldsymbol{\beta}} f(\epsilon_i) d\epsilon_i - \int_{-\infty}^{\tau_1 - \boldsymbol{x}_i\boldsymbol{\beta}} f(\epsilon_i) d\epsilon_i \\
&= F(\tau_2 - \boldsymbol{x}_i\boldsymbol{\beta}) - F(\tau_1 - \boldsymbol{x}_i\boldsymbol{\beta}) \\
\Pr(y_i = 3) &= \Pr(\tau_2 - \boldsymbol{x}_i\boldsymbol{\beta} \leq \epsilon_i < \infty) \\
&= 1 - \Pr(\epsilon_i < \tau_2 - \boldsymbol{x}_i\boldsymbol{\beta}) \\
&= 1 - \int_{-\infty}^{\tau_2 - \boldsymbol{x}_i\boldsymbol{\beta}} f(\epsilon_i) d\epsilon_i \\
&= 1 - F(\tau_2 - \boldsymbol{x}_i\boldsymbol{\beta})
\end{aligned}$$

The probabilistic choice mechanism is illustrated in Figure 10.1.

When $F(.)$ is chosen to be the standard logistic distribution, $\Lambda(.)$, then the equations for $\Pr(y_i = 1) \cdots \Pr(y_i = 3)$ specify the choice probabilities of the **ordered logit model**. If, instead, $f(.)$ is chosen to be the standard normal distribution, $\Phi(.)$, then these equations specify the choice probabilities of the **ordered probit model**. The differences between these two models are fairly minimal, although they make different variance assumptions, so that the scale of the parameters is different as well. Note that one could have chosen an asymmetric error distribution as well, but this is not commonly done for ordinal regression models.

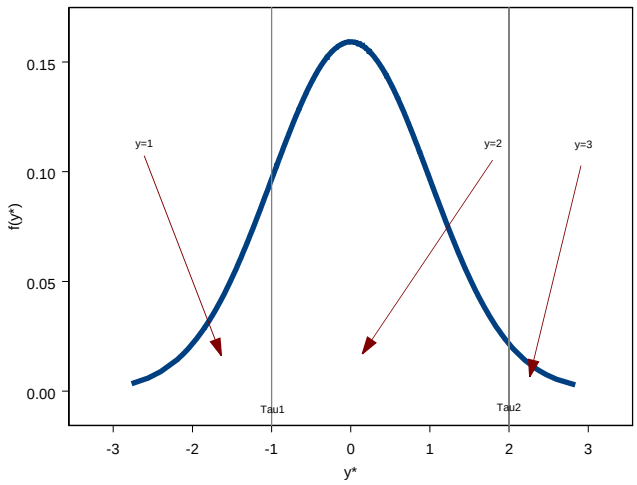

Figure 10.1: Choice Mechanism in Ordinal Regression Models

10.1.2 General Model

In general, imagine that an individual i chooses from among M alternatives, $1, 2, \cdots, M$, that are rank-ordered with respect to the underlying latent trait y_i^*. The latent trait is continuous and can be modeled via

$$y_i^* \quad = \quad \boldsymbol{x}_i\boldsymbol{\beta} + \epsilon_i \tag{10.1}$$

Here $\boldsymbol{x}_i$ is a vector of attributes of the individual and does not include a constant. Further, ϵ_i is a stochastic component.

Alternative m is chosen if

$$\tau_{m-1} \leq y_i^* < \tau_m \quad = \quad \tau_{m-1} - \boldsymbol{x}_i\boldsymbol{\beta} \leq \epsilon_i < \tau_m - \boldsymbol{x}_i\boldsymbol{\beta}$$

where the τ's are ancillary parameters and the following conditions hold: $\tau_{m-1} < \tau_m$, $\tau_0 = -\infty$, and $\tau_M = \infty$. In probabilistic terms

$$\Pr(y_i = m) \quad = \quad F(\tau_m - \boldsymbol{x}_i\boldsymbol{\beta}) - F(\tau_{m-1} - \boldsymbol{x}_i\boldsymbol{\beta}) \tag{10.2}$$

where $F(.)$ is specified either as the standard logistic or the standard normal CDF. Specifically, the choice probabilities of the ordered logit model are given by

$$\Pr(y_i = m) \quad = \quad \Lambda(\tau_m - \boldsymbol{x}_i\boldsymbol{\beta}) - \Lambda(\tau_{m-1} - \boldsymbol{x}_i\boldsymbol{\beta})$$

while those of the ordered probit model are given by

$$\Pr(y_i = m) \quad = \quad \Phi(\tau_m - \boldsymbol{x}_i\boldsymbol{\beta}) - \Phi(\tau_{m-1} - \boldsymbol{x}_i\boldsymbol{\beta})$$

10.1.3 Estimation

Identification

As was the case with the binary logit and probit models, the latent variable y_i^* is unobserved and does not have a fixed measurement scale. This means that the scale of the parameters in $\boldsymbol{\beta}$ is also unidentified, making it impossible to estimate the ordered logit and probit models without making some identifying assumptions. We identify the scale of y_i^* and hence of $\boldsymbol{\beta}$ by fixing the variance of ϵ_i. In the ordered probit model, $V[\epsilon_i] = 1$; in the ordered logit model, $V[\epsilon_i] = \frac{\pi^2}{3}$. Because the variances (and other features) of the standard normal and standard logistic distributions are different, the scale of the ordered probit and logit coefficients is as well. In general, the ordered logit estimates are about 1.7 times the size of the ordered logit coefficients.

Another identifying constraint is that we need to exclude the constant from $\boldsymbol{x}_i$. It is easy to see why. In an empty model, in which there are no predictors, the only information that we have are estimates of $M - 1$ probabilities (the remaining probability is not unique, since it follows from the axiom that all probabilities must sum to 1). In this empty model, there are $M - 1$ parameters in the form of cut points. This means that the model is just-identified. But if we add a constant, then the number of estimated parameters increases to M. With $M - 1$ pieces of information, this means that the model is no longer identified. Thus, if one adds a constant, as Greene (2003) proposes, then one should drop one of the cut points.[1]

Likelihood and Log-Likelihood Functions

The ordered logit and probit models are estimated using MLE. The likelihood function is derived by recognizing that we draw a sample of n independent observations that yields observed choices $1, 2 \cdots M$. Focusing on a particular alternative m, the joint density is given by

$$\prod_{y_i=m} \Pr(y_i = m) \quad = \quad \prod_{y_i=m} [F(\tau_m - \boldsymbol{x}_i\boldsymbol{\beta}) - F(\tau_{m-1} - \boldsymbol{x}_i\boldsymbol{\beta})]$$

[1] Whether one estimates $M - 2$ cut points and a constant or $M - 1$ cut points and no constant is immaterial for the estimated effects of the substantive predictors in $\boldsymbol{x}_i$.

Aggregation across all of the alternatives then yields the following likelihood:

$$\mathcal{L} = \prod_{m=1}^{M} \prod_{y_i=m} [F(\tau_m - \boldsymbol{x}_i\boldsymbol{\beta}) - F(\tau_{m-1} - \boldsymbol{x}_i\boldsymbol{\beta})]$$

This can be written more elegantly when we create an indicator for the alternative that was selected:

$$z_{im} = \begin{cases} 1 & \text{if } y_i = m \\ 0 & \text{if } y_i \neq m \end{cases}$$

Now,

$$\mathcal{L} = \prod_i \prod_m [F(\tau_m - \boldsymbol{x}_i\boldsymbol{\beta}) - F(\tau_{m-1} - \boldsymbol{x}_i\boldsymbol{\beta})]^{z_{im}} \tag{10.3}$$

The log-likelihood function is

$$\ell = \sum_i \sum_m z_{im} \ln [F(\tau_m - \boldsymbol{x}_i\boldsymbol{\beta}) - F(\tau_{m-1} - \boldsymbol{x}_i\boldsymbol{\beta})] \tag{10.4}$$

where we substitute $\Lambda(.)$ for $F(.)$ in the case of ordered logit analysis and $\Phi(.)$ for $F(.)$ in the case of ordered probit analysis.

The ordered logit/probit log-likelihood is relatively straightforward and optimization of this function generally poses few problems. The first-order conditions are

$$\frac{\partial \ell}{\partial \boldsymbol{\beta}'} = \sum_i \sum_m z_{im} \frac{f(\tau_{m-1} - \boldsymbol{x}_i\boldsymbol{\beta}) - f(\tau_m - \boldsymbol{x}_i\boldsymbol{\beta})}{F(\tau_m - \boldsymbol{x}_i\boldsymbol{\beta}) - F(tau_{m-1} - \boldsymbol{x}_i\boldsymbol{\beta})} \boldsymbol{x}_i = \mathbf{0}$$

$$\frac{\partial \ell}{\partial \tau_k} = \sum_i \sum_m z_{im} \frac{\delta_{m,k} f(\tau_m - \boldsymbol{x}_i\boldsymbol{\beta}) - \delta_{m-1,k} f(\tau_{m-1} - \boldsymbol{x}_i\boldsymbol{\beta})}{F(\tau_m - \boldsymbol{x}_i\boldsymbol{\beta}) - F(\tau_{m-1} - \boldsymbol{x}_i\boldsymbol{\beta})} = 0$$

where $\delta_{m,k} = 1$ if $m = k$ and $\delta_{m,k} = 0$ if $m \neq k$ is the Kronecker delta (see Maddala 1983). While these first-order conditions will have to be evaluated numerically, they are not complex and convergence should be speedy.

10.1.4 Interpretation

As with the binary logit and probit models, the ordered logit and probit coefficients cannot be directly interpreted since the effect of a unit increase in a predictor depends on that predictor's initial value. Various interpretation methods exist, including marginal effects, predicted probabilities and discrete changes in predicted probabilities and, for the ordered logit model, odds ratios.

Marginal Effects and Elasticities

Marginal Effects One useful way of interpreting the coefficients from an ordered logit or probit model is to compute the marginal effects of predictors. The marginal effect of a predictor x_k is defined as[2]

$$\frac{\partial \pi_m}{\partial x_k} = [f(\tau_{m-1} - \boldsymbol{x}\boldsymbol{\beta}) - f(\tau_m - \boldsymbol{x}\boldsymbol{\beta})]\,\beta_k \tag{10.5}$$

where $\pi_m = \Pr(y = m)$ and $f(.)$ is the standard logistic or standard normal PDF. This expression is the slope of the curve that relates $\Pr(y = m|\boldsymbol{x})$ to x_k. It can be interpreted as the instantaneous change in the probability of choosing alternative m for a very small change in x_k.

In order to evaluate the marginal effects, we need to set the covariates to a particular value. There are two ways of doing this. When computing **marginal effects at the mean** (MEM), we set all of the covariates equal to their sample means. Thus,

$$MEM_k = [f(\tau_{m-1} - \bar{\boldsymbol{x}}\boldsymbol{\beta}) - f(\tau_m - \bar{\boldsymbol{x}}\boldsymbol{\beta})]\,\beta_k$$

When computing **average marginal effects** (AME), we use the sample unit's actual values on the covariates. Thus, a unique marginal effect is computed for each unit. The AME is then the average of those idiosyncratic effects:

$$AME_k = \frac{1}{n}\sum_{i=1}^{n} [f(\tau_{m-1} - \boldsymbol{x}_i\boldsymbol{\beta}) - f(\tau_m - \boldsymbol{x}_i\boldsymbol{\beta})]\,\beta_k$$

[2] *Proof:* Let $\pi_m = \Pr(y = m)$, then

$$\begin{aligned}\frac{\partial \pi_m}{\partial x_k} &= \frac{\partial F(\tau_m - \boldsymbol{x}\boldsymbol{\beta})}{\partial x_k} - \frac{\partial F(\tau_{m-1} - \boldsymbol{x}\boldsymbol{\beta})}{\partial x_k} \\ &= \frac{dF(z_m)}{dz_m}\frac{\partial z_m}{\partial x_k} - \frac{dF(z_{m-1})}{dz_{m-1}}\frac{\partial z_{m-1}}{\partial x_k}\end{aligned}$$

Here we rely on the chain rule and define $z_m = \tau_m - \boldsymbol{x}\boldsymbol{\beta}$ and $z_{m-1} = \tau_{m-1} - \boldsymbol{x}\boldsymbol{\beta}$. It is easily demonstrated that $dF(z_m)/dz_m = f(z_m)$ and $dF(z_{m-1})/dz_{m-1} = f(z_{m-1})$. Further, $\partial z_m/\partial x_k = -\beta_k$ and $\partial z_{m-1}/\partial x_k = -\beta_k$. Substituting these results yields

$$\frac{\partial \pi_m}{\partial x_k} = -\beta_k f(\tau_m - \boldsymbol{x}\boldsymbol{\beta}) - -\beta_k f(\tau_{m-1} - \boldsymbol{x}_k)$$

Rearranging terms yields (10.5).

One of the advantages of AME is that we do not rely upon unrealistic values for the covariates, since AME is based on values that actually occurred in the sample.

Grouping the marginal effects of all predictors into a single vector, $\boldsymbol{\gamma} = [f(\tau_{m-1} - \boldsymbol{x\beta}) - f(\tau_m - \boldsymbol{x\beta})]\,\boldsymbol{\beta}$, the variance-covariance matrix of the estimated marginal effects can be computed via the delta method:

$$\mathbf{V}[\hat{\gamma}] \;=\; \left[\frac{\partial \hat{\boldsymbol{\gamma}}}{\partial \hat{\boldsymbol{\beta}}'}\right] \mathbf{V}[\hat{\boldsymbol{\beta}}] \left[\frac{\partial \hat{\boldsymbol{\gamma}}}{\partial \hat{\boldsymbol{\beta}}'}\right]'$$

where

$$\begin{aligned}\frac{\partial \hat{\boldsymbol{\gamma}}}{\partial \hat{\boldsymbol{\beta}}'} \;=\; & \left[f(\hat{\tau}_{m-1} - \boldsymbol{x}\hat{\boldsymbol{\beta}}) - f(\hat{\tau}_m - \boldsymbol{x}\hat{\boldsymbol{\beta}})\right]\mathbf{I} + \\ & \left[f'(\hat{\tau}_m - \boldsymbol{x}\hat{\boldsymbol{\beta}}) - f'(\hat{\tau}_{m-1} - \boldsymbol{x}\hat{\boldsymbol{\beta}})\right]\boldsymbol{\beta x}\end{aligned}$$

where $f'(.)$ is the first derivative of the PDF. In the ordered probit model, $f(\tau_k - \boldsymbol{x\beta}) = \phi(\tau_k - \boldsymbol{x\beta})$ and $f'(tau_k - \boldsymbol{x\beta}) = (\boldsymbol{x\beta} - \tau_k)\phi(\tau_k - \boldsymbol{x\beta})$, for $k = m, m-1$. In the ordered logit model, $f(\tau_k - \boldsymbol{x\beta}) = \lambda(\tau_k - \boldsymbol{x\beta}) = \Lambda(\tau_k - \boldsymbol{x\beta})\,[1 - \Lambda(\tau_k - \boldsymbol{x\beta}]$ and $f'(\tau_k - \boldsymbol{x\beta}) = 1 - 2\Lambda(\tau_k - \boldsymbol{x\beta})$. With a measure of the sampling variability of the marginal effects in place, one can now construct confidence intervals or perform hypothesis tests. Specifically, the $100(1-\alpha)\%$ confidence interval for the marginal effect of a predictor x_k is given by

$$\hat{\gamma}_k - z_{\alpha/2} s.e._{\hat{\gamma}_k} \leq \gamma_k \leq \hat{\gamma}_k + z_{\alpha/2} s.e._{\hat{\gamma}_k}$$

As was the case with the binary logit and probit models, extreme care should be taken with the interpretation of **nonlinear models** and **interaction effects** in ordinal regression model. First, imagine that we are estimating an ordered logit/probit model where the underlying latent variable is modeled as $y_i^* = \beta_1 x_i + \beta_2 x_i^2 + \boldsymbol{x}_i\boldsymbol{\beta} + \epsilon_i$. With some calculus, it can then be shown that the marginal effect is given by

$$\frac{\partial \pi_m}{\partial x_k} \;=\; [f_{m-1} - f_m]\,(\beta_1 + 2\beta_2 x)$$

where $f_m = f(\tau_m - \beta_1 x - \beta_2 x^2 - \boldsymbol{x\beta})$, $f_{m-1} = f(\tau_{m-1} - \beta_1 x - \beta_2 x^2 - \boldsymbol{x\beta})$. Now consider an ordered logit/probit model where $y_i^* = \beta_1 x_i + \beta_2 z_i + \beta_{12} x_i z_i + \boldsymbol{x}_i\boldsymbol{\beta}$. Here, the marginal effect is given by

$$\frac{\partial \pi_m}{\partial x \partial z} \;=\; (f_{m-1} - f_m)\,\beta_{12} + \left(f'_m - f'_{m-1}\right)(\beta_1 + \beta_{12} z)\,(\beta_2 + \beta_{12} x)$$

where $f_m = f(\tau_m - \beta_1 x - \beta_2 z - \beta_{12} xz - \boldsymbol{x\beta})$, $f_{m-1} = f(\tau_{m-1} - \beta_1 x - \beta_2 z - \beta_{12} xz - \boldsymbol{x\beta})$, $f'_m = f'(\tau_m - \beta_1 x - \beta_2 z - \beta_{12} xz - \boldsymbol{x\beta})$, and $f'_{m-1} = f'(\tau_{m-1} - \beta_1 x - \beta_2 z - \beta_{12} xz - \boldsymbol{x\beta})$. This equation implies that a non-zero interaction effect can exist even when $\beta_{12} = 0$. Moreover, the sign of the interaction effect depends on more than the sign of β_{12} alone.

Elasticities Marginal effects can be transformed into elasticities by multiplying them through by a factor x_{ik}/π_m:

$$\kappa_k = \frac{\partial \pi_m}{\partial x_{ik}} \frac{x_{ik}}{\pi_m}$$

The advantage of elasticities is that they have a straightforward interpretation, namely that a one percentage point change in the predictor changes the probability of choosing alternative m by κ_k percentage points.

Discrete Change in Predicted Probabilities

Predicted Probabilities Marginal effects are not the only way to interpret the results from an ordered logit or probit model. Predicted probabilities provide a useful alternative to marginal effects, one that has the advantage that it can be applied to both continuous and dummy predictors. Let $\pi_{im} = \Pr(y_i = m)$, then the estimated predicted probability is equal to

$$\hat{\pi}_{im} = F(\hat{\tau}_m - \boldsymbol{x}_i\hat{\boldsymbol{\beta}}) - F(\hat{\tau}_{m-1} - \boldsymbol{x}_i\hat{\boldsymbol{\beta}}) \tag{10.6}$$

Typically, we set the predictors to their means to compute the predicted probabilities, although other simulation scenarios are possible. A measure of the sampling variability in $\hat{\pi}_i$ can be calculated via the delta method:

$$V[\hat{\pi}_{im}] = \left[f(\hat{\tau}_{m-1} - \boldsymbol{x}_i\hat{\boldsymbol{\beta}}) - f(\hat{\tau}_m - \boldsymbol{x}_i\hat{\boldsymbol{\beta}})\right]^2 \boldsymbol{x}_i \mathbf{V}[\hat{\boldsymbol{\beta}}] \boldsymbol{x'}_i$$

Knowledge of this sampling variance allows one to compute test statistics and confidence intervals on the predicted probabilities.

Discrete Change While predicted probabilities give a sense of the probability of a particular choice given a certain profile of covariate values, they

do not directly speak to the impact of a change in one of the covariates. For this, one can compute a discrete change measure:

$$\frac{\Delta \pi_m}{\Delta x_k} = \Pr(y = m|\boldsymbol{x}, x_k + \delta) - \Pr(y = m|\boldsymbol{x}, x_k) \tag{10.7}$$

It is conventional to set all of the remaining covariates to their means, so that the discrete change is computed as $\Pr(y = 1|\bar{\boldsymbol{x}}, x_k + \delta) - \Pr(y = 1|\bar{\boldsymbol{x}}, x_k)$. However, other covariate values may be chosen as well, as long as they remain constant across the values of y and the simulated change in x_k.

The change in x_k can be set at different values. Common choices include the following.

1. A **unit change** relative to $\bar{x}_k$, so that $\delta = 1$ and $\Delta\pi_m/\Delta x_k = \Pr(y = m|\bar{\boldsymbol{x}}, \bar{x}_k + 1) - \Pr(y = m|\bar{\boldsymbol{x}}, \bar{x}_k)$. This is tantamount to setting all covariates to their mean values initially and then increasing only x_k by one unit.

2. A **centered unit change** relative to $\bar{x}_k$, so that x_k moves from $\bar{x}_k - .5$ to $\bar{x}_k + .5$. The discrete change in predicted probability is then $\Delta\pi_m/\Delta x_k = \Pr(y = m|\bar{\boldsymbol{x}}, \bar{x}_k + .5) - \Pr(y = m|\bar{\boldsymbol{x}}, \bar{x}_k - .5)$.

3. A **standard deviation change** relative to $\bar{x}_k$, so that x_k moves from $\bar{x}_k - .5s_k$ to $\bar{x}_k + .5s_k$, where s_k is the sample standard deviation of x_k. The discrete change is then defined as $\Delta\pi_m/\Delta x_k = \Pr(y = m|\bar{\boldsymbol{x}}, \bar{x}_k + .5s_k) - \Pr(y = m|\bar{\boldsymbol{x}}, \bar{x}_k - .5s_k)$.

4. It is also possible to compute the **maximum possible change**. Here we let x_k move from the minimum to the maximum in the sample, so that δ is equal to the range and the discrete change is defined as $\Delta\pi_m/\Delta x_k = \Pr[y = m|\bar{\boldsymbol{x}}, \max(x_k)] - \Pr[y = m|\bar{\boldsymbol{x}}, \min(x_k)]$. This manner of computing discrete change is quite common in political analysis, although it is sometimes criticized for creating an exaggerated view of the effects of predictors because very few cases may fall at the extremes.

5. For dummy predictors, computing the maximum possible change is tantamount to computing a **change from 0 to 1**. Thus, $\Delta\pi_m/\Delta x_k = \Pr(= m|\bar{\boldsymbol{x}}, x_k = 1) - \Pr(y = m|\bar{\boldsymbol{x}}, x_k = 0)$. This computation is not subject to the same criticism as maximum possible change. On the contrary, this is actually one of the best ways to characterize the effect of dummy predictors.

6. **Percentile change** provides another method of operationalizing discrete change. Here, we let x_k move from the pth percentile to the $100-p$th percentile, e.g. from the 10th to the 90th percentile. Letting $x_{k,p}$ and $x_{k,100-p}$ denote the values of x_k that correspond to the pth and $100-p$th percentiles, respectively, discrete change is defined as $\Delta\pi_m/\Delta x_k = \Pr(y=m|\bar{\boldsymbol{x}}, x_{k,100-p}) - \Pr(y=m|\bar{\boldsymbol{x}}, x_{k,p})$.

One should pick the change in x_k that best captures a change that is substantively interesting. Once this change has been selected, it should be applied to all values of y.

In order to compute confidence intervals for the discrete change measures one needs an estimate of the standard errors. Let $\boldsymbol{x}_a$ and $\boldsymbol{x}_b$ denote two sets of values of the covariates, e.g. $\boldsymbol{x}_a = \bar{\boldsymbol{x}}, \max(x_k)$ and $\boldsymbol{x}_b = \bar{\boldsymbol{x}}, \min(x_k)$. Then the discrete change is given by $\Delta\pi_m/\Delta x = \Pr(y=m|\boldsymbol{x}_a) - \Pr(y=m|\boldsymbol{x}_b) = [F(\tau_m - \boldsymbol{x}_a\boldsymbol{\beta}) - F(\tau_{m-1} - \boldsymbol{x}_a\boldsymbol{\beta})] - [F(\tau_m - \boldsymbol{x}_b\boldsymbol{\beta}) - F(\tau_{m-1} - \boldsymbol{x}_b\boldsymbol{\beta})]$. The variance of this change can be computed via the delta method, which yields

$$V\left[\frac{\Delta\hat{\pi}_m}{\Delta x}\right] = \left[\frac{\partial\Delta\hat{\pi}_m/\Delta x}{\partial\hat{\boldsymbol{\beta}}'}\right]\mathbf{V}[\hat{\boldsymbol{\beta}}]\left[\frac{\partial\Delta\hat{\pi}_m/\Delta x}{\partial\hat{\boldsymbol{\beta}}'}\right]'$$

where

$$\begin{aligned}\left[\frac{\partial\Delta\hat{\pi}_m/\Delta x}{\partial\hat{\boldsymbol{\beta}}'}\right] &= \left[f(\hat{\tau}_{m-1} - \boldsymbol{x}_a\hat{\boldsymbol{\beta}}) - f(\hat{\tau}_m - \boldsymbol{x}_a\hat{\boldsymbol{\beta}})\right]\boldsymbol{x}_a - \\ &\quad \left[f(\hat{\tau}_{m-1} - \boldsymbol{x}_b\hat{\boldsymbol{\beta}}) - f(\hat{\tau}_m - \boldsymbol{x}_b\hat{\boldsymbol{\beta}})\right]\boldsymbol{x}_b\end{aligned}$$

Computing these variances can be computationally intensive but fortunately software is available that will perform the necessary operations.

With predicted probabilities, as with marginal effects, nonlinear models and interactions require special attention. Consider, for example, the latent variable model $y^* = \beta_1 x + \beta_2 z + \beta_{12} xz + \boldsymbol{x}\boldsymbol{\beta}$, where x and z are both dummy variables. The discrete change in predicted probabilities, holding all else at the mean, is given by

$$\begin{aligned}\frac{\Delta\pi_m}{\Delta x\Delta z} &= [F(\tau_m - \beta_1 - \beta_2 - \beta_{12} - \bar{\boldsymbol{x}}\boldsymbol{\beta}) - \\ &\quad F(\tau_{m-1} - \beta_1 - \beta_2 - \beta_{12} - \bar{\boldsymbol{x}}\boldsymbol{\beta})] - \\ &\quad [F(\tau_m - \beta_2 - \bar{\boldsymbol{x}}) - F(\tau_m - \beta_2 - \bar{\boldsymbol{x}}\boldsymbol{\beta})] - \\ &\quad [F(\tau_m - \beta_1 - \bar{\boldsymbol{x}}) - F(\tau_m - \beta_1 - \bar{\boldsymbol{x}}\boldsymbol{\beta})] + \\ &\quad [F(\tau_m - \bar{\boldsymbol{x}}) - F(\tau_m - \bar{\boldsymbol{x}}\boldsymbol{\beta})]\end{aligned}$$

One issue that arises with the discrete change measure is that there are as many discrete changes as there are values of y. Long (1997) proposes to summarize all those changes in what he refers to as the **average absolute discrete change**, i.e. the average of the absolute values of the discrete changes in predicted probabilities. The formula is given by

$$\bar{\Delta} = \frac{1}{M}\sum_{m=1}^{M}\left|\frac{\Delta\pi_m}{\Delta x_k}\right| \tag{10.8}$$

The reason for taking the absolute values is that $\sum_m \Delta\pi_m/\Delta x_k = 0$, i.e. the discrete changes cancel each other out. This is not true of the absolute values of the discrete changes.

Cumulative Odds Ratios

The cumulative odds ratio is a particular useful tool for interpreting ordered logit models. It is based on the notion of cumulative odds, which is defined as[3]

$$\begin{aligned}\Omega_m &= \frac{\Pr(y_i \leq m|\boldsymbol{x}_i)}{\Pr(y_i > m|\boldsymbol{x}_i)} \\ &= \exp\left(\tau_m - \boldsymbol{x}_i\boldsymbol{\beta}\right)\end{aligned} \tag{10.9}$$

[3] *Proof:* In the ordered logit model,

$$\Pr(y_i \leq m) = \frac{\exp\left(\tau_m - \boldsymbol{x}_i\boldsymbol{\beta}\right)}{1+\exp\left(\tau_m - \boldsymbol{x}_i\boldsymbol{\beta}\right)}$$

Further,

$$\Pr(y_i > m) = 1 - \Pr(y_i \leq m) = \frac{1}{1+\exp\left(\tau_m - \boldsymbol{x}_i\boldsymbol{\beta}\right)}$$

Thus, the cumulative odds are defined as

$$\begin{aligned}\Omega_m &= \frac{\Pr(y_i \leq m)}{1-\Pr(y_i \leq m)} \\ &= \frac{\exp\left(\tau_m - \boldsymbol{x}_i\boldsymbol{\beta}\right)}{1+\exp\left(\tau_m - \boldsymbol{x}_i\boldsymbol{\beta}\right)}\frac{1+\exp\left(\tau_m - \boldsymbol{x}_i\boldsymbol{\beta}\right)}{1} \\ &= \exp\left(\tau_m - \boldsymbol{x}_i\boldsymbol{\beta}\right)\end{aligned}$$

Thus, the cumulative odds show the probability of choosing alternative m, or a lower ranked alternative, versus the probability of choosing a higher ranked alternative.

We can evaluate the cumulative odds at value x_k of a predictor and then again at value $x_k + \delta$, while holding all other predictors constant. This produces the cumulative odds ratio:[4]

$$\frac{\Omega_m(x_k + \delta, \boldsymbol{x})}{\Omega_m(x_k, \boldsymbol{x})} = \exp(-\beta_k \delta) \tag{10.10}$$

where β_k is the effect associated with the predictor and $\boldsymbol{x}$ contains all of the remaining predictors (which are held constant). We can interpret this as follows: for a δ unit increase in x_k, the odds of an outcome being less than or equal to m changes by a factor $\exp(-\beta_k \delta)$, holding all other predictors constant. For interpretation purposes, δ is frequently set equal to 1.

It is sometimes useful to transform the factor change measure into a percentage change measure. This can be done via

$$100 \times \frac{\Omega_m(x_k + \delta, \boldsymbol{x}) - \Omega_m(x_k, \boldsymbol{x})}{\Omega_m(x_k, \boldsymbol{x})} = 100 \times [\exp(-\beta_k \delta) - 1]$$

This measure shows the percentage by which the cumulative odds are increased or decreased.

An important property of the cumulative odds ratio should be noted. If one considers (10.10), it is clear that the right-hand side does not contain a

[4] *Proof:* At x_k, the cumulative odds can be written as

$$\begin{aligned} \Omega_m(x_k, \boldsymbol{x}) &= \exp(\tau_m - \beta_k x_k - \boldsymbol{x}\boldsymbol{\beta}) \\ &= \exp(\tau_m) \exp(-\beta_k x_k) \exp(-\boldsymbol{x}\boldsymbol{\beta}) \end{aligned}$$

At $x_k + \delta$, the cumulative odds can be written as

$$\begin{aligned} \Omega_m(x_k + \delta, \boldsymbol{x}) &= \exp(\tau_m - \beta_k(x_k + \delta) - \boldsymbol{x}\boldsymbol{\beta}) \\ &= \exp(\tau_m) \exp(-\beta_k x_k) \exp(-\beta_k \delta) \exp(-\boldsymbol{x}\boldsymbol{\beta}) \end{aligned}$$

Thus,

$$\begin{aligned} \frac{\Omega_m(x_k + \delta, \boldsymbol{x})}{\Omega_m(x_k, \boldsymbol{x})} &= \frac{\exp(\tau_m) \exp(-\beta_k x_k) \exp(-\beta_k \delta) \exp(-\boldsymbol{x}\boldsymbol{\beta})}{\exp(\tau_m) \exp(-\beta_k x_k) \exp(-\boldsymbol{x}\boldsymbol{\beta})} \\ &= \exp(-\beta_k \delta) \end{aligned}$$

subscript m. This means that the factor change in the cumulative odds is the same, regardless of which category receives the focus. For example, if $\beta_k = .5$ and there are 4 categories in y, then a unit increase in x_k changes the cumulative odds by a factor of .61, regardless of whether one contrasts category 1 with 2, 3, and 4, or 1 and 2 with 3 and 4, or 1, 2, and 3 with 4. This is known as the proportional odds assumption. We shall revisit this assumption later in this chapter.

10.1.5 Model Fit

As always, an important aspect of modeling is assessing the model fit. How well does the model do in terms of accounting for the response variable? In the context of ordered logit and probit models, there are two useful approaches toward assessing model fit: correct predictions and pseudo-R^2.

Correct predictions

As we have seen, it is possible to compute predicted probabilities using (10.6). Based on these predicted probabilities, one can make a prediction concerning the response variable y. Specifically, one can label as the predicted response that value of y that has the largest associated predicted probability. Thus,

$$\hat{y}_i \quad = \quad m \quad \text{if } \hat{\pi}_{im} > \hat{\pi}_{ij} \; \forall j \neq m$$

We can then tabulate the predicted responses against the actual responses, using a table such as Table 10.1. The frequency of correct predictions is given by $\sum_{m=1}^{M} n_{mm}$ and the percentage of correct classifications (CC) is equal to

$$CC \quad = \quad \frac{\sum_{m=1}^{M} n_{mm}}{n} \times 100$$

When judging CC it is important to keep in mind how well one would have predicted in the absence of any covariate information. In this case, the best prediction of y would be the mode and we can simply compare CC to the percentage of cases that fall into the modal category.

Pseudo-R^2

A simulation study by Veall and Zimmermann (1993) demonstrates that two pseudo-R^2 measures work well in the context of ordered logit and probit analysis. The first is McKelvey and Zavoina's (1975) pseudo-R^2 measure, which

Table 10.1: Correct Prediction in Ordered Regression Models

	y				
$\hat{y}$	1	2	$\cdots$	M	Total
1	n_{11}	n_{12}	$\cdots$	n_{1M}	$n_{1.}$
2	n_{21}	n_{22}	$\cdots$	n_{2M}	$n_{2.}$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
M	n_{M1}	n_{M2}	$\cdots$	n_{MM}	$n_{M.}$
Total	$n_{.1}$	$n_{.2}$	$\cdots$	$n_{.M}$	n

is based on the latent variable y^* and is described in (9.19). The second is Aldrich and Nelson's (1984) LR-based pseudo-R^2, described in (9.20), with the Veall-Zimmermann correction. Note that the correction requires knowledge of ℓ_0, the log-likelihood of a model containing cut points only. This log-likelihood is given by

$$\ell_0 = \sum_{m=1}^{M} n_m \ln\left(\frac{n_m}{n}\right)$$

where n_m is the number of cases who selected response m.[5]

10.1.6 Example

Estimation Results

As an example of ordered logit and probit we consider the determinants of retrospective economic evaluations in 2004. Since 1980, the American National Election Studies have asked respondents to judge whether, during the preceding year, the economy had worsened (1), improved (3), or stayed the same (2). We are treating these responses as an ordinal scale that depends on a person's employment status (1 = unemployed; 0 = other), education, income, age, race (1 = black; 0 = white), gender (1 = female; 0 = male), and partisanship (as captured by dummies for Republicans and Democrats; independents are the baseline category). Table 10.2 shows the ordered logit

[5]In Stata, ℓ_0 can be obtained via the `fitstat` command.

Table 10.2: Ordinal Regression Models of Economic Perceptions in 2004

	Ordered Logit		Ordered Probit	
Predictor	**Estimate**	**S.E.**	**Estimate**	**S.E.**
Democrat	$-.580^{*}$	.250	$-.345^{*}$	.148
Republican	1.162^{**}	.255	$.695^{**}$	.151
Age	$-.001$	.004	$-.000$	.002
Education	$.154^{+}$	.085	$.087^{+}$	.051
Income	$.219^{**}$	.061	$.128^{**}$	.036
Black	$-.416^{*}$	.199	$-.264^{*}$	.117
Female	$-.369^{**}$	.134	$-.225^{**}$	.080
Unemployed	$-.393$	.391	$-.221$	.218
1st Cut Point	$.670^{+}$	.391	$.387^{+}$	.234
2nd Cut Point	2.345^{**}	.400	1.378^{**}	.237
ℓ	-833.799		-833.111	
LR χ^2	241.160		242.540	
p	.000		.000	
$R^2_{M\&Z}$	.254		.286	
$R_{V\&Z}$	.192		.312	

Notes: $n = 895$. $^{**}p < .01$, $^{*}p < .05$, $^{+}p < .10$ (two-tailed). Models estimated using Stata's `ologit` and `oprobit` commands.

and probit results from this analysis. These results were obtained by running the following Stata commands:

```
ologit retro3 dem rep age female black educ income unemployed
if year==2004

oprobit retro3 dem rep age female black educ income unemployed
if year==2004
```

Judging by the pseudo-R^2 values, the model has a mediocre fit to the data. Further evidence of this is obtained when we compute the correct classifications. Only about 55% of the cases are correctly classified in the ordered logit model and about 56% are correctly classified in the ordered probit model. These are disappointing numbers, considering that we would

have predicted almost 45% of the cases correctly had we just considered the modal category (worse) of the retrospective evaluations.[6]

Interpretation

Table 10.2 allows us to draw a number of conclusions. First, we observe statistically significant effects from partisanship, income, race, and gender. Education is marginally significant, while employment status is not at all significant. Second, the positive effects on Republican, education, and income mean that Republicans, the better educated, and individuals with higher family incomes tended to view the economy more positively in 2004. On the other hand, the negative effects for Democrat, black, and female suggest that Democrats, blacks, and women tended to view the economy more negatively. Beyond these basic conclusions, however, a thorough interpretation of the results requires that we use one of the interpretative tools discussed earlier. I shall illustrate these tools here for the ordered logit model, with a focus on the effects of income, race, and partisanship.

Marginal Effects and Elasticities Stata's `mfx` command produces marginal effects at the mean. For ordered logit and probit models, the syntax is given by

```
mfx, predict(p outcome(#)) [eyex]
```

Here `outcome(#)` is the outcome for which one seeks marginal effects. To obtain all possible marginal effects, one needs to issue as many commands

[6]To compute the predicted outcomes I used the following syntax after the `ologit` and `oprobit` commands:

```
predict p1 p2 p3 if e(sample)
gen predout=.
replace predout=1 if p1>p2 & p1>p3
replace predout=2 if p2>p1 & p2>p3
replace predout=3 if p3>p1 & p3>p2
```

Here, `p1` is the predicted probability of saying that the economy has gotten worse, `p2` is the predicted probability of saying that the economy has stayed the same, and `p3` is the predicted probability of saying that the economy has gotten better.

Table 10.3: Marginal Effect of Income on Economic Perceptions

Category	Effect	S.E.	z	p	Elasticity
Worse	−.054	.015	−3.610	.000	−.366
Same	.019	.006	3.250	.001	.147
Better	.035	.010	3.600	.000	.513

Notes: Marginal effects at the mean were computed using Stata's `mfx` command. Marginal effects are based on ordered logit results.

as there are outcomes, varying the outcome number in each run. Note that `mfx` computes the discrete change in the predicted probability for dummy predictors such as race. It will only compute marginal effects proper for continuous predictors. Adding the option `eyex` causes Stata to compute elasticities instead of marginal effects.

Table 10.3 displays the average marginal effects for income. We observe statistically significant average marginal effects for all of the response categories of the economic perception variable. The marginal effect is negative for the category "worse", which means that a very small increase in income reduces the likelihood of this response. The marginal effects for "same" and "better" are positive, implying an increased probability of those responses for a very small change in income. When transformed into elasticities, the effects are substantively sizable. A one percentage point increase in income is associated with a .37% decline in the probability of responding "worse", a .15% increase in the probability of responding "same", and a .51% increase in the probability of responding "better."

Predicted Probabilities and Discrete Change in Predicted Probabilities Predicted probabilities can be computed in a variety of ways. One option is the `prtab` command that we discussed in Chapter 9. For example,

```
prtab black
```

yields three tables that distinguish between the predicted probabilities for blacks and whites, one for each outcome. The results from this command are summarized in Table 10.4. This table clearly shows the tendency toward greater economic pessimism among black respondents.

Table 10.4: Race and Economic Perceptions

	Race	
Response	**White**	**Black**
Worse	.411	.514
Same	.377	.336
Better	.212	.151

Notes: Predicted probabilities were computed using the `prtab` command while holding all other predictors at their mean. Predicted probabilities are based on ordered logit estimation results.

Another way of obtaining predicted probabilities is the `prgen` command, which is particularly useful if the goal is to graph the predicted probabilities. The following command sequence produces the graph shown in Figure 10.2:

```
prgen income, from(1) to(5) gen(inc) rest(mean) ncases(5)

label variable incp1 "Worse"

label variable incp2 "Same"

label variable incp3 "Better"

graph twoway connected incp1 incp2 incp3 incx, xtitle("Income")
ytitle("Prob")
```

This graph clearly shows the declining probability of a "worse" response as income increases and the increasing probabilities of "same" and "better" responses.

A third approach toward computing predicted probabilities is the `prvalue` command. For ordered logit and probit models, the syntax is essentially the same as that for binary logit and probit models. For example, the following command sequence allows us to compare the predicted probabilities for Democrats and Republicans:

```
prvalue, x(dem=1 rep=0) rest(mean) delta save lev(99)
```

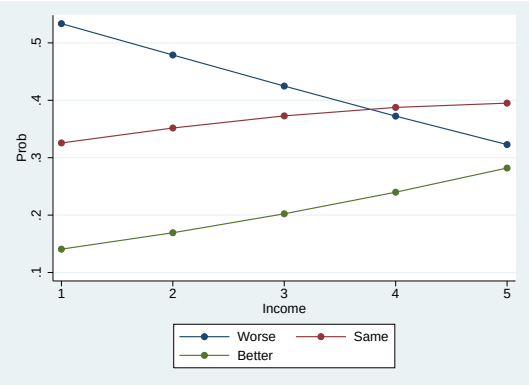

Figure 10.2: Economic Retrospection as a Function of Income

```
prvalue, x(dem=0 rep=1) rest(mean) delta dif lev(99)
```

The results are shown in Table 10.5, which reveals stark differences in the economic perceptions of Democrats and Republicans in 2004. Specifically, whereas the most likely response for Democrats was that the economy had worsened, for Republicans the most likely response was that it had improved.

Finally, discrete change measures can be obtained via the `prchange` command. For dummy predictors, this command shows the maximum discrete change in predicted probabilities. For continuous covariates, the command shows maximum change, centered unit change, and standard deviation change. In addition to reporting the changes in the predicted probabilities for each of the response categories, the `prchange` command also shows the average absolute change. Table 10.6 shows the maximum discrete change measures for partisanship, race, and income, while holding all other predictors constant at the mean. The table reveals powerful effects especially for Republican and income, revealing the by now familiar pattern that Republicans and wealthy respondents are more likely to see the economy in a positive light than everyone else.

The Cumulative Odds Ratio In Stata, the cumulative odds ratio can be computed via the `listcoef` command. Two options may be specified: `factor` reports the factor change in the cumulative odds, while `percent` computes the percentage change. Note that the `listcoef` command computes the odds ratio as the the ratio of $\Pr(y > m)$ over $\Pr(y \leq m)$ instead of $\Pr(y \leq m)$ over $\Pr(y > m)$. This means that the cumulative odds is

Table 10.5: Partisanship and Economic Perceptions

Response	**Democrats**	**Republicans**	Δ
Worse	.615	.219	$-.396^{**}$
Same	.280	.380	$.100^{**}$
Better	.105	.401	$.296^{**}$

Notes: $^{**}p < .01$ (two-tailed). Predicted probabilities were computed using the `prgen` command while holding all other predictors at their mean. Predicted probabilities are based on ordered logit estimation results.

Table 10.6: Discrete Change in Predicted Probabilities

Predictor	**Worse**	**Same**	Better	$\bar{\Delta}$
Democrat	.141	$-.048$	$-.093$	.094
Republican	$-.273$	.075	.198	.182
Black	.103	$-.042$	$-.061$	.069
Income	$-.211$	.069	.141	.140

Notes: Discrete change measures were computed using the `prchange` command while holding all other predictors at their mean. The table entries are based on ordered logit estimation results.

Table 10.7: Cumulative Odds Ratios

Predictor	Factor	Percent
Democrat	.560	−44.000
Republican	3.198	219.800
Black	.660	−34.000
Income	1.245	24.400

Notes: Cumulative odds ratios reflect the ratio of $\Pr(y > m)$ over $\Pr(y \leq m)$ and were computed using the `listcoef` command.

computed as $\exp(\beta_k \delta)$ instead of $\exp(-\beta_k \delta)$. Of course, one can always move from one definition of the cumulative odds to the next by inverting the result. This can also be done by requesting the `reverse` option with the `listcoef` command.

Table 10.7 shows the factor and percentage changes in the cumulative odds. For example, Democrats are 44% less likely to choose a more optimistic response category (e.g. "better") compared to a less optimistic response category (e.g. "same" or "worse"). For Republicans, the result is reversed: the odds of choosing a more optimistic response (e.g. "same" or "better") compared to a less optimistic response (e.g. "worse") are almost 220% greater for Republicans.

10.1.7 Heteroskedastic Ordered Logit and Probit

Derivation, Estimation, and Interpretation

The ordered logit and probit models have a built-in homoskedasticity assumption that is necessary to fix the scale of the parameters. But what happens when the homoskedasticity assumption is false? In this case, we incorrectly fix the scale of y^* and, by extension, $\boldsymbol{\beta}$ to a common metric. Instead, we should be fixing the scales of y^* and $\boldsymbol{\beta}$ differently for different sub-groups. By not doing so, the parameter estimates will be inconsistent, as will be the estimated standard errors.

It is possible to do away with the homoskedasticity assumption by specifying a so-called heteroskedastic ordered logit or probit model (see Alvarez

and Brehm 1995, 1998, 2002). Here, I shall show the derivation of the heteroskedastic ordered probit model, although the derivation of the heteroskedastic ordered logit model is analogous. Consider the following variance model for the disturbances

$$\sigma_i^2 = [\exp(\boldsymbol{z}_i\boldsymbol{\gamma})]^2$$

As we saw in Section 9.2.7, this is equivalent to the variance model that Harvey (1979) proposed. As before, $\boldsymbol{z}_i$ does not contain a constant. Hence, $\sigma_i^2 = 1$ when $\boldsymbol{\gamma} = \mathbf{0}$.

Given the variance model, we can create weighted disturbances

$$\begin{aligned}\delta_i &= \frac{\epsilon_i}{\sigma_i} \\ &= \frac{\epsilon_i}{\exp(\boldsymbol{z}_i\boldsymbol{\gamma})}\end{aligned}$$

that have unit variances. If we divide ϵ_i by $\exp(\boldsymbol{z}_i\boldsymbol{\gamma})$, then the other terms in the choice model should be divided in a similar way. Thus,

$$\Pr(y_i = m) = \Phi\left(\frac{\tau_m - \boldsymbol{x}_i\boldsymbol{\beta}}{\exp(\boldsymbol{z}_i\boldsymbol{\gamma})}\right) - \Phi\left(\frac{\tau_{m-1} - \boldsymbol{x}_i\boldsymbol{\beta}}{\exp(\boldsymbol{z}_i\boldsymbol{\gamma})}\right)$$

This is the core of the heteroskedastic ordered probit model.

The log-likelihood function for the model is given by

$$\ell = \sum_i \sum_m z_{im} \ln\left[\Phi\left(\frac{\tau_m - \boldsymbol{x}_i\boldsymbol{\beta}}{\exp(\boldsymbol{z}_i\boldsymbol{\gamma})}\right) - \Phi\left(\frac{\tau_{m-1} - \boldsymbol{x}_i\boldsymbol{\beta}}{\exp(\boldsymbol{z}_i\boldsymbol{\gamma})}\right)\right]$$

(see Alvarez and Brehm 1998). This is a complex log-likelihood function to optimize and convergence problems are not uncommon.

Interpretation of the heteroskedastic ordered probit model is also complex. Let w_k denote a covariate that is included in $\boldsymbol{x}$, $\boldsymbol{z}$, or in both. Then the marginal effect is defined as

$$\begin{aligned}\frac{\partial \pi_m}{\partial w_k} = {} & \phi\left(\frac{\tau_m - \boldsymbol{x}_i\boldsymbol{\beta}}{\exp(\boldsymbol{z}_i\boldsymbol{\gamma})}\right) \frac{\boldsymbol{x}\boldsymbol{\beta}\gamma_k - \beta_k - \tau_m\gamma_k}{\exp(\boldsymbol{z}_i\boldsymbol{\gamma})} - \\ & \phi\left(\frac{\tau_{m-1} - \boldsymbol{x}_i\boldsymbol{\beta}}{\exp(\boldsymbol{z}_i\boldsymbol{\gamma})}\right) \frac{\boldsymbol{x}\boldsymbol{\beta}\gamma_k - \beta_k - \tau_{m-1}\gamma_k}{\exp(\boldsymbol{z}_i\boldsymbol{\gamma})}\end{aligned}$$

This simplifies only when w_k occurs in just one set of the predictors. For example, if w_k is included in $\boldsymbol{x}$ but not $\boldsymbol{z}$, then all terms involving γ_k will

drop out. And if w_k appears in $\boldsymbol{z}$ but not in $\boldsymbol{x}$, then the terms involving β_k will drop out. However, if w_k occurs in both $\boldsymbol{x}$ and $\boldsymbol{z}$, then the effect can deviate dramatically from what is suggested by β_k or γ_k.

Example

Stata currently has no built-in capability to estimate the heteroskedastic ordinal regression models. However, Richard Williams has written a program, `oglm`, that, as one of its options, will estimate a heteroskedastic ordered logit or probit model. The syntax is as follows:

```
oglm depvar [ indepvars ], hetero(varlist) link(logit/probit)
```

Specifying `link(logit)` will cause Stata to estimate a heteroskedastic ordered logit model, whereas `link(probit)` results in estimation of a heteroskedastic ordered probit model.

Table 10.8 shows the results of a heteroskedastic ordered probit model where the variance model includes political knowledge as a predictor. This specification reflects the common hypothesis that greater knowledge produces more predictability of survey responses (Alvarez and Brehm 1995, 1998, 2002). The results reveal a marginally significant effect for knowledge in the expected negative direction.[7] Note that the inclusion of knowledge in the variance model causes the significance levels of the other variables to decline, although this may also be a consequence of the reduced sample size.

We can test whether a heteroskedastic specification is necessary by performing a Wald or LR test on the null hypothesis $H_0 : \boldsymbol{\gamma} = \mathbf{0}$. In our example, this test is redundant because there is only one predictor in the variance model and we can simply ascertain whether it is significant. Since knowledge is only significant at the .10 level, there is serious doubt about the necessity for a heteroskedastic model, at least in its current specification.

Some Cautionary Remarks

When performing a heteroskedastic ordinal regression analysis it should be kept in mind that one is tinkering with a fundamental identifying assumption. This can be quite risky. Monte Carlo simulations by Keele and Park (2005)

[7] Stata estimates $\ln(\sigma_i$ in lieu of σ_i. This is desirable because $\ln(\sigma_i$ is not bounded, whereas σ_i is bounded to be non-negative. Note that we can always move to σ_i by exponentiating the coefficients.

Table 10.8: Heteroskedastic Ordered Probit Model of Economic Perceptions

Predictor	Estimate	S.E.
Model for π		
Democrat	$-.218$	.167
Republican	$-.838^{**}$	.170
Age	.002	.003
Education	.084	.053
Income	$.152^{**}$	.038
Black	$-.295^{*}$	.128
Female	$-.203^{*}$	.084
Unemployed	$-.202$	.251
1st Cut Point	$.688^{**}$	.258
2nd Cut Point	1.648^{**}	.262
Variance Model		
Knowledge	$-.135^{+}$	.078
ℓ	-732.754	
Wald test/p	3.020	.082

Notes: $n = 348$. $^{**}p < .01$, $^{+}p < .10$ (two-tailed). Estimates obtained using the `oglm` command. The Wald test assesses $H_0 : \boldsymbol{\gamma} = \mathbf{0}$.

demonstrate that serious biases can creep into the estimates of $\boldsymbol{\beta}$ and, to a lesser extent, $\boldsymbol{\gamma}$, especially in the presence of model misspecification. The standard errors may also be off by a considerable amount if the variance model is misspecified. However, these problems appear to be less severe than they are for heteroskedastic binary logit and probit models, a result that may well have to do with the increase in information that is available from ordinal scales.

10.2 Alternatives to Ordered Logit and Probit

10.2.1 The Parallel Regression Assumption

What the Assumption Means

In addition to homoskedastic disturbances, ordered logit and probit assume that the effect of predictors is constant across the categories of the response variable. This is known as the **parallel regression assumption**. The assumption is particularly obvious when we consider the cumulative choice probabilities:

$$\Pr(y_i \leq m) \quad = \quad F(\tau_m - \boldsymbol{x}_i\boldsymbol{\beta})$$

Key here is the absence of a m-subscript on $\boldsymbol{\beta}$, which implies that the cumulative choice probabilities depend only on the change in cut points, not on a change in the effect of the predictors. When we graph the cumulative choice curves we obtain parallel lines, as is shown in Figure 10.3. This explains why the assumption is referred to as the parallel regression assumption.

In the ordered logit model, an alternative way of thinking about the parallel regression assumption is in terms of **proportional odds**. Consider the following cumulative log-odds ratio:

$$\begin{aligned} \ln\left[\frac{\Pr(y_i \leq m|\boldsymbol{x}_{i1})/\Pr(y_i > m|\boldsymbol{x}_{i1})}{\Pr(y_i \leq m|\boldsymbol{x}_{i2})/\Pr(y_i > m|\boldsymbol{x}_{i2})}\right] &= \ln\left[\frac{\exp(\tau_m - \boldsymbol{x}_{i1}\boldsymbol{\beta})}{\exp(\tau_m - \boldsymbol{x}_{i2}\boldsymbol{\beta})}\right] \\ &= (\boldsymbol{x}_{i1} - \boldsymbol{x}_{i2})\boldsymbol{\beta} \end{aligned}$$

where $\boldsymbol{x}_{i1}$ and $\boldsymbol{x}_{i2}$ are two different sets of values of the covariates. We see that the log of the cumulative odds ratio is proportional to the distance between

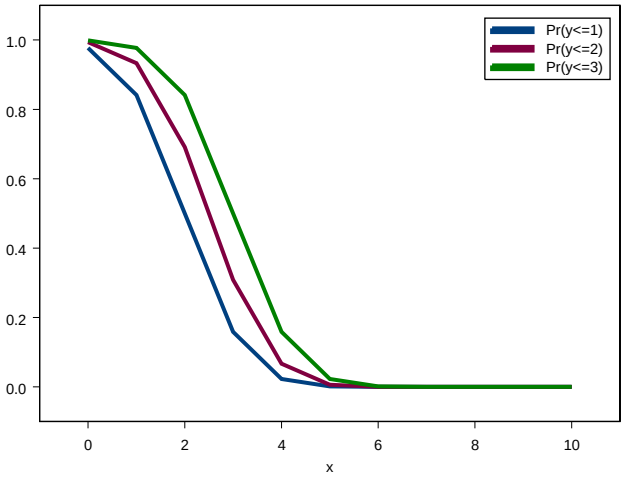

Figure 10.3: Parallel Regression Assumption

the values of the predictors. Critically, it does not depend on changing effects of those predictors.

The parallel regression assumption is a rather stringent assumption that is frequently violated (Long and Freese 2003). When it is, the parameter estimates that come out of an ordered logit or probit analysis may well be inconsistent. It is easy to see why. Imagine that a predictor has a positive effect for some categories of y and a negative effect for other categories. When we constrain the effect to be constant across all of the categories of y, as we have to do under the parallel regression assumption, then the net effect might well come out to zero. This would give the false impression that the predictor is inconsequential. Given the potential for inconsistent parameter estimates, it is essential that one checks the validity of the parallel regression assumption. Fortunately, there are several procedures that will allow you to do this.

Testing for Parallelism

An Approximate Likelihood Ratio Test Wolfe and Gould (1998) have proposed an approximate LR test of the parallel regression assumption. The test is based on the cumulative probability curves described in Figure 10.3.

If parallelism holds, then we can think of the ordinal logit and probit models in terms of $M-1$ binary regressions[8] of the variety

$$\Pr(y_i \leq m) \quad = \quad F(\tau_m - \boldsymbol{x}_i\boldsymbol{\beta})$$

Central here is the constraint that the effect of the predictors is the same across the $M-1$ regressions. We call this model 1, which requires estimation only of the p elements contained in $\boldsymbol{\beta}$ and τ. If parallelism does not hold, then we should be estimating $M-1$ equations of the form

$$\Pr(y_i \leq m) \quad = \quad F(\tau_m - \boldsymbol{x}_i\boldsymbol{\beta}_m)$$

This is model 2 and it requires estimation of $p(M-1)$ parameters.

An exact LR test would recognize that model 1 is nested into model 2—model 1 is a special case of model 2 that comes about by constraining the elements of $\boldsymbol{\beta}_m$ to be identical across the categories of y. The exact LR test statistic is then defined by

$$-2\left[\ell_1 - \ell_2\right] \quad \sim \quad \chi^2_{p(M-2)} \tag{10.11}$$

where the degrees of freedom are given by the difference in the numbers of estimated parameters between the two models.

To compute the exact LR test statistic, we need to take into consideration the correlations that exist between the binary responses. For example, $\Pr(y \leq 2)$ and $\Pr(y \leq 1)$ contain a common response options (1) and are therefore not independent. The correlation between successive responses is given by

$$\rho_{y_{ij},y_{ik}} \quad = \quad \sqrt{\frac{\pi_{ij}(1-\pi_{ij})}{\pi_{ik}(1-\pi_{ik})}}$$

for $j < k$. It is difficult to estimate the second model while considering these correlations. Thus, in practice, we treat the $M-1$ regressions as independent, as would be the case if one were to run a multinomial logit analysis (see Chapter 11). Let ℓ_2^{ib} denote the log-likelihood of model 2 conceived of as independent binaries. In general, $\ell_2^{ib} < \ell_2$ but Wolfe and Gould (1998)

[8]There are only $M-1$ regressions because the cumulative probability of the last response category is fixed at 1.

maintain that the difference is small. Thus, (10.11) can be approximated via $-2\left[\ell_1 - \ell_2^{ib}\right]$.

Wolfe and Gould (1998) provide the Stata program `omodel` to perform the approximate LR test. The syntax is as follows:

```
omodel logit/probit depvar indepvars
```

The `omodel logit` command performs a ordered logit analysis as well as a test of the parallel regression assumption; `omodel probit` also tests the parallel regression assumption and additionally estimates an ordered probit model.

A Lagrange Multiplier Test* The idea of binary regressions can also be used to develop an LM test. This test is implemented in SAS but not Stata. Consider a set of $M-1$ binary choice models $\Pr(y \leq m) = F(\tau_m - \mathbf{x}\boldsymbol{\beta}_m)$. We impose the constraint $\boldsymbol{\beta}_1 = \boldsymbol{\beta}_2 = \cdots \boldsymbol{\beta}_{M-1} = \boldsymbol{\beta}$. This is the parallel regression assumption. Thus, the constrained model that we estimate is $\Pr(y \leq m) = F(\tau_m - \mathbf{x}\boldsymbol{\beta})$. Now form the LM test statistic

$$LM \quad = \quad \boldsymbol{\nabla}_{\boldsymbol{\beta}}' \mathbf{V}[\boldsymbol{\beta}] \boldsymbol{\nabla}_{\boldsymbol{\beta}} \overset{asy}{\sim} \chi^2_{p(M-2)}$$

where $\boldsymbol{\nabla}_{\boldsymbol{\beta}}$ and $\mathbf{V}[\boldsymbol{\beta}]$ are based on the constrained parameters.

The Brant Test The LR and LM tests are omnibus tests. As such, they are very good at showing *if* there is a problem with the parallel regression assumption. However, they are not good at showing *where* the problem is located if there is one. Here a Wald test procedure developed by Brant (1990) is more useful, although it can only be used with ordered logit.

As a Wald test procedure, the Brant test is based on the unrestricted model, i.e. $M-1$ binary regressions. The test starts by estimating $\boldsymbol{\beta}_m$ and its variance. Let

$$z_m \quad = \quad \begin{cases} 1 & y > m \\ 0 & y \leq m \end{cases}$$

Note that these binaries are reversed from what we used in the LR and LM tests, which focused on $\Pr(y_i \leq m$ rather than $\Pr(y_i > m)$. Then we

estimate $M-1$ separate binary regressions yielding $\hat{\boldsymbol{\beta}}_m$ and $\mathbf{V}[\hat{\boldsymbol{\beta}}_m]$. We can also compute predicted probabilities[9]

$$\hat{\pi}_m \;=\; F(-\hat{\tau}_m + \mathbf{x}\hat{\boldsymbol{\beta}}_m)$$

Next, we obtain the covariance between $\hat{\boldsymbol{\beta}}_m$ and $\hat{\boldsymbol{\beta}}_l$. This covariance is given by

$$\hat{\mathbf{C}}[\hat{\boldsymbol{\beta}}_m, \hat{\boldsymbol{\beta}}_l] \;=\; (\mathbf{X}'_+\mathbf{W}_{mm}\mathbf{X}_+)^{-1}\mathbf{X}'_+\mathbf{W}_{ml}\mathbf{X}_+(\mathbf{X}'_+\mathbf{W}_{ll}\mathbf{X}_+)^{-1}$$

where $\mathbf{X}_+$ is a $n \times (p+1)$ matrix that is formed by augmenting the matrix of predictors, $\mathbf{X}$, with a leading column of 1s. Further, $\mathbf{W}_{ml}$ is a $n \times n$ diagonal matrix with entries $\hat{\pi}_m - \hat{\pi}_m\hat{\pi}_l$ for $m \leq l$.

We now combine the estimates. Let $\hat{\boldsymbol{\beta}}^* = (\hat{\boldsymbol{\beta}}'_1\ \hat{\boldsymbol{\beta}}'_2 \cdots \hat{\boldsymbol{\beta}}'_{M-1})$. It can be shown that $\hat{\boldsymbol{\beta}}^* \overset{asy}{\sim} MVN$ (multivariate normal) with $\mathcal{E}[\hat{\boldsymbol{\beta}}_m] \approx \boldsymbol{\beta}_m$ and

$$\hat{\mathbf{V}}[\hat{\boldsymbol{\beta}}^*] \;=\; \begin{bmatrix} \hat{\mathbf{V}}[\hat{\boldsymbol{\beta}}_1] & \hat{\mathbf{C}}[\hat{\boldsymbol{\beta}}_1, \hat{\boldsymbol{\beta}}_2] & \cdots & \hat{\mathbf{C}}[\hat{\boldsymbol{\beta}}_1, \hat{\boldsymbol{\beta}}_{M-1}] \\ \hat{\mathbf{C}}[\hat{\boldsymbol{\beta}}_2, \hat{\boldsymbol{\beta}}_1] & \hat{\mathbf{V}}[\hat{\boldsymbol{\beta}}_2] & \cdots & \hat{\mathbf{C}}[\hat{\boldsymbol{\beta}}_2, \hat{\boldsymbol{\beta}}_{M-1}] \\ \vdots & \vdots & \ddots & \vdots \\ \hat{\mathbf{C}}[\hat{\boldsymbol{\beta}}_{M-1}, \hat{\boldsymbol{\beta}}_1] & \hat{\mathbf{C}}[\hat{\boldsymbol{\beta}}_{M-1}, \hat{\boldsymbol{\beta}}_2] & \cdots & \hat{\mathbf{V}}[\hat{\boldsymbol{\beta}}_{M-1}] \end{bmatrix}$$

The penultimate step is to create the pairwise contrasts with $\hat{\boldsymbol{\beta}}_1$. The parallel regression assumption implies

$$\mathbf{D}\boldsymbol{\beta}^* \;=\; \mathbf{0}$$

where

$$\mathbf{D} \;=\; \begin{bmatrix} \mathbf{I} & -\mathbf{I} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{I} & \mathbf{0} & -\mathbf{I} & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{I} & \mathbf{0} & \mathbf{0} & \cdots & -\mathbf{I} \end{bmatrix}$$

is a $p(J-2) \times p(J-1)$ contrast matrix. This matrix always selects $\boldsymbol{\beta}_1$ and subtracts $\boldsymbol{\beta}_2 \cdots \boldsymbol{\beta}_{M-1}$, setting the results equal to zero, as is implied by the

[9] In an ordered logit model, $\Pr(y > m) = \Pr(\epsilon > \tau_m - \boldsymbol{x}\boldsymbol{\beta}_m) = \Pr(\epsilon < -[\tau_m - \boldsymbol{x}\boldsymbol{\beta}_m]) = \Pr(\epsilon < -\tau_m + \boldsymbol{x}\boldsymbol{\beta}_m)$.

parallel regression assumption. Thus, the pairwise contrasts are the essence of the Brant test.[10]

The Brant test statistic is now given by

$$W = (\mathbf{D}\hat{\boldsymbol{\beta}}^*)'[\mathbf{D}\hat{\mathbf{V}}(\hat{\boldsymbol{\beta}}^*)\mathbf{D}']^{-1}(\mathbf{D}\hat{\boldsymbol{\beta}}^*) \overset{asy}{\sim} \chi^2_{p(M-2)} \tag{10.12}$$

By selecting the proper elements of $\boldsymbol{\beta}_m$ and the proper elements of the variance-covariance matrix we can also identify covariates for which the parallel regression assumption fails. The procedure is implemented in Stata through the `brant` command, whose syntax is

```
brant [, detail]
```

(where detail causes Stata to show the estimates from the binary regressions).

Example

Does the parallel regression assumption hold for the ordered logit analysis of economic perceptions that is reported in Table 10.2? When we run

```
omodel logit retro3 dem rep age female black educ income
unemployed if year==2004
```

we obtain an approximate LR test statistic of 12.70. When referred to a χ^2-distribution with 8 degrees of freedom we obtain $p = .123$. This suggests that parallelism seems reasonable for the economic perception model.

The Brant test leads to the same overall conclusion: $W = 11.87$, $p = .157$. However, when we peruse the results for specific predictors, two problem cases emerge: the test statistics for gender and race are both significant at the .05 level, as is shown in Table 10.9. This would be reason to consider estimating one of the alternative models that we shall discuss next.

10.2.2 The Generalized Ordered Logit Model

Derivation, Estimation, and Interpretation

The generalized ordered logit model is one of the models that bypasses the parallel regression assumption. The model was first proposed by McCullagh and Nelder (1989) and subsequently elaborated by Clogg and Shihadeh

[10]It is arbitrary that we form contrasts with $\boldsymbol{\beta}_1$. We could have formed contrasts with any of the coefficient vectors.

Table 10.9: Brant Test of Economic Perceptions Model

Predictor	W	p
Democrat	.40	.528
Republican	.12	.728
Age	1.38	.239
Education	1.74	.188
Income	.66	.416
Black	4.10	.043
Female	3.98	.046
Unemployed	.02	.897
Overall	11.87	.157

Notes: Results obtained via the `brant` command.

(1994), Fahrmeir and Trutz (1994), Fu (1998), Peterson and Harrell (1990), and Williams (2006). It has been implemented in Stata through the `gologit` (Fu 1998) and `gologit2` (Williams 2006) commands.

The basic model considers the probability of a response greater than m and is given by

$$\Pr(y_i > m) = \frac{\exp(\alpha_m + \mathbf{x}_i\boldsymbol{\beta}_m)}{1 + \exp(\alpha_m + \mathbf{x}_i\boldsymbol{\beta}_m)} \tag{10.13}$$

for $m = 1, 2, \cdots M-1$. The m-subscript on $\boldsymbol{\beta}$ implies that the effects of the predictors can vary across the categories of y. The model in (10.13) contains several sub-models:

1. If $M = 2$, then the model reduces to a binary logit model.
2. If $M > 2$, then the model is similar to the Brant (1990) specification.
3. If $\boldsymbol{\beta}_m = \boldsymbol{\beta}\ \forall m$ and all predictors, then the model reduces to the ordered logit model. (Note that the ordered logit cut points are equal to $-\alpha_m$.)

   If $\boldsymbol{\beta}_m = \boldsymbol{\beta}\ \forall m$ and a subset of the predictors, then we obtain the **partial proportional odds** model.

The partial proportional odds model is often a useful compromise between the parsimony of the standard ordered logit model and the potentially greater

veracity of the Brant (1990) specification. It allows researchers to retain the parallel regression assumption where this seems warranted and to drop it where it seems inconsistent with the data.

The log-likelihood function for the generalized ordered logit model is given by

$$\ell \;=\; \sum_m \sum_i z_{im} \ln \left[F(\alpha_{m-1} + \mathbf{x}_i \boldsymbol{\beta}_{m-1}) - F(\alpha_m + \mathbf{x}_i \boldsymbol{\beta}_m) \right]$$

where $F(.)$ is the standard logistic distribution, $\alpha_0 = \infty$, and $\alpha_M = -\infty$. This is a fairly straightforward log-likelihood function and convergence problems are rare.

Interpretation is done most straightforwardly in terms of predicted probabilities. For $y_i = 1$, the predicted probability is given by

$$\Pr(y_i = 1) \;=\; 1 - \frac{\exp(\alpha_1 + \boldsymbol{x}_i \boldsymbol{\beta}_1)}{1 + \exp(\alpha_1 + \boldsymbol{x}_i \boldsymbol{\beta}_1)}$$

For $y_i = 2 \cdots M - 1$, the predicted probabilities are given by

$$\Pr(y_i = m) \;=\; \frac{\exp(\alpha_{m-1} + \boldsymbol{x}_i \boldsymbol{\beta}_{m-1})}{1 + \exp(\alpha_{m-1} + \boldsymbol{x}_i \boldsymbol{\beta}_{m-1})} - \frac{\exp(\alpha_m + \boldsymbol{x}_i \boldsymbol{\beta}_m)}{1 + \exp(\alpha_m + \boldsymbol{x}_i \boldsymbol{\beta}_m)}$$

Finally, for $y_i = M$ the predicted probability is

$$\Pr(y_i = M) \;=\; \frac{\exp(\alpha_{M-1} + \boldsymbol{x}_i \boldsymbol{\beta}_{M-1})}{1 + \exp(\alpha_{M-1} + \boldsymbol{x}_i \boldsymbol{\beta}_{M-1})}$$

These predicted probabilities can be parlayed into discrete change measures in the usual manner.

Example

Fu (1998) and Williams (2006) have developed Stata add-on programs that will estimate the generalized ordered logit model. Of these programs, the `gologit2` routine of Williams is the most flexible as it allows users to specify partial proportional odds models. The syntax for this program is as follows:

`gologit2` *`depvar`* [*`indepvars`*] [`,`
`pl|pl(`*`varlist`*`)|npl|npl(`*`varlist`*`)|autofit`]

Table 10.10: Generalized Ordered Logit Model of Economic Perceptions

	> Worse		> Same	
Predictor	**Estimate**	**S.E.**	**Estimate**	**S.E.**
Democrat	$-.581^{*}$	.249	$-.581^{*}$	.249
Republican	1.135^{**}	.254	1.135^{**}	.254
Age	$-.001$	.004	$-.001$	.004
Education	$.153^{+}$	.085	$.153^{+}$	.085
Income	$.219^{**}$	.060	$.219^{**}$	.060
Black	$-.328$	.203	-1.031^{**}	.387
Female	$-.227$	.150	$-.582^{**}$	.172
Unemployed	$-.377$	.390	$-.377$	.390
Constant	$-.756^{+}$	.392	-2.186^{**}	.402

Notes: $n = 895$. $^{**}p < .01$, $^{*}p < .05$, $^{+}p < .10$ (two-tailed). Models estimated using Stata's `gologit2` command: `gologit2 retro3 dem rep age female black educ income unemployed if year==2004, npl(black female)`.

Here the `pl` option causes the program to assume that the parallel regression assumption holds for all predictors ("pl" stands for parallel lines). The option `pl(`*`varlist`*`)` applies the parallel regression assumption only to those predictors included in *varlist*. The option `npl` lifts the parallel regression assumption for all of the predictors, whereas `npl(`*`varlist`*`)` lifts it for the predictors included in *varlist* ("npl" stands for non-parallel lines). Finally, `autofit` causes the program to find a model that relaxes the parallel regression assumption where this is necessary in terms of model fit (for details see Williams 2006).

Table 10.10 shows the results from a generalized ordered logit analysis where I relaxed the parallel regression assumption for gender and race. As the table shows, gender and race do little to discriminate "same" and "better" responses from "worse" responses. However, they have sizable effects on discriminating "better" responses from "same" and "worse".

When considering the predicted probabilities, we see that women are .05 points more likely to believe that the economy had worsened and .06 points less likely to think that it had improved. Blacks are .07 points more likely

to believe that the economy had worsened and .06 points less likely to think that it had improved. These comparisons hold all other predictors at the mean (age, education, and income) or mode (Democrat, Republican, female, black, and unemployed).

10.2.3 The Stereotype Regression Model

Derivation, Estimation, and Interpretation

The stereotype regression model provides an alternative framework for avoiding the parallel regression assumption. This model was first proposed by Anderson (1984) and subsequently developed by DiPrete (1990) and Lunt (2001). It is implemented through Stata's `mclest` and `slogit` commands.

The stereotype regression model estimates a common coefficient vector $\boldsymbol{\beta}$ but in addition estimates a set of parameters that act as multipliers on the coefficients. These parameters are specific to the values of y. The model is given by

$$\Pr(y_i = m) = \frac{\exp(\theta_m - \phi_m \boldsymbol{x}_i\boldsymbol{\beta})}{1 + \sum_{j=1}^{M-1} \exp(\theta_j - \phi_j \boldsymbol{x}_i\boldsymbol{\beta})}$$

for $y_i < M$ and

$$\begin{aligned}\Pr(y_i = M) &= 1 - \frac{\sum_{j=1}^{M-1} \exp(\theta_j - \phi_j \boldsymbol{x}_i\boldsymbol{\beta})}{1 + \sum_{j=1}^{M-1} \exp(\theta_j - \phi_j \boldsymbol{x}_i\boldsymbol{\beta})} \\ &= \frac{1}{1 + \sum_{j=1}^{M-1} \exp(\theta_j - \phi_j \boldsymbol{x}_i\boldsymbol{\beta})}\end{aligned}$$

This can be written more elegantly in log-odds form:

$$\ln\left(\frac{\Pr(y_i = m)}{\Pr(y_i = l)}\right) = (\theta_m - \theta_l) - (\phi_m - \phi_l)\boldsymbol{x}_i\boldsymbol{\beta} \tag{10.14}$$

This formulation clearly shows that the effect of a predictor is allowed to vary by a scalar factor, $\phi_m - \phi_l$, that depends on the pair of alternatives under consideration. (The term $\theta_m - \theta_l$ is a function of the different cut points for different values of y.)

The model in (10.14) is identified only after we impose some constraints. Specifically, we impose three identifying constraints: (1) $\theta_M = 0$, (2) $\phi_1 = 1$,

and (3) $\phi_M = 0$. We also impose the ordinality constraint $\phi_1 = 1 > \phi_2 > \cdots > \phi_{J-1} > \phi_M = 0$; this constraint ensures that the ordinal nature of y is preserved in the estimation.

With these constraints in place, estimation is relatively straightforward. The log-likelihood function is given by

$$\ell = \sum_{i=1}^{n} \sum_{m=1}^{M} z_{im} \ln \pi_{im}$$

where $z_{im} = 1$ if alternative m was chosen and $\pi_{im} = \Pr(y_i = m)$. In my experience, convergence problems are rare, although estimation tends to be slower than for the ordered logit model. Interpretation is best done in terms of predicted probabilities, using the formulas presented above.

Example

Stata will estimate the stereotype regression model using the `mclest` command (Hendrickx 2000) or the `slogit` command. Here we shall focus on the latter command, whose syntax is

```
slogit depvar indepvars
```

Note that Stata's implementation of the stereotype regression model does not impose an ordinality constraint; if necessary, the user will have to specify this.[11]

Table 10.11 shows the results from the stereotype regression model of economic perceptions. The estimates satisfy the ordinality constraint without further intervention. Considering the log-odds of the "same" and "worse" responses, we see that the effect of the predictors is changed by a factor $\phi_2 - \phi_1 = .550 - 1 = -.450$. Considering the log-odds for "better" and "same" responses, the effect of the predictors is changed by a slightly larger factor of $\phi_3 - \phi_2 = 0 - .550 = -.550$.

Table 10.12 shows the maximum discrete changes in predicted probabilities, while holding all other covariates at the mean. This table shows particularly powerful effects for partisanship, income, and race. Republicans and wealthier respondents tended to view the economy in a more positive light in 2004, whereas Democrats and blacks tended to view it more negatively.

[11] The reason for the lack of the ordinality constraint is that Stata's command handles more than the ordinal case alone. For details see the Stata documentation.

Table 10.11: Stereotype Regression Model of Economic Perceptions

Predictor	**Estimate**	**S.E.**
Democrat	$-.874^{*}$	.369
Republican	1.544^{**}	.366
Age	$-.001$	.006
Education	$.228^{+}$	.124
Income	$.305^{**}$	.089
Black	$-.837^{*}$	.337
Female	$-.570^{**}$	.198
Unemployed	$-.663$	.594
ϕ_1	1.000	
ϕ_2	$.550^{**}$	.056
ϕ_3	.000	
θ_1	2.068^{**}	.574
θ_2	1.275^{**}	.339
θ_3	.000	

Notes: $n = 895$. $^{**}p < .01$, $^{*}p < .05$, $^{+}p < .10$ (two-tailed). Models estimated using Stata's `slogit` command.

Table 10.12: Discrete Change in the Stereotype Regression Model

Predictor	**Worse**	**Same**	Better	Δ
Democrat	.139	$-.029$	$-.110$	.092
Republican	$-.242$	.039	.203	.162
Age	.011	$-.002$	$-.009$	.008
Education	$-.108$	.026	.082	.072
Income	$-.192$	.039	.154	.128
Female	.091	$-.019$	$-.072$	.061
Black	.130	$-.038$	$-.092$	.087
Unemployed	.103	$-.031$	$-.072$	.069

Notes: Maximum discrete change measures are reported. All other predictors are held constant at their means.

Chapter 11

Probabilistic Choice Models

Probabilistic choice models (PCMs) are well-suited for formulating and testing theories of political choice. These models have several key characteristics. First, decision makers are assumed to be utility maximizers. This is a strict but common assumption in political science, although alternatives such as satisficing exist (e.g. Simon 1985). Second, utility contains a random component so that utility maximization—and ultimately choice—is probabilistic. The stochastic component captures several elements, including (1) unobserved attributes of the alternative, (2) unobserved attributes of the decision maker, (3) measurement error, and (4) proxies or instrumental variables (Manski 1997). Third, utility contains two systematic components in the form of characteristics of the decision maker and attributes of the alternatives (i.e. the options among which the decision maker has to choose).

The historical development of PCMs dates back to the 1920s, starting with the ideas of Thurstone. Major work on these models occurred in the 1950s-1970s, although political science's usage of PCMs is of a much more recent date. Conceptually, i.e. in terms of functional form, PCMs fall into three major categories:

1. *Lucean Forms:* These models are based on the ideas that Duncan Luce developed in the 1950s and 1960s concerning individual choice behavior. Models in this category assume "independence from irrelevant alternatives" (IIA). The multinomial logit model falls into this category.

2. *Thurstonian Forms:* These models are based on Leon Thurstone's ideas from the 1920s. They do not impose an IIA assumption. The multinomial probit model falls into this category.

3. *Tverskian Forms:* These models are based on Amos Tversky's theories of elimination-by-aspects and elimination-by-tree. They assume a sequential choice process. McFadden's nested logit model also falls into this category.

We will discuss examples from each of these categories. Before doing so, however, let us introduce some common notation.

11.1 The Nature of Probabilistic Choice

Let y_m denote a choice variable defined over alternatives m.[1] Further, let $m = 1, 2, \cdots, M$, which means that there are M alternatives in total. Underlying each alternative is a latent utility, y_m^*. The utility maximization postulate implies that

$$y_m = \begin{cases} 1 & \text{if } y_m^* = \max(y_1^*, y_2^*, \cdots y_M^*) \\ 0 & \text{otherwise} \end{cases}$$

In other words, if the utility of alternative m is greater than the utility of all other alternatives, then m will be chosen; otherwise it will not.

So far, we have been silent about the nature of the utilities. Stochastic utility maximization implies that these utilities have a random component. Specifically,

$$y_m^* = v_m + \epsilon_m$$

where v_m is a systematic component, which can be modeled out in terms of individual characteristics and attributes of the alternatives, while ϵ_m is a random component for which we need to make distributional assumptions. Thus, the condition that m is chosen if y_m^* is maximal means

$$\begin{aligned} y_m^* > y_k^* &\Leftrightarrow \\ v_m + \epsilon_m > v_k + \epsilon_k &\Leftrightarrow \\ \epsilon_k < \epsilon_m + v_m - v_k & \end{aligned}$$

$\forall\ m \neq k$ (i.e. for all other alternatives, not including m). Depending on the distributional assumptions on ϵ, this basic structure produces different PCMs.

[1]For the sake of simplicity, I will omit subscripts for the individual decision maker. However, such subscripts are assumed throughout this discussion. When it is useful we shall explicitly introduce them, as when we discuss the multinomial logit model.

11.2 Multinomial and Conditional Logit

11.2.1 Derivation

Choice Probabilities

The multinomial and conditional logit models can be derived from the standard Type-I maximum extreme value distribution that we saw earlier in the discussion of the gompit model (see Chapter 9.3.1 and Chapter 16). As applied to ϵ, the density and distribution functions of this distribution are:

$$\begin{aligned} f(\epsilon) &= \exp(-\epsilon)\exp[-\exp(-\epsilon)] \\ &= \exp[-\epsilon - \exp(-\epsilon)] \\ F(\epsilon) &= \exp[-\exp(-\epsilon)] \end{aligned}$$

In addition, we assume that $calE[\epsilon_m, \epsilon_k] = 0$ for $m \neq k$, i.e. the errors are independent. It is this independence that embodies the IIA assumption.

With this distributional assumption in place, it is now possible to derive the choice probabilities. This gets quite mathematical, and you could easily skip to the end of this section to see the final result if the math becomes too eye-blinding. To make the derivation more concrete, we start by considering a choice problem involving three alternatives (e.g. choosing among three different political parties in an election). We focus on the probability that the first alternative will be chosen. Thus, we want to know $\Pr(y = 1)$. Using the random utility maximization postulate and the distributional assumptions we have

$$\begin{aligned} \Pr(y = 1) &= \Pr(\epsilon_2 < \epsilon_1 + v_1 - v_2 \cap \epsilon_3 < \epsilon_1 + v_1 - v_3) \\ &= \Pr(\epsilon_2 < \epsilon_1 + v_1 - v_2)\Pr(\epsilon_3 < \epsilon_1 + v_1 - v_3) \end{aligned}$$

where the second equation follows from the assumed independence of the errors.

Evaluating this equation is a bit complicated because the right-hand side of the inequalities is not fixed (as it was in binary and ordered response models). On the contrary, it is stochastic since ϵ_1 is random. This means that we actually will have to integrate over ϵ_1 as well as ϵ_2 or ϵ_3. For example,

$$\Pr(\epsilon_2 < \epsilon_1 + v_1 - v_2) = \int_{-\infty}^{\infty} \left(\int_{-\infty}^{\epsilon_1 + v_1 - v_2} f(\epsilon_2) d\epsilon_2 \right) f(\epsilon_1) d\epsilon_1$$

where the part in parentheses captures the notion that $\epsilon_2 < \epsilon_1 + v_1 - v_2$ and the remainder of the expression addresses the stochastic nature of ϵ_1. We can simplify this expression considerably

$$\int_{-\infty}^{\infty} \left(\int_{-\infty}^{\epsilon_1 + v_1 - v_2} f(\epsilon_2) d\epsilon_2 \right) f(\epsilon_1) d\epsilon_1 \quad = \quad \int_{-\infty}^{\infty} F(\epsilon_1 + v_1 - v_2) f(\epsilon_1) d\epsilon_1$$

In terms of the Type-I extreme value distribution this becomes

$$\begin{aligned} F(\epsilon_1 + v_1 - v_2) f(\epsilon_1) &= \exp(-e^{-\epsilon_1 - v_1 + v_2}) \exp(-\epsilon_1 - e^{-\epsilon_1}) \\ &= \exp(-e^{-\epsilon_1 - v_1 + v_2} - e^{-\epsilon_1} - \epsilon_1) \\ &= \exp\left[-\epsilon_1 - e^{-\epsilon_1}\left(1 + e^{v_2 - v_1}\right)\right] \\ &= \exp\left[-\epsilon_1 - e^{-\epsilon_1}\left(1 + \frac{e^{v_2}}{e^{v_1}}\right)\right] \end{aligned}$$

so that

$$\Pr(\epsilon_2 < \epsilon_1 + v_1 - v_2) \quad = \quad \int_{-\infty}^{\infty} \exp\left[-\epsilon_1 - e^{-\epsilon_1}\left(1 + \frac{e^{v_2}}{e^{v_1}}\right)\right] d\epsilon_1$$

Similarly,

$$\Pr(\epsilon_3 < \epsilon_1 + v_1 - v_2) \quad = \quad \int_{-\infty}^{\infty} \exp\left[-\epsilon_1 - e^{-\epsilon_1}\left(1 + \frac{e^{v_3}}{e^{v_1}}\right)\right] d\epsilon_1$$

Hence,

$$\begin{aligned} \Pr(Y = 1) &= \Pr(\epsilon_2 < \epsilon_1 + v_1 - v_2) \Pr(\epsilon_3 < \epsilon_1 + v_1 - v_3) \\ &= \int_{-\infty}^{\infty} \exp\left[-\epsilon_1 - e^{-\epsilon_1}\left(1 + \frac{e^{v_2}}{e^{v_1}}\right)\right] \times \\ &\quad \exp\left[-\epsilon_1 - e^{-\epsilon_1}\left(1 + \frac{e^{v_3}}{e^{v_1}}\right)\right] d\epsilon_1 \\ &= \int_{-\infty}^{\infty} \exp\left[-\epsilon_1 - e^{-\epsilon_1}\left(1 + \sum_{k \neq 1} \frac{e^{v_k}}{e^{v_1}}\right)\right] d\epsilon_1 \end{aligned}$$

This looks quite ugly but fortunately considerable simplification is possible. Let $\lambda_1 = \ln[1 + (\sum_{k \neq 1} \exp(v_k)/\exp(v_1))] = \ln[\sum_k \exp(v_k)/\exp(v_1)]$,[2]

[2]To get the second form for λ we write $1 + \sum_{k \neq 1} \exp(v_k)/\exp(v_1)$ as $(\exp(v_1) + \sum_{k \neq 1} \exp(v_k))/\exp(v_1)$. We now realize that the numerator sums over all alternatives, so that it immediately follows that we have $\sum_k \exp(v_k)/\exp(v_1)$. All that is left to do then is to take the natural logarithm in order to obtain λ.

then

$$\begin{aligned}\int_{-\infty}^{\infty} \exp\left[-\epsilon_1 - e^{-\epsilon_1}\left(1 + \sum_{k\neq 1}\frac{e^{v_k}}{e^{v_1}}\right)\right] d\epsilon_1 &= \\ \int_{-\infty}^{\infty} \exp\left\{-\epsilon_1 - \exp[-(\epsilon_1 - \lambda_1)]\right\} d\epsilon_1 &= \\ \left.\frac{1}{\exp\left[\frac{1}{\exp(\epsilon_1)}\exp(\lambda_1)\right]\exp(\lambda_1)}\right|_{-\infty}^{\infty}&\end{aligned}$$

For $\epsilon_1 \to \infty$, $\exp(\epsilon_1) \to \infty$, and $1/\exp(\epsilon_1) \to 0$. Hence, the denominator of the expression above may be written as $\exp(0 \times \exp(\lambda_1))\exp(\lambda_1) = \exp(0)\exp(\lambda_1) = \exp(\lambda_1)$. The full expression is just the reciprocal of this. For $\epsilon_1 \to -\infty$, $\exp(\epsilon_1) \to 0$, and $1/\exp(\epsilon_1) \to \infty$. Hence, the denominator of the expression above may be written as $\exp(\infty \times \exp(\lambda_1))\exp(\lambda_1) = \exp(\infty)\exp(\lambda_1) = \infty$. The full expression is just the reciprocal of this, i.e., 0. Thus,

$$\begin{aligned}\left.\frac{1}{\exp\left[\frac{1}{\exp(\epsilon_1)}\exp(\lambda_1)\right]\exp(\lambda_1)}\right|_{-\infty}^{\infty} &= \frac{1}{\exp(\lambda_1)} - 0 \\ &= \exp(-\lambda_1)\end{aligned}$$

Substitution gives

$$\begin{aligned}\exp(-\lambda_1) &= \exp\left[-\ln\left(\sum_k \frac{e^{V_k}}{e^{V_1}}\right)\right] \\ &= \frac{e^{V_1}}{\sum_k e^{V_k}}\end{aligned}$$

In general terms, this long derivation produces the following functional form for the PCM:

$$\Pr(y = m) = \frac{e^{v_m}}{\sum_k e^{v_k}} \tag{11.1}$$

This is the basic formula for the Luce model.[3]

[3]Note that the derivation shown here deviates from Luce and draws from McFadden (1974, 1981). However, the final model is identical.

Table 11.1: Conditional versus Multinomial Logit

Model	Predictors	Parameters
Conditional Logit	Variable across choices	Fixed across choices
Multinomial Logit	Fixed across choices	Variable across choices

Modeling the Choice Probabilities

The Luce model does not specify a model for the v_m. The advantage of modeling these components is that we can determine how individual characteristics and attributes of the alternatives influence the choice probabilities. Ultimately, political choice models are about understanding the influences on choice behavior.

In practice, there are two main approaches to modeling the v_m (see Table 11.1). A first approach is to let the values of predictors vary across choices, while the parameters are constant across choices. This produces the so-called **conditional logit model**, which was developed by Daniel McFadden (1974, 1981). A second approach is to keep the values of predictors constant across choices, as would be the case with individual characteristics (but, of course, not choice attributes), while the parameters vary across choices. This produces the so-called **multinomial logit** model, which has a long history in the social sciences (and is the near-exclusive focus of Long's chapter on multinomial choice models). Both of these models can be considered special cases of the Luce model, in that they propose specific functional forms for the v_m. The two models can be combined in a hybrid model, as we shall discuss.

Conditional Logit Let π_{im} be the choice probability for alternative m in individual i.[4] As we have seen, π_{im} depends on v_{im}. In the conditional logit model,

$$v_{im} \quad = \quad \boldsymbol{x}_{im}\boldsymbol{\beta}$$

[4]Since we will be introducing individual characteristics, it is now useful to introduce individual-level subscripts. This will also help in discussing the data structure.

where $\boldsymbol{x}_{im}$ is a vector of attributes of the mth choice as perceived by the ith decision maker. Thus

$$\pi_{im} = \frac{\exp(\boldsymbol{x}_{im}\boldsymbol{\beta})}{\sum_{k=1}^{M}\exp(\boldsymbol{x}_{ik}\boldsymbol{\beta})} \tag{11.2}$$

We clearly see that the predictors vary across alternatives (whence the m subscript on $\boldsymbol{x}$). We also clearly see that the parameters are fixed across alternatives (whence the lack of a m subscript on $\boldsymbol{\beta}$).

Multinomial Logit In the multinomial logit model, the v_{im} are understood in terms of individual characteristics whose effects may vary across alternatives. Thus

$$v_{im} = \boldsymbol{z}_i\boldsymbol{\gamma}_m$$

Hence,

$$\pi_{im} = \frac{\exp(\boldsymbol{z}_i\boldsymbol{\gamma}_m)}{\sum_{k=1}^{M}\exp(\boldsymbol{z}_i\boldsymbol{\gamma}_k)} \tag{11.3}$$

These expressions clearly show that the predictors do not vary across alternatives (they lack a m subscript), but the parameters associated with those predictors do vary across alternatives (they are subscripted in m).

Hybrid Model Most statistical software packages that estimate the conditional logit model will actually allow you to include individual characteristics of the decision maker. This produces hybrid logit models where

$$v_{im} = \boldsymbol{x}_{im}\boldsymbol{\beta} + \boldsymbol{z}_i\boldsymbol{\gamma}_m$$

and

$$\pi_{im} = \frac{\exp(\boldsymbol{x}_{im}\boldsymbol{\beta} + \boldsymbol{z}_i\boldsymbol{\gamma}_m)}{\sum_{k=1}^{M}\exp(\boldsymbol{x}_{ik}\boldsymbol{\beta} + \boldsymbol{z}_i\boldsymbol{\gamma}_k)} \tag{11.4}$$

11.2.2 Identification

The models presented above are not identified. That is, we can arbitrarily change the vector of parameters without altering the implied choice probabilities. Thus, the same choice probability can map onto an infinitely large number of values for the parameters.

Multinomial Logit

The identification problem is easily demonstrated for the multinomial logit model. Imagine that instead of specifying a vector of parameters $\boldsymbol{\gamma}_m$ we add an arbitrary constant $\boldsymbol{\tau}$, how would this affect the choice probabilities? We now have

$$\begin{aligned}
\pi_{im} &= \frac{\exp[\boldsymbol{z}_i(\boldsymbol{\gamma}_m + \boldsymbol{\tau})]}{\sum_{k=1}^{M} \exp[\boldsymbol{z}_i(\boldsymbol{\gamma}_k + \boldsymbol{\tau})]} \\
&= \frac{\exp(\boldsymbol{z}_i\boldsymbol{\gamma}_m + \boldsymbol{z}_i\boldsymbol{\tau})}{\sum_{k=1}^{M} \exp(\boldsymbol{z}_i\boldsymbol{\gamma}_k + \boldsymbol{z}_i\boldsymbol{\tau})} \\
&= \frac{\exp(\boldsymbol{z}_i\boldsymbol{\gamma}_m)}{\sum_{k=1}^{M} \exp(\boldsymbol{z}_i\boldsymbol{\gamma}_k)} \times \frac{\exp(\boldsymbol{z}_i\boldsymbol{\tau})}{\exp(\boldsymbol{z}_i\boldsymbol{\tau})}
\end{aligned}$$

It is clear that the second term on the right-hand side is equal to one and hence does not change the probabilities on the left-hand side. Thus, we see that it is possible to arbitrarily change the parameters without changing the implied choice probabilities. This means that starting with a set of choice probability estimates (e.g., proportions of individuals who choose a particular alternative), it is impossible to derive a unique set of parameter estimates. The model is unidentified, at least, if we do not impose some restrictions on the parameters.

In practice, two kinds of constraint are imposed on the parameters. First, we could impose a normalizing constraint, such that the parameters sum to 0. Second, one can constrain to zero the parameters for one of the alternatives. It is arbitrary which alternative is constrained in this manner. This alternative becomes the baseline category to which we can compare the other alternatives. The second type of constraint is more common.

To see the implications of the second type of constraint, imagine that we set $\boldsymbol{\gamma}_1 = \mathbf{0}$, i.e. we constrain the parameters for the first alternative. Then

$$\begin{aligned}
\pi_{i1} &= \frac{\exp(\boldsymbol{z}_i\mathbf{0})}{\exp(\boldsymbol{z}_i\mathbf{0}) + \sum_{k=2}^{M} \exp(\boldsymbol{z}_i\boldsymbol{\gamma}_k)} \\
&= \frac{1}{1 + \sum_{k=2}^{M} \exp(\boldsymbol{z}_i\boldsymbol{\gamma}_k)}
\end{aligned}$$

For the remaining alternatives,

$$\pi_{im} = \frac{\exp(\boldsymbol{z}_i\boldsymbol{\gamma}_m)}{1 + \sum_{k=2}^{M} \exp(\boldsymbol{z}_i\boldsymbol{\gamma}_k)}$$

where $m \neq 1$. When we now add an arbitrary constant to $\boldsymbol{\gamma}_m$, then the choice probabilities will be affected. Thus, the parameters are identified.

Conditional Logit

The conditional logit model also has an identification problem but this is more localized, pertaining to the constant term only. To see this problem, let us partition $\boldsymbol{x}_{im}$ into $(1\ \boldsymbol{x}^*_{im})$, where $\boldsymbol{x}^*_{im}$ contains information about the covariates and 1 is the constant. The parameter vector is partitioned in a similar manner: $\boldsymbol{\beta}' = (\beta_0\ \boldsymbol{\beta}^{*\prime})$. We can then write

$$\begin{aligned}
\pi_{im} &= \frac{\exp(\boldsymbol{x}_{im}\boldsymbol{\beta})}{\sum_k \exp(\boldsymbol{x}_{ik}\boldsymbol{\beta})} \\
&= \frac{\exp(\beta_0 + \boldsymbol{x}^*_{im}\boldsymbol{\beta}^*)}{\sum_k \exp(\beta_0 + \boldsymbol{x}^*_{ik}\boldsymbol{\beta}^*)} \\
&= \frac{\exp(\beta_0)\exp(\boldsymbol{x}^*_{im}\boldsymbol{\beta}^*)}{\sum_k \exp(\beta_0)\exp(\boldsymbol{x}^*_{ik}\boldsymbol{\beta}^*)} \\
&= \frac{\exp(\boldsymbol{x}^*_{im}\boldsymbol{\beta}^*)}{\sum_k \exp(\boldsymbol{x}^*_{ik}\boldsymbol{\beta}^*)} \times \frac{\exp(\beta_0)}{\exp(\beta_0)}
\end{aligned}$$

The second term on the right-hand side of the last equation is always one, no matter what value of β_0 is selected. As such, this term does not affect the choice probabilities. Thus, we can pick any value for β_0 and will still be able to recover the choice probabilities for the alternatives. This means that no unique estimate of β_0 can be derived from the choice probabilities. The solution to this identification problem is that β_0 is typically normalized at 0 and will not be estimated.

11.2.3 Estimation

Estimation of the multinomial/conditional logit models is straightforward. We have a series of choice probabilities π_{im} associated with each of the dummies y_{im}. The likelihood function is then given by

$$\begin{aligned}
\mathcal{L} &= \prod_{y_{i1}=1} \pi_{i1} \prod_{y_{i2}=1} \pi_{i2} \cdots \prod_{y_{iM}=1} \pi_{iM} \\
&= \prod_{i=1}^{n} \prod_{m=1}^{M} \pi_{im}^{y_{im}}
\end{aligned}$$

Here π_{im} is given by (11.2) for the conditional logit model, by (11.3) for the multinomial logit model, and by (11.4) for the hybrid model. The log-likelihood function is given by

$$\ell = \sum_{i=1}^{n} \sum_{m=1}^{M} y_{im} \ln \pi_{im}$$

For the multinomial logit model, the derivatives have the following simple form:

$$\frac{\partial \ell}{\partial \boldsymbol{\gamma}'_m} = \sum_i (y_{im} - \pi_{im}) \boldsymbol{z}_i$$

for $m = 1, \cdots, M$. For the conditional logit model, the derivatives are:

$$\frac{\partial \ell}{\partial \boldsymbol{\beta}'} = \sum_i \sum_m y_{im} (\boldsymbol{x}_{im} - \bar{\boldsymbol{x}}_{im})$$

where $\bar{\boldsymbol{x}}_{im} = \sum_m \pi_{im} \boldsymbol{x}_{im}$. The first-order condition requires that both sets of derivatives are set equal to zero. Solving the resulting equations requires numerical methods, but due to the straightforward nature of the derivatives convergence is generally fast and unproblematic.

11.2.4 Interpretation

Multinomial Logit

Marginal Effects and Elasticities One way to interpret multinomial logit results is to compute marginal effects. With a little calculus it can be shown that

$$\frac{\partial \pi_m}{\partial z_l} = \pi_m \left(\gamma_{lm} - \sum_k \gamma_{lk} \pi_k \right)$$

(i-subscripts have been suppressed so as not to clutter the notation). Here z_l is a particular attribute of the decision maker and γ_{lk} (for $k = 1, \cdots, M$) is the effect of this attribute on π_k. It is assumed that z_l is continuous. We can interpret the marginal effect as the change in the probability of choosing alternative m for a very small change in z_l. Note that the marginal

effect depends on many factors, including the probabilities of choosing other alternatives and the effect of z_l on those choice probabilities. Thus, it is not necessarily the case that the marginal effect will be of the same sign as $\hat{\gamma}_{lm}$.

As usual, the computation of marginal effects requires that we fix the values of the predictors. It is common to set the predictors to their means, which results in marginal effects at the mean. It is also possible to compute average marginal effects, which requires that we use the covariate values for each observation in the marginal effects formula and then average the results. By using the delta method, the approximate asymptotic variance of the marginal effects can be computed (see Greene 2003). This can be used to calculate confidence intervals.

It is often helpful to transform marginal effects into elasticities. For the predictor z_l, the elasticity with respect to π_m is given by

$$\kappa_{z_l} = z_l \left(\gamma_{lm} - \sum_k \gamma_{lk} \pi_k \right)$$

This can be interpreted as the percentage change in π_m that arises from a one percentage point increase in z_l.

Predicted Probabilities and Discrete Change in Predicted Probabilities In the multinomial logit model, predicted probabilities are a convenient way of assessing the impact of predictors. Assuming the first alternative is used as the baseline, we have

$$\pi_{i1} = \frac{1}{1 + \sum_{k=2}^{M} \exp(\boldsymbol{z}_i \boldsymbol{\gamma}_k)}$$

For the non-baseline categories, we have

$$\pi_{im} = \frac{\exp(\boldsymbol{z}_m \boldsymbol{\gamma}_j)}{1 + \sum_{k=2}^{M} \exp(\boldsymbol{z}_i \boldsymbol{\gamma}_k)}$$

for $m = 2, 3, \cdots, M$.

There are many moving parts in these equations. To insulate the effect of a single predictor, z_l, it is conventional to let it vary between values a and b while holding all of the remaining covariates constant, usually at their mean values (but other choices are possible). This allows the computation of the

discrete change in the predicted probabilities, which is given by

$$\frac{\Delta \pi_m}{\Delta z_l} = \Pr(y_m = 1|\bar{\boldsymbol{z}}, z_l = b) - \Pr(y_m = 1|\bar{\boldsymbol{z}}, z_l = a)$$

Common choices for the limits of z_l include the following.

1. $a = 0$ and $b = 1$; this is particularly useful when ascertaining the effect of dummy predictors.
2. $a = \bar{z}_l$ and $b = \bar{z}_l + 1$; this gives a unit change from the mean of z_l.
3. $a = \bar{z}_l - .5$ and $b = \bar{z}_l + .5$; this is the centered unit change.
4. $a = \bar{z}_l - .5 s_{z_l}$ and $b = \bar{z}_l + .5 s_{z_l}$; this represents a standard deviation change.
5. $a = min(z_l)$ and $b = max(z_l)$; this represents the maximum change

In each case, it is possible to compute the variance of the discrete change via the delta method. This allows one to test whether the change is significantly different from zero.

To summarize the changes in predicted probabilities across all of the alternatives, it is possible to compute an average absolute discrete change measure (see Long 1997; Long and Freese 2003):

$$\bar{\Delta} = \frac{1}{M} \sum_{m=1}^{M} \left| \frac{\Delta \pi_m}{\Delta z_l} \right|$$

This measure gives an overall impression of how a predictor affects the probabilities of choosing different alternatives.

The Odds Ratio Finally, the odds ratio is a method of interpretation that can be advantageous. The odds give the probability of choosing one alternative rather than another given a particular set of values of the covariates. One type of odds involves the baseline category. If we assume, without loss of generality, that the first alternative serves as the baseline, then the odds relative to this alternative are given by

$$\begin{aligned} \Omega_{m|1} &= \frac{\pi_m}{\pi_1} \\ &= \exp(\boldsymbol{z}_i \boldsymbol{\gamma}_m) \end{aligned}$$

A second type of odds compares two non-baseline categories:

$$\begin{aligned}\Omega_{m|k} &= \frac{\pi_m}{\pi_k} \\ &= \exp[\boldsymbol{z}_i(\boldsymbol{\gamma}_m - \boldsymbol{\gamma}_k)]\end{aligned}$$

Now imagine that we compute the odds under two different scenarios. In the first scenario, we consider a value of z_l for one of the predictors while in the second scenario that predictor is set to $z_l + \delta$. The remaining predictors are kept constant across the two scenarios. The odds ratio involving the baseline category is now given by

$$\frac{\Omega_{m|1}(\boldsymbol{z}, z_l + \delta)}{\Omega_{m|1}(\boldsymbol{z}, z_l)} = \exp(\gamma_{lm}\delta)$$

For non-baseline categories, the odds ratio is

$$\frac{\Omega_{m|k}(\boldsymbol{z}, z_l + \delta)}{\Omega_{m|k}(\boldsymbol{z}, z_l)} = \exp[(\gamma_{lm} - \gamma_{lk})\delta]$$

This expression shows a number of nice features of the odds ratio. First, it does not depend on the level of z_l, just on the change in this variable. Second, it does not depend on any of the other predictors (unlike marginal effects and discrete changes). Also note that the ratio is driven by the differences in the effects of a predictor across two alternatives. This forms an interesting counter-point to the odds ratio for the conditional logit model.

Conditional Logit

Marginal Effects and Elasticities For the conditional logit model, a bit of calculus shows that the marginal effect of x_{kl} with respect to π_m is given by

$$\frac{\partial \pi_m}{\partial x_{lk}} = \pi_m \left(d - \pi_k\right) \beta_l$$

where $d = 1$ if $m = k$ and $d = 0$ otherwise. We see that the marginal effect depends on all attribute sets (through π_m and π_k). Depending on how we select the values of the covariates, either marginal effects at the mean or average marginal effects can be computed.

It is often helpful to convert the marginal effect into an elasticity. The formula is

$$\kappa_{x_{lk}} = x_{lk}\,(d - \pi_k)\,\beta_l$$

This can be interpreted as the percentage change in π_m for a one percentage point increase in x_{lk}. When $k = m$ the expression for κ presents a **self-elasticity**: it gives the impact on the probability of choosing alternative m when an attribute of this alternative changes. When $k \neq m$, then κ gives the **cross-elasticity**, i.e. the impact on the probability of choosing alternative m when the attribute of some other alternative changes. We say that two alternatives are *complements* if their cross-elasticity is negative. We say that they are *substitutes* when their cross-elasticity is positive. Finally, the alternatives are *independent* when the cross-elasticity is zero.[5]

Predicted Probabilities and Discrete Change in Predicted Probabilities In the conditional logit model, the predicted probability is given by

$$\pi_m = \frac{\exp(\boldsymbol{x}_m\boldsymbol{\beta})}{\sum_{k=1}^{M}\exp(\boldsymbol{x}_k\boldsymbol{\beta})}$$

This can be parlayed into a discrete change measure via

$$\frac{\Delta\pi_m}{\Delta x_{lm}} = \Pr(y_m = 1|\boldsymbol{x}_m, x_{lm} = b) - \Pr(y_m = 1|\boldsymbol{x}_m, x_{lm} = a)$$

It is conventional to set the remaining predictors to their mean values, i.e. $\boldsymbol{x}_m = \bar{\boldsymbol{x}}_m$. In terms of the values a and b, these can be selected in the same way as in the multinomial logit model.

The Odds Ratio The odds is again defined as the ratio of the choice probabilities for two different alternatives. It is computed via

$$\Omega_{m|k} = \exp([\boldsymbol{x}_m - \boldsymbol{x}_k]\boldsymbol{\beta})$$

[5]In economics, cross-elasticities often refer to changes in the demand for one good as a result of changes in the price of another good. With complementary goods, a price increase in one leads to reduced demand for the other good. With substitutes, a price increase in one good leads to increased demand for the other good.

Note that this reflects an important difference with the multinomial logit model. That is, in the conditional logit model the odds are driven by the differences in the attributes. By contrast, in the multinomial logit model the odds are driven by the differences in parameters.

The odds ratio reflects what happens when we change the value of an attribute of an alternative from x_{lm} to $x_{lm} + \delta$. In this case,

$$\frac{\Omega_{m|k}(\boldsymbol{x}, x_{lm} + \delta)}{\Omega_{m|k}(\boldsymbol{x}, x_{lm})} = \exp(\delta\beta_l)$$

is the factor by which the odds of choosing m compared to k change.

11.2.5 Hypothesis Testing

A variety of hypotheses can be tested in the multinomial and conditional logit models. One hypothesis is that a particular covariate has a zero effect in the population. It is straightforward to test this hypothesis in the conditional logit model. If the null hypothesis is that x_{lm} has no effect, then we can simply test the simple hypothesis $H_0 : \beta_l = 0$ using a LR or Wald test procedure. In the multinomial logit model, the test is a bit more complex. Since each covariate has $M-1$ parameters associated with it, the hypothesis that a particular attribute of the decision maker, z_l, does not influence choice amounts to: $H_0 : \gamma_{l2} = \cdots = \gamma_{lM} = 0$ (assuming that the first alternative is the baseline). This joint hypothesis is most easily tested using a LR or Wald test approach.

A second type of hypothesis that can be useful to test involves an equality constraint whereby two parameters are restricted to be equal. In the multinomial logit model, this test can be used to ascertain the **distinguishability** of two alternatives. Two alternatives m and k are non-distinct if the effect of all of the covariates is the same for both alternatives, i.e. $H_0 : \gamma_{lm} - \gamma_{lk} = 0$ $\forall l$. In this case, no information is lost by combining the alternatives. Long (1997) suggests a LR test for evaluating the null hypothesis that proceeds as follows:

1. Limit the choice set to m and k.

2. Perform a binary logit analysis on this choice set, using the same predictors as in the multinomial logit model.

3. Perform an LR test on the hypothesis that all of the predictors have null effects. Failure to reject this hypothesis constitutes evidence that alternatives m and k are not distinct.

An alternative approach is to perform a Wald test; this will be illustrated in the example below.

11.2.6 Model Fit

The fit of the multinomial and conditional logit models is typically ascertained using pseudo-R^2 measures. It is again useful to use the Aldrich and Nelson (1984) pseudo-R^2 measure, which is based on the LR test statistic, and to normalize it so that it has a range from 0 to 1. For this purpose, we can apply the Veall-Zimmermann correction that we discussed earlier.

As you will recall, the Veall-Zimmermann correction entails computing ℓ_0, the log-likelihood of an empty model. In multinomial logit analysis, the most common null model is one that omits all of the predictors and includes constants only. The log-likelihood of this model is given by

$$\ell_0 = \sum_{m=1}^{M} n_m \ln\left(\frac{n_m}{n}\right)$$

In conditional logit analysis, the most widely used null model is one that contains no parameters. Here, $\pi_{im} = 1/M$ and

$$\ell_0 = -n \ln M$$

11.2.7 Example

To illustrate the conditional logit, multinomial logit, and hybrid models we consider voting behavior in the Netherlands in the 1994 parliamentary elections for the 2nd Chamber—the main chamber of the legislature (and the only chamber that is directly elected). In this example, we model the vote for the four largest parties, to wit CDA, D66, PvdA, and VVD.[6] We consider

[6]CDA = Christian Democratic Appeal (Christian Democrats), D66 = Democrats 66 (left liberal), PvdA = Labor Party (Social Democrats), and VVD = People's Party for Freedom and Democracy (right liberal).

one party attribute: issue distance to the voter.[7] We consider six voter characteristics, which generalize across the alternatives: (1) age, (2) gender, (3) education, (4) income, (5) religiosity, and (6) left-right self placement. We first estimate a conditional logit model, next a multinomial logit model, and finally a hybrid model.

The analysis is based on the Dutch Parliamentary Election Study (DPES), 1994. After eliminating cases with missing values on the predictors or cases with a vote choice different from CDA, D66, PvdA, or VVD, the sample size is 993 observations for the multinomial logit analysis. For the conditional logit model, the sample size is discussed below.

Multinomial Logit Analysis

Estimation Results We begin by specifying a multinomial logit analysis in which vote choice is predicted from the individual level characteristics (age, gender, education, income, religiosity, and left-right self-placement). The baseline category is voting for the PvdA—i.e. the parameters for this choice have been set equal to 0. The model was estimated using the `mlogit` command:

```
mlogit vote age educ income male religious leftright, b(1)
```

where `b(1)` identifies the first category of `vote` as the baseline category, which corresponds to PvdA. The estimation results are shown in Table 11.2.

Interpretation To interpret the results I shall focus first on marginal effects at the mean and their corresponding elasticities, which can be obtained using the familiar `mfx` command. Table 11.3 shows the marginal effects with respect to voting for the CDA. For example, a very small increase in L-R self-placement, which corresponds to a movement to the political right, increases the probability of voting for the CDA by about .06 points. The corresponding elasticity is 1.7, which means that a one percent increase in left-right self-placement corresponds to a 1.7% increase in the probability of voting for the CDA.

We now consider discrete change in the predicted probabilities. Maximum discrete change measures were obtained using the `prchange` command and

[7] This is measured in terms of five issues: (a) euthanasia, (b) crime, (c) reduction of income differences, (d) nuclear power, and (e) integration of minorities.

Table 11.2: Multinomial Logit Model of Dutch Vote Choice in 1994

Predictor	CDA	VVD	D66
Age	.011	$-.017^{*}$	$-.044^{**}$
Education	$.197^{+}$	$.281^{**}$	.057
Income	$.088^{**}$	$.134^{**}$	.043
Male	.056	.082	$-.120$
Religious	2.357^{**}	.282	$.387^{+}$
L-R Placement	$.750^{**}$	$.893^{**}$	$.250^{**}$
Constant	-7.652^{**}	-6.300^{**}	$-.186$

Notes: $n = 993$. Table entries are ML multinomial logit coefficients. PvdA is the baseline category. $^{**}p < .01,^{*}\, p < .05,^{+}\, p < .10$ (two-tailed).

Table 11.3: Marginal Effects for the Multinomial Logit Vote Choice Model

Predictor	$\frac{\partial \pi}{\partial x}$	$s.e.$	κ
Age	$.005^{**}$	.001	1.123
Education	.013	.013	.203
Income	.005	.004	.185
Male	.010	.027	.029
Religious	$.315^{**}$	.028	.937
L-R	$.058^{**}$	.008	1.722

Notes: Marginal effects at the mean computed on the probability of voting for CDA. For male and religious the marginal effect is computed as the discrete change in π. Marginal effect estimates obtained via `mfx, predict(p outcome(2))`. Elasticity estimates obtained via `mfx, predict(p outcome(2)) eyex`. $^{**}p < .01$ (two-tailed).

Table 11.4: Discrete Change for the Multinomial Logit Vote Choice Model

Predictor	PvdA	CDA	VVD	D66	$\bar{\Delta}$
Age	.213	.316	−.076	−.453	.264
Education	−.146	.051	.165	−.069	.108
Income	−.203	.053	.197	−.047	.125
Male	−.001	.010	.021	−.031	.016
Religious	−.175	.315	−.084	−.055	.157
L-R	−.749	.242	.678	−.171	.460

Notes: Maximum discrete changes reported. All other predictors are set to their means. Estimates obtained via `prchange var, rest(mean)`.

are reported in Table 11.4. We observe particularly sizable effects for age, left-right self-placement and religion. For example, the oldest respondent is predicted to be .45 less likely to vote for D66 than the youngest respondent, suggesting the appeal of this party among young but not older voters. Moving from the most left-wing to the most right-wing ideological position reduces the likelihood of voting for the PvdA by .75 points on the average, while increasing the likelihood of voting for the VVD by .68 points. Religious respondents on the average have a .32 greater probability of voting for the CDA than non-religious respondents. This is testimony to the explicitly Christian foundation of the CDA, whereas the remaining parties are secular in nature.

Finally, consider the odds ratios reported in Table 11.5, which were obtained using the `listcoef` command. These ratios draw comparisons of the remaining alternatives with the PvdA. For example, the odds of choosing the CDA in lieu of the PvdA are over 10 times greater for a religious as compared to a non-religious person.

The `listcoef` command also allows one to draw comparisons between non-baseline categories. For example, one of the comparisons that is reported concerns the probability of voting for the CDA as compared to the VVD. These odds are almost 8 times greater for religious than for non-religious people. We could also have computed this result directly from Table 11.5: $\Omega_{2|1}/\Omega_{3|1} = \Omega_{2|3} = 10.559/1.325 \approx 8$.

Table 11.5: Odds Ratios for the Multinomial Logit Vote Choice Model

Predictor	CDA	VVD	D66
Age	1.012	.983	.957
Education	1.218	1.325	1.059
Income	1.092	1.143	1.044
Male	1.058	1.086	.887
Religious	10.559	1.325	1.473
L-R Placement	2.117	2.443	1.284

Notes: Comparisons with the baseline category (PvdA). Odds ratios obtained via `listcoef`.

Hypothesis Testing Looking through the results in Table 11.2, we see that gender is never statistically significant. We can test this more formally. Let γ_{4m} be the coefficient for male. Then we can test $H_0 : \gamma_{4m} = 0$ for $m = 2, 3, 4$. We do this in Stata by typing

```
test male
```

This produces $W = 1.02$, $p = .797$. Thus, we fail to reject the null hypothesis that gender had no impact on major party vote choice in the Netherlands in 1994.

To illustrate a Wald test approach to distinguishability let us consider if the alternatives PvdA and D66 can be collapsed. The relevant null hypothesis is $H_0 : \gamma_{l1} = \gamma_{l4}$ for $l = 1, 2, \cdots, 6$. To test this hypothesis in Stata, we begin by estimating the multinomial logit model with a baseline other than alternative 1 (PvdA) or 4 (D66). For example, we could make the CDA the baseline and estimate `mlogit vote age religious educ male income leftright, b(2)`. We then execute the following command

```
test[1=4]
```

The resulting Wald test statistic is 64.56 with $p < .01$. Thus, we reject the hypothesis that the alternatives of PvdA and D66 are indistinguishable and can be collapsed into a single alternative.

Table 11.6: Person-Choice Matrix

Id	Party	Choice	Issues	L-R	Age
1	PvdA	0	6.00	7	29
1	CDA	1	2.00	7	29
1	VVD	0	6.00	7	29
1	D66	0	6.00	7	29
2	PvdA	1	2.00	3	56
2	CDA	0	7.00	3	56
2	VVD	0	11.00	3	56
2	D66	0	2.00	3	56

Conditional Logit Analysis

Estimation Results Let us now consider a conditional logit analysis in which vote choice is predicted from issue distance and the party's seat share from the previous election (measured as a percentage of the total seats, which is 150). We can estimate the conditional logit model using Stata's `clogit` command. However, before we can do this we need to restructure the data into **person-choice matrix**.

As opposed to a normal data matrix, which contains one row per individual, a person-choice matrix contains multiple rows for each individual, each row corresponding to an alternative in the choice set. Table 11.6 shows a fragment of a person-choice matrix for the DPES data. This example illustrates several points very clearly. First, each individual appears four times, i.e. as often as there are alternatives in the choice set. Second, a dichotomous variable ("Choice") keeps track of whether a respondent chose a particular alternative. This corresponds to the y_{im} variable described before. Third, party attributes vary across the rows belonging to the same individual. Fourth, individual characteristics are constant across the rows belonging to the same individual.

Creating a person-choice matrix is greatly simplified through the `mclgen` program of John Hendrickx. In this case, we can simply run

```
mclgen vote
```

where `vote` is the choice variable in the original data set. Since the `mclgen` command transforms the data, you should make sure that the original data

is saved before you run the command.[8] The `mclgen` command adds two new variables to the data set: (1) `_strata` is a person identifier (`id` in Table 11.6) and (2) `_didep` is a choice variable (`choice` in Table 11.6). The new data set has $4n$ observations or, in general Mn observations. Due to missing data on the issue distance variables, $4n = 2340$ for our analysis.

We can now run the conditional logit analysis using Stata's `clogit` command. The syntax for our model is

```
clogit _didep dist seat89, group(_strata)
```

This command produces the results reported in Table 11.7. The results show a positive effect of prior seat share, suggesting that larger parties tend to attract more votes. They show a negative effect for issue distance, which means that the probability of voting for a party declines the further that party is removed from the voter.

Interpretation We begin by interpreting the results using discrete change in the predicted probabilities. Unfortunately, the `prchange` command does not work for conditional logit analysis, so we will have to create the predictions ourselves. Setting 1989 seat share and issue distances to the mean values for each party and varying the issue distance for the PvdA from minimum to maximum, we obtain the following changes in the predicted probabilities: -.932 for PvdA, .239 for CDA, .226 for VVD, and .467 for D66.[9] Thus, hold-

[8] After running the command, you may have to collapse some variables in the original data. Specifically, if you created separate variables for each party's attributes, you would want to stack them. For instance, imagine that we had created four different issue distance variables, one for each of the parties. We now want to combine them into a single issue distance variable. This can be done as follows:

```
gen dist=.
replace dist=pvdadist if vote==1
replace dist=cdadist if vote==2
replace dist=vvddist if vote==3
replace dist=d66dist if vote==4
```

where `pvdadist`, `cdadist`, `vvddist`, and `d66dist` are the individual vote share variables.

[9] The mean issue distance is 10.3 for CDA, 9.3 for VVD, and 7.1 for D66. The 1989 seat share is 32.7 for PvDA, 36 for CDA, 14.7 for VVD, and 8 for D66. To compute

Table 11.7: Conditional Logit Model of Dutch Vote Choice in 1994

Predictor	Estimate	SE
Issue Distance	$-.393^{**}$	.023
Seat Share 1989	$.021^{**}$	.004

Notes: $n = 2340$. $^{**}p < .01$ (two-tailed). Estimates generated via `clogit _didep dist seat89, group(_strata)`.

ing all else constant, there is a sizable decline in the probability of voting for the PvdA, which benefits D66 the most.

The results can also be interpreted using marginal effects and elasticities. While it is possible to use the `mfx` command to compute marginal effects, these are not of the same variety as those described in the formula earlier.[10] But it is easy to compute the marginal effects and elasticities by hand.[11] The resulting marginal effects are shown in Table 11.8. These clearly show the pattern of declining vote probabilities for a party when the issue distance

the predicted vote probability for the PvdA when the issue distance to this party is zero (minimum), one computes

$$\frac{e^{-.393\times 0+.021\times 32.7}}{e^{-.393\times 0+.021\times 32.7}+e^{-.393\times 10.3+.021\times 36}+e^{-.393\times 9.3+.021\times 14.7}+e^{-.393\times 7.1+.021\times 8}}$$

At the maximum issue distance (29) this becomes

$$\frac{e^{-.393\times 29+.021\times 32.7}}{e^{-.393\times 29+.021\times 32.7}+e^{-.393\times 10.3+.021\times 36}+e^{-.393\times 9.3+.021\times 14.7}+e^{-.393\times 7.1+.021\times 8}}$$

The implications of changing the distance to the PvdA for the vote probabilities for other parties can be computed in a similar manner.

[10]The `mfx` command does not compute marginal effects for individual observations and, what is more, assumes that the fixed effect is zero. See the Stata manuals for details.

[11]To compute marginal effects at the mean, one starts by computing the predicted probabilities for all of the alternatives while setting all of the predictors equal to their alternative-specific means. The mean issue distances are 9.1, 10.3, 9.3, and 7.1 for PvdA, CDA, VVD, and D66, respectively. The seat shares for these parties are 32.7, 36, 14.7, and 8, respectively. The resulting predicted probabilities are .279 for PvdA, .187 for CDA, .178 for VVD, and .355 for D66. Once the predicted probabilities have been computed, then the marginal effects are computed in a straightforward manner. For example, the effect of a change in the issue distance with respect to PvdA on the probability of voting

Table 11.8: Marginal Effects for the Conditional Logit Vote Choice Model

	Change in Probability for			
Change in Issue Distance	**PvdA**	**CDA**	**VVD**	**D66**
PvdA	−.079	.021	.020	.039
CDA	.021	−.060	.013	.026
VVD	.020	.013	−.058	.025
D66	.039	.026	.025	−.090

Notes: Table entries are marginal effects at the mean.

to that party increases by a small amount. The table also shows that other parties benefit differentially from such increases in the issue distance.

Table 11.9 converts the marginal effects into elasticities. Here we see that a one percent increase in the issue distance to the PvdA reduces the probability of voting for this party by roughly 2.6%. Such an increase boosts the probability of voting for the CDA by almost 1 percentage point. The marginal effects reveal an important regularity. The cross-elasticities are identical for a change in the predictor. For example, a change in the issue distance with respect to the PvdA results in identical cross-elasticities for CDA, VVD, and D66. This is a direct consequence of the IIA assumption, as we shall see below.

Finally, the results can be interpreted using the odds ratio. This ratio can be obtained via the `listcoef` command. This yields a factor change in

for that party is

$$\frac{\pi_1}{x_{11}} = .279 \times (1 - .279) \times -.393$$

where π_1 is the probability of a PvdA vote and x_{11} is the issue distance for the PvdA. Similarly, the effect of a change in the issue distance with respect to the PvdA on the probability of voting for the CDA is

$$\frac{\pi_2}{x_{11}} = .187 \times (0 - .279) \times -.393$$

The elasticity in the PvdA vote probability with respect to PvdA issue distance is $9.1 \times (1 - .279) \times -.393$. The elasticity in the CDA vote probability with respect to PvdA issue distance is $9.1 \times (0 - .279) \times -.393$.

Table 11.9: Elasticities for the Conditional Logit Vote Choice Model

	Change in Probability for			
Change in Issue Distance	**PvdA**	**CDA**	**VVD**	**D66**
PvdA	−2.573	.995	.995	.995
CDA	.755	−2.935	.755	.755
VVD	.647	.647	−2.989	.647
D66	.997	.997	.997	−1.811

Notes: Table entries are elasticities computed at the mean. The diagonal elements are self-elasticities, while the off-diagonal elements are cross-elasticities.

the odds of .675 for issue distance. Thus the odds of choosing, for example, PvdA compared to CDA decrease by a factor .675 when issue distance to the PvdA increases by one unit. This corresponds to a 32.5% decrease in the odds, which can be considered sizable.

Hybrid Model

Let us now combine the party and individual attributes into a single model. We now need to make sure that the individual attributes are interacted with a set of dummies for the alternatives, so that the effects of those attributes can vary across choice options. Interaction expansion is the easiest approach here. Specifically, we can issue the command

```
xi:  clogit _didep dist i.vote i.vote|age i.vote|educ
i.vote|income i.vote|male i.vote|religious i.vote|leftright,
group(_strata)
```

This produces the results in Table 10.10. Note that PvdA once again serves as the baseline category: the parameters for this alternative are all fixed at zero.[12] Interpretation proceeds as before for the conditional and multinomial components of the model.

[12]I have dropped the seat share variable since it causes perfect collinearity and has a small effect to begin with.

Table 11.10: Hybrid Lucean Model of Dutch Vote Choice in 1994

Predictor	**Estimate**	**SE**
Issue Distance	$-.352^{**}$	.027
Age × CDA	.002	.013
Age × VVD	$-.020$	.013
Age × D66	$-.054^{**}$	.012
Education × CDA	$-.017$	.169
Education × VVD	.190	.166
Education × D66	$-.009$	.135
Income × CDA	$.123^{*}$	.060
Income × VVD	$.093^{+}$	.056
Income × D66	.022	.044
Male × CDA	.029	.376
Male × VVD	$-.258$	.371
Male × D66	$-.005$	.295
Religiosity × CDA	2.303^{**}	.439
Religiosity × VVD	$.681^{+}$	.372
Religiosity × D66	$.696^{*}$	.302
Left-Right × CDA	$.598^{**}$	.114
Left-Right × VVD	$.680^{**}$	.114
Left-Right × D66	$.277^{**}$	.089
CDA	-5.627^{**}	1.219
VVD	-4.351^{**}	1.142
D66	$-.050$	.841

Notes: $n = 2124$. $^{**}p < .01$, $^{*}p < .05$, $^{+}p < .10$ (two-tailed).

11.2.8 The IIA Assumption

The Meaning of IIA

Independence of irrelevant alternatives (IIA) means that the ratio of the choice probabilities for two alternatives, m and k, is independent from all other alternatives, l, in the choice set. That is, regardless of whether those alternatives are in or out off the choice set, the probability of choosing m compared to the probability of choosing k remains the same. This property is built into the Luce model:

$$\begin{aligned} \frac{\pi_{im}}{\pi_{ik}} &= \left[\frac{\exp(V_{im})/\sum_l \exp(V_{il})}{\exp(V_{ik})/\sum_l \exp(V_{il})}\right] \\ &= \frac{\exp(V_{im})}{\exp(V_{ik})} \end{aligned}$$

We see that the ratio of choice probabilities for m and k is driven entirely by the utilities of those alternatives, not by anything else. It does not matter if we model these utilities in terms of the multinomial logit model or in terms of the conditional logit model.

The IIA assumption is attractive from a computational point of view. It means that we can assume independent errors across alternatives, which greatly simplifies the log-likelihood function. However, from a substantive perspective the assumption is often quite unattractive. It implies certain empirical regularities that may not hold in the empirical world, especially when it comes to political choice behavior.

Consider, for example, a political system with three political parties. Party R operates on the right of the political spectrum. Parties $L1$ and $L2$ operate on the left of the political spectrum, roughly in the same spot. Imagine that a voter treats $L1$ and $L2$ equivalently because they operate in the same ideological location. Imagine further that the voter is indifferent between the right-wing party and the parties on the left. We then would expect choice probabilities $\pi_R = .5$, $\pi_{L1} = .25$, and $\pi_{L2} = .25$—the voter is equally likely to vote for the left and right, but two equally attractive parties on the left cause the vote to be split there. The odds of voting R versus $L1$ thus depends on the presence of $L2$. If $L2$ is present, $\pi_R/\pi_{L1} = 2$. However, if $L2$ should cease to exist, then $\pi_R/\pi_{L1} = 1$ since the voter now simply faces a choice between a party from the right and from the left, both of which are equally attractive. This seems a reasonable adjustment in the voter's

behavior, but it violates the IIA assumption. Under IIA, the voter should be twice as likely to vote for R than for $L1$, even if $L2$ ceases to exist. This kind of "foolish consistency" in behavior seems maladaptive, but that is what IIA implies.

The reason we are concerned about the IIA assumption is that it affects the consistency of the estimators of the multinomial and conditional logit models. The estimators for these models are consistent only if the IIA assumption holds. If the IIA assumption is incorrect, as will often be the case, then consistency of the estimators is not guaranteed. This is sufficient reason to test whether the IIA assumption holds. Several test procedures are available.

Testing the IIA Assumption

The Hausman-McFadden Test The Hausman-McFadden (1984) test is based on a simple intuition: if IIA holds, then the estimates obtained from considering two alternatives in isolation should be roughly comparable to those obtained in the full choice set. After all, under IIA, the ratio π_{ij}/π_{ik} is constant across alternative specifications of the choice set, which suggests that the parameters influencing the probabilities in this ratio are constant as well. Thus, Hausman and McFadden suggest estimating the parameters in a subset of the alternatives and again in the full set. A comparison of the estimates, taking into consideration sampling variability, can then shed light on the fate of the IIA assumption.

More formally, let M_f be a conditional, multinomial, or hybrid model that is estimated on the complete choice set. We assume that $\hat{\boldsymbol{\theta}}_f$ is consistent; that is, asymptotically $\boldsymbol{\theta}$ reduces to $\boldsymbol{\beta}$ in conditional logit and to $\boldsymbol{\gamma}_m$ in multinomial logit. Let M_s be a model estimated on the choice sub-set consisting of alternatives m and k. Under IIA, $\hat{\boldsymbol{\theta}}_s$ also is consistent. Under IIA, then, we would expect $\hat{\boldsymbol{\theta}}_s$ and $\hat{\boldsymbol{\theta}}_f$ both to converge to $\boldsymbol{\theta}$. Consequently, we would expect $\hat{\boldsymbol{\theta}}_s - \hat{\boldsymbol{\theta}}_f \stackrel{asy}{\rightarrow} \mathbf{0}$. If we now also take into consideration sampling fluctuation, then we can construct the following test statistic:

$$T_{HM} \;=\; (\hat{\boldsymbol{\beta}}_s - \hat{\boldsymbol{\beta}}_f)' \left[\hat{\mathbf{V}}_s - \hat{\mathbf{V}}_f\right]^{-1} (\hat{\boldsymbol{\beta}}_s - \hat{\boldsymbol{\beta}}_f) \stackrel{asy}{\sim} \chi^2_K \qquad (11.5)$$

where K is the number of predictors. Under IIA, $T_{HM} = 0$. As the IIA assumption becomes less reasonable, we will obtain larger values of the test

statistic; when T_{HM} exceeds the critical value, then we reject the null assumption of IIA.

It is important to point out that the test is valid only if $\hat{\boldsymbol{\theta}}_f$ is consistent and asymptotically efficient. It should also be stressed that all test results are asymptotic. It is somewhat dangerous and, as such, not recommended to rely on this test in small samples. If the sample size is small and/or $\hat{\boldsymbol{\theta}}_f$ is not consistent, then T_{HM} may come out negative. The same can happen when $\hat{\mathbf{V}}_s - \hat{\mathbf{V}}_f$, i.e. is not positive definite. This can happen when $\hat{\mathbf{V}}_s - \hat{\mathbf{V}}_f \to \mathbf{0}$, which is frequently the case. In this case, inversion of the variance-covariance matrix is also problematic, further adding to computational difficulties.

These computational problems can be illustrated for the conditional logit model from Table 11.7. It is sometimes argued that the presence of D66 changes the relative choice probabilities for the remaining alternatives. We could consider what happens if we drop this alternative from the choice set. The Stata syntax for doing so is

```
clogit _didep dist seat89, group(_strata)

est store F

clogit _didep dist seat89 if vote!=4, group(_strata)

est store S

hausman F S, alleq cons
```

The last command performs the hausman test on the estimates stored in `F` and `S`. The `alleq` option indicates that all parameters are hypothesized to be the same across the two models. The `cons` option tells Stata to include the constant in the test as well.[13] When we run the commands, Stata returns $T_{HM} = -11.52$, which is obviously a nonsensical value. It also warns us that the model may fail to meet the asymptotic assumptions of the Hausman-McFadden test.

Seemingly Unrelated Estimation Seemingly unrelated estimation or SUE is a methodology for combining estimates from different estimations that may or may not be partially overlapping and that may or may not be based

[13]This is not relevant for the conditional logit model because we are not estimating a constant. However, if the command were used for the multinomial logit model, then it is important to test the constants as well.

on the same estimation method. The SUE method creates a proper variance-covariance matrix of the sandwich type (White 1982, 1994). This is one of the distinctive advantages when compared to the Hausman-McFadden test because it means that tests based on SUE are less likely to return inadmissible results.

The statistical rationale of SUE can be found in Weesie (1999) and White (1982, 1994). Apart from the computation of the variance-covariance matrix, its logic in the context of testing the IIA assumption is rather similar to that of the Hausman-McFadden test. After running the conditional logit commands on the full and reduced choice sets and storing the results, the test proceeds as follows in Stata:

```
suest F S

test[F_didp=S_didep], cons
```

This produces a test statistic of 10.80 with $p < .05$, suggesting that the estimates for the complete and reduced choice sets are not the same. This is evidence that the IIA assumption is violated in the DPES, at least when the irrelevant alternative is defined as D66.

Small-Hsiao Test For the multinomial logit model, the IIA assumption may also be tested using the Small-Hsiao (1982) exact test. This is a correction of a LR test that compares the log-likelihood implied by the parameters of the full model to the log-likelihood of a restricted model that comes about by dropping one of the alternatives. This test was used by McFadden, Tye, and Train (1977) and is given by

$$LR \;=\; -2\,(\ell_f - \ell_s)$$

where ℓ_s is the log-likelihood for the restricted choice set and ℓ_f is the log-likelihood for the complete choice set.

Small and Hsiao (1982) pointed out that this test is problematic because it ignores the potential correlation between $\hat{\boldsymbol{\theta}}_s$ and $\hat{\boldsymbol{\theta}}_f$. This tends to bias the test in favor of the IIA assumption, i.e. it is less likely that the assumption is rejected. To overcome this problem, Small and Hsiao propose dividing the sample into two independent segments, A and B. Segment A is used to estimate $\hat{\boldsymbol{\theta}}_f$. Segment B is also used to estimate $\hat{\boldsymbol{\theta}}_f$ but, in addition, produces $\hat{\boldsymbol{\theta}}_s$. The estimates for the unrestricted choice set are combined using

$$\hat{\boldsymbol{\theta}}_f^{AB} \;=\; (1\sqrt{2})\hat{\boldsymbol{\theta}}_f^{A} + (1 - 1/\sqrt{2})\hat{\boldsymbol{\theta}}_f^{B}$$

This combined estimate is then used to compute the log-likelihood on sample B. This log-likelihood can then be compared to the restricted choice set log-likelihood to form

$$T_{SH} = -2\left(\ell_f^B - \ell_s^B\right) \tag{11.6}$$

In Stata, the Small-Hsiao test can be performed using the `smshiao` command that was written by Nick Winter. The command has the following basic syntax:

```
smhsiao depvar [ indepvars ], elim(yvalue)
```

where *yvalue* is the alternative that we wish to remove from the choice set. When we run the command eliminating D66 from the choice set, we obtain $T_{SH} = 10.80$, $p = .148$. Thus, we cannot reject the null hypothesis of IIA for the multinomial logit model.[14]

11.3 Multinomial Probit

11.3.1 Derivation

Choice Probabilities

We can remove the IIA assumption by allowing the stochastic elements of a decision maker's utilities over the alternatives to be correlated. Specifically, in the expression $y_m^* = v_m + \epsilon_m$, we can assume that the ϵs follow a multivariate normal distribution. More precisely, let $\boldsymbol{\epsilon}' = (\epsilon_1\ \epsilon_2 \cdots \epsilon_M)$ be a vector of random utility components over the alternatives. Then, it is assumed that $\boldsymbol{\epsilon} \sim N(\mathbf{0}, \boldsymbol{\Sigma})$, where

$$\boldsymbol{\Sigma} = \begin{pmatrix} \sigma_1^2 & \sigma_{12} & \cdots & \sigma_{1M} \\ \sigma_{12} & \sigma_2^2 & \cdots & \sigma_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{1M} & \sigma_{2M} & \cdots & \sigma_M^2 \end{pmatrix}$$

[14] The test divides the sample randomly in roughly equally sized groups. Since the groups can vary across iterations, so can the values of T_{SH}. Thus, I recommend running this a couple of times and to compare the results. Alternatively, one can add the `s(varname)` option, where `varname` specifies the name of a variable that you have created that splits the sample into two groups.

Note that there are two key differences with Lucean choice models: (1) the diagonal elements of $\mathbf{\Sigma}$ are not constrained to be identical and fixed and (2) the off-diagonal elements are not constrained to be zero.

We can now relate the distributional assumption to a choice mechanism. To focus our thoughts, I will again start by presenting the results for a choice problem containing three alternatives. Furthermore, I shall focus on the probability that the first alternative is chosen.[15] From the earlier discussion, $y_1 = 1$ if $y_1^* > y_2^*$ and $y_1^* > y_3^*$. Substituting $y_m^* = v_m + \epsilon_m$ yields the condition that $v_1 + \epsilon_1 > v_2 + \epsilon_2$ and $v_1 + \epsilon_1 > v_3 + \epsilon_3$. We can reformulate this condition by rearranging the terms: $\epsilon_2 - \epsilon_1 < v_1 - v_2$ and $\epsilon_3 - \epsilon_1 < v_1 - v_3$. In probabilistic terms,

$$\Pr(y_1 = 1) \;=\; \Pr(\epsilon_2 - \epsilon_1 < v_1 - v_2 \cap \epsilon_3 - \epsilon_1 < v_1 - v_3)$$

To simplify notation, let $\eta_{km} = \epsilon_k - \epsilon_m$ and let $v_{mk} = v_m - v_k$, then

$$\Pr(y_1 = 1) \;=\; \Pr(\eta_{21} < v_{12} \cap \eta_{31} < v_{13})$$

Evaluation of this last expression requires the introduction of the normality assumption. In addition, we need to invoke some basic statistical results about linear composites, since η_{km} is a linear function of two stochastic variables. Specifically, $\mathcal{E}[\eta_{km}] = 0$, $V[\eta_{km}] = \sigma_m^2 + \sigma_k^2 - 2\sigma_{mk}$, and $\mathcal{E}[\eta_{km}, \eta_{lm}] = \sigma_m^2 + \sigma_{kl} - \sigma_{mk} - \sigma_{ml}$ (for $m \neq l$).[16] Further, when the ϵs follow a multivariate normal distribution, then the ηs also follow a multivariate normal distribution.

With these results in place, we now know that η_{21} and η_{31} follow a bivariate normal distribution with covariance matrix

$$\mathbf{\Omega}_1 \;=\; \begin{pmatrix} \sigma_1^2 + \sigma_2^2 - 2\sigma_{12} & \sigma_1^2 + \sigma_{23} - \sigma_{21} - \sigma_{31} \\ \sigma_1^2 + \sigma_{23} - \sigma_{21} - \sigma_{31} & \sigma_1^2 + \sigma_3^2 - 2\sigma_{13} \end{pmatrix}$$

Let us refer to this distribution as $f(\eta_{21}, \eta_{31})$. Then it follows that the probability of choosing alternative 1 is given by

$$\pi_1 \;=\; \int_{-\infty}^{v_{12}} \int_{-\infty}^{v_{13}} f(\eta_{21}, \eta_{31}) d\eta_{21} d\eta_{31}$$

[15] As before, individual subscripts are suppressed in the formulas for the sake of simplicity.

[16] These results should look familiar, but they are also easily derived. First, $\mathcal{E}[\eta_{km}] = \mathcal{E}[\epsilon_k - \epsilon_m] = \mathcal{E}[\epsilon_k] - \mathcal{E}[\epsilon_m] = 0 - 0 = 0$. Second, $V[\eta_{km}] = \mathcal{E}[(\epsilon_k - \epsilon_m)^2] = \mathcal{E}[\epsilon_k^2 - 2\epsilon_k\epsilon_m + \epsilon_m^2] = \mathcal{E}[\epsilon_k^2] - 2\mathcal{E}[\epsilon_k\epsilon_m] + \mathcal{E}[\epsilon_m^2] = \sigma_k^2 - 2\sigma_{mk} + \sigma_m^2$. Finally, $\mathcal{E}[\eta_{km}, \eta_{lm}] = \mathcal{E}[(\epsilon_k - \epsilon_m)(\epsilon_l - \epsilon_m)] = \mathcal{E}[\epsilon_k\epsilon_l - \epsilon_k\epsilon_m - \epsilon l\epsilon_m + \epsilon_m^2] = \mathcal{E}[\epsilon_k\epsilon_l] - \mathcal{E}[\epsilon_k\epsilon_m] - \mathcal{E}[\epsilon l\epsilon_m] + \mathcal{E}[\epsilon_m^2] = \sigma_{kl} - \sigma_{km} - \sigma_{lm} + \sigma_m^2$.

These results imply the following general formula for the choice probabilities of the multinomial probit model:

$$\pi_m = \int_{-\infty}^{v_{m1}} \int_{-\infty}^{v_{m2}} \cdots \int_{-\infty}^{v_{mM}} \phi_{M-1}(\eta_{1m}, \eta_{2m}, \cdots \eta_{Mm}) d\eta_{1m} d\eta_{2m} \cdots d\eta_{Mm} \tag{11.7}$$

Note that if there are M alternatives, this is a $M-1$-variate integral. The function ϕ_{M-1} is a $M-1$-variate normal distribution. The covariance matrix of this distribution has diagonal elements of the form $\sigma_m^2 + \sigma_k^2 - 2\sigma_{mk}$ (for $k = 1, 2, \cdots M$ and $k \neq m$) and off-diagonal elements of the form $\sigma_m^2 + \sigma_{kl} - \sigma_{mk} - \sigma_{ml}$ (for $k, l = 1, 2, \cdots M$ and $k \neq m$, $l \neq m$, and $k \neq l$).

The multinomial probit model avoids the IIA assumption because the ratio of two choice probabilities depends on all of the alternatives in the choice set. Specifically,

$$\frac{\pi_m}{\pi_k} = \frac{\int_{-\infty}^{v_{m1}} \int_{-\infty}^{v_{m2}} \cdots \int_{-\infty}^{v_{mM}} \phi_{M-1}(\eta_{1m}, \eta_{2m}, \cdots \eta_{Mm}) d\eta_{1m} d\eta_{2m} \cdots d\eta_{Mm}}{\int_{-\infty}^{v_{k1}} \int_{-\infty}^{v_{k2}} \cdots \int_{-\infty}^{v_{kM}} \phi_{M-1}(\eta_{1k}, \eta_{2k}, \cdots \eta_{Mk}) d\eta_{1k} d\eta_{2k} \cdots d\eta_{Mk}}$$

We see that both the numerator and the denominator involve the remaining alternatives. Thus, removing one of these alternatives will impact the ratio π_m/π_k.

Modeling the Choice Probabilities

The development of the multinomial probit model so far does not consider any covariates. The manner in which covariates are handled is to specify v_m in terms of a linear function of the predictors:

$$v_{im} = \boldsymbol{x}_{im}\boldsymbol{\beta} + \boldsymbol{z}_i\boldsymbol{\gamma}_m$$

where the subscript i denotes a particular decision maker. In terms of v_{mk} this implies

$$\begin{aligned} v_{imk} &= v_{im} - v_{ik} \\ &= (\boldsymbol{x}_{im} - \boldsymbol{x}_{ik})\boldsymbol{\beta} + \boldsymbol{z}_i(\boldsymbol{\gamma}_m - \boldsymbol{\gamma}_k) \end{aligned} \tag{11.8}$$

A variety of more specific models can be derived from this general specification, including models that only consider attributes of the alternatives or models that only consider attributes of the decision maker.

11.3.2 Identification

As described by (11.7)-(11.8), the multinomial probit model is unidentified. That is, multiple values of $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}_m$ are consistent with the same choice probabilities π_{im}. The identification problem has two roots. First, for the choice probabilities all that matters are utility comparisons (i.e. is y_m^* greater than y_k^*?) If we try to say something about the actual utility level, as in (11.8) then there is simply not enough information in the choice probabilities to do this. Second, as we saw with binary choice models, the scale of utility is arbitrary. This means that the scale can be changed arbitrarily, resulting in a different set of parameter estimates.

The identification problem is resolved by imposing a number of identifying constraints. First, we focus on utility differences. This means that we draw comparisons to a baseline category, which is usually the first alternative. Thus, we set $\boldsymbol{\gamma}_1 = \mathbf{0}$ for the baseline category. In addition, ϵ_{i1} is assumed to be uncorrelated with the other errors and to have a variance of 1. Second, we fix the utility scale by setting Set $V[\epsilon_{im}] = 1$ for one of the non-baseline alternatives (typically $m = 2$).[17] Thus, in total, $M-2$ variances are estimated and $M(M-3)/2+1$ correlations. Another way to think of this is that $\boldsymbol{\Sigma}$ contains at most $M(M-1)/2-1$ identifiable parameters.

Keane (1992) warns that these are the absolute minimum restrictions. They are necessary but not always sufficient to ensure identification. In practice, it happens frequently that one has to impose additional restrictions. One can do so either by constraining the effects of predictors or by imposing more structure onto $\boldsymbol{\Sigma}$. In terms of the latter approach, a couple of possibilities suggest themselves. One is to impose a homoskedastic error structure, whereby $\sigma_m^2 = \sigma^2$ for all m. Another approach is to assume an exchangeable correlation structure, which means that all of the correlation parameters are assumed to be equal ($\rho_{mk} = \rho \; \forall \; m \neq q$). It is also impossible to impose other restrictions. For example, one can assume partial or complete independence among the alternatives. With partial independence, the errors of some alternatives are correlated while the remaining errors are treated as independent. With complete independence, all of the errors are treated as independent. Of course, if we combine independence with homoskedasticity, then the multinomial probit model is similar to the hybrid logit model described in the previous section, the only difference being a

[17] Alternatively, we can set $V[\eta_{ikm}] = 1$ (typically $k = 2$, $m = 1$). This is the approach that Train (2003) uses.

different distributional assumption.

To illustrate the minimum identifying assumptions, let us consider once more a choice problem with three alternatives. We treat the 1st alternative as the baseline. Thus, $\boldsymbol{\gamma}_1 = \mathbf{0}$, $\sigma_1^2 = 1$, and $\sigma_{1k} = 0$ (for $k \neq 1$). Further, we set $\sigma_2^2 = 1$. With these constraints in place,

$$\pi_{i1} = \int_{-\infty}^{v_{i12}} \int_{-\infty}^{v_{i13}} \phi_2(\eta_{i21}, \eta_{i31}) d\eta_{i21} d\eta_{i13}$$

where $v_{i12} = (\boldsymbol{x}_{i1} - \boldsymbol{x}_{i2})\boldsymbol{\beta} - \boldsymbol{z}_i\boldsymbol{\gamma}_2$, $v_{i13} = (\boldsymbol{x}_{i1} - \boldsymbol{x}_{i3})\boldsymbol{\beta} - \boldsymbol{z}_i\boldsymbol{\gamma}_3$, and ϕ_2 is the bivariate normal PDF with covariance matrix

$$\boldsymbol{\Sigma}_1 = \begin{bmatrix} 2 & \\ 1 + \sigma_{23} & 1 + \sigma_3^2 \end{bmatrix}$$

For the second alternative, we have

$$\pi_{i2} = \int_{-\infty}^{v_{i21}} \int_{-\infty}^{v_{i23}} \phi_2(\eta_{i12}, \eta_{i32}) d\eta_{i12} d\eta_{i32}$$

where $v_{i21} = (\boldsymbol{x}_{i2} - \boldsymbol{x}_{i1})\boldsymbol{\beta} + \boldsymbol{z}_i\boldsymbol{\gamma}_2$, $v_{i23} = (\boldsymbol{x}_{i2} - \boldsymbol{x}_{i3})\boldsymbol{\beta} + \boldsymbol{z}_i(\boldsymbol{\gamma}_2 - \boldsymbol{\gamma}_3)$, and ϕ_2 is the bivariate normal PDF with covariance matrix

$$\boldsymbol{\Sigma}_2 = \begin{bmatrix} 2 & \\ 1 - \sigma_{23} & 1 + \sigma_3^2 - 2\sigma_{23} \end{bmatrix}$$

Finally,

$$\pi_{i3} = \int_{-\infty}^{v_{i31}} \int_{-\infty}^{v_{i32}} \phi_2(\eta_{i13}, \eta_{i23}) d\eta_{i13} d\eta_{i23}$$

where $v_{i31} = (\boldsymbol{x}_{i3} - \boldsymbol{x}_{i1})\boldsymbol{\beta} + \boldsymbol{z}_i\boldsymbol{\gamma}_3$, $v_{i32} = (\boldsymbol{x}_{i3} - \boldsymbol{x}_{i2})\boldsymbol{\beta} + \boldsymbol{z}_i(\boldsymbol{\gamma}_3 - \boldsymbol{\gamma}_2)$, and ϕ_2 is the bivariate normal PDF with covariance matrix

$$\boldsymbol{\Sigma}_3 = \begin{bmatrix} 1 + \sigma_3^2 & \\ \sigma_3^2 - \sigma_{23} & 1 + \sigma_3^2 - 2\sigma_{23} \end{bmatrix}$$

11.3.3 Estimation

Estimation of the multinomial probit model can be quite complicated. The reason is that the choice probabilities and therefore the log-likelihood function involves multiple integrals. The evaluation of such integrals remains a computational challenge and until recently many believed that the multinomial probit model was merely of theoretical interest (see e.g. Maddala 1983). Computational advances in recent years, however, have made estimation feasible, although it remains complex.

Maximum Likelihood Estimation

The likelihood function for the multinomial probit model is given by

$$\mathcal{L} \;=\; \prod_i \prod_m \pi_{im}^{y_{im}}$$

The log-likelihood function is

$$\ell \;=\; \sum_i \sum_m y_{im} \ln(\pi_{im})$$

This looks innocent enough; indeed, this is exactly the same as what we had for the conditional/multinomial logit models. Once we substitute for π_{im}, however, things look considerably nastier:

$$\ell \;=\; \sum_i \sum_m y_{im} \times \ln\left[\int_{-\infty}^{v_{m1}} \int_{-\infty}^{v_{m2}} \cdots \int_{-\infty}^{v_{mM}} \phi_{M-1}(\eta_{1m}, \eta_{2m}, \cdots \eta_{Mm}) d\eta_{1m} d\eta_{2m} \cdots d\eta_{Mm}\right]$$

If the number of alternatives is three, then it turns out that optimization of this log-likelihood is not too complicated. In this case, we need only to evaluate a double integral. Moreover, the gradient involves only a single integral. Since optimization occurs in terms of the gradient, MLE thus requires that we evaluate a single integral and this generally does not pose too many difficulties if the model is properly identified.

Things become considerably hairier, however, when the number of alternatives exceeds three. In this case, the gradients involve multiple integrals. These remain difficult to evaluate even with the computational power that is available nowadays. In general, the log-likelihood can be evaluated only if the integrals can be approximated or bypassed altogether. There are several ways to accomplish this, but the most common approach is maximum simulated likelihood estimation.

Maximum Simulated Likelihood Estimation

The intuition behind maximum *simulated* likelihood estimation (MSLE) is that estimation of the multinomial probit model would be straightforward if we could use simulated values of the π_{im} in the log-likelihood function

(instead of having to derive those values through the evaluation of complex integrals). Lerman and Manski (1981) were the first to propose this estimation procedure. The general idea is to perform model estimation in two steps: (1) simulate π_{im} and then (2) maximize

$$\ell \;=\; \sum_i \sum_m y_{im} \ln(\tilde{\pi}_{im})$$

where $\tilde{\pi}_{im}$ denotes the simulated choice probability for alternative m in individual i.

The trick is, of course, to obtain values of $\tilde{\pi}_{im}$ that perform well. This problem was solved in the 1990s when Geweke (1991), Keane (1990, 1994), and Hajivassiliou and McFadden (1998; Hajivassiliou, McFadden, and Ruud 1996) independently developed similar methods for simulating multivariate normal probabilities. These methods are now commonly referred to under the name of Geweke-Hajivassiliou-Keane or GHK estimator. The GHK estimator is the most widely used MSLE method because it is more reliable and more accurate than alternative approaches (see Hajivassiliou, McFadden, and Ruud 1996).

The details of the GHK estimator are complex—they are described in Section 11.3.6. Basically, the estimation procedure starts by using Cholesky decomposition to replace ϵ with e, a set of uncorrelated standard normal errors. Next, the es are drawn from an appropriate truncated distribution. Then simulated choice probabilities are obtained in a recursive fashion. The whole process is repeated a number of times and the results are then averaged and entered into the log-likelihood function. It can be demonstrated that this procedure generates consistent and asymptotically efficient estimators. The GHK estimator is now implemented in a number of statistical packages, including Stata (as of version 9.0).

11.3.4 Interpretation

Marginal Effects and Elasticities

Discrete Change in Predicted Probabilities

11.3.5 Example: Vote Choice in the 1994 Dutch Parliamentary Elections

11.3.6 Maximum Simulated Likelihood Estimation*

How does the GHK-estimator simulate the probabilities π_{im}?[18] Unfortunately, to answer this question we need a fairly technical discussion. As always, feel free to skip this material if the math is too overwhelming. In this case, all you need to remember is that it is possible to accurately simulate the choice probabilities of the multinomial probit model so that MSLE estimation of that model is feasible.

Let us first consider a choice model with three alternatives, where we focus on the probability of choosing the 1st alternative. As per our earlier discussion, alternative 1 is chosen if $y_{i1}^* > y_{i2}^*$ and if $y_{i1}^* > y_{i3}^*$. Here, $y_{im}^* = v_{im} + \epsilon_{im}$. Taking utility differences with respect to the 1st alternative we get

$$\begin{aligned} y_{ik}^* - y_{i1}^* &= (v_{ik} - v_{i1}) + (\epsilon_{ik} - \epsilon_{i1} \\ &= -v_{i1k} + \eta_{ik1} \end{aligned}$$

for $k = 2, 3$. The vector η_{ik1} has a variance-covariance matrix of

$$\mathbf{\Sigma}_1 = \begin{bmatrix} \sigma_1^2 + \sigma_2^2 - 2\sigma_{12} & \sigma_1^2 + \sigma_{23} - \sigma_{12} - \sigma_{13} \\ \sigma_1^2 + \sigma_{23} - \sigma_{12} - \sigma_{13} & \sigma_1^2 + \sigma_3^2 - 2\sigma_{13} \end{bmatrix}$$

We begin by performing a Cholesky decomposition of $\mathbf{\Sigma}_1$. This produces a lower triangular matrix, $\mathbf{L}_1$, such that $\mathbf{L}_1\mathbf{L}'_1 = \mathbf{\Sigma}_1$. More specifically,

$$\mathbf{L}_1 = \begin{bmatrix} c_{aa} & 0 \\ c_{ab} & c_{bb} \end{bmatrix}$$

where $c_{aa} = \sqrt{\sigma_1^2 + \sigma_2^2 - 2\sigma_{12}}$, $c_{ab} = (\sigma_1^2 + \sigma_{23} - \sigma_{12} - \sigma_{13})/\sqrt{\sigma_1^2 + \sigma_2^2 - 2\sigma_{12}}$, and $c_{bb}^2 = \sigma_1^2 + \sigma_3^2 - 2\sigma_{13} - ((\sigma_1^2 + \sigma_{23} - \sigma_{12} - \sigma_{13})^2/(\sigma_1^2 + \sigma_2^2 - 2\sigma_{12}))$.

[18] For a slightly different discussion of the GHK-approach, as well as a political science application, see Lawrence (1997). The discussion here draws heavily from Train (2003).

Using the Cholesky decomposition, the original correlated error terms η can be expressed in terms of a series of uncorrelated standard normal deviates, e. That is

$$\begin{aligned} \boldsymbol{\eta}_{i1} &= \mathbf{L}_1 \boldsymbol{e}_{i1} \\ \begin{bmatrix} \eta_{i21} \\ \eta_{i31} \end{bmatrix} &= \begin{bmatrix} c_{aa} & 0 \\ c_{ab} & c_{bb} \end{bmatrix} \begin{bmatrix} e_{i21} \\ e_{i31} \end{bmatrix} \end{aligned}$$

Chapter 12

Event Count Models

12.1 The Poisson Regression Model

12.1.1 Derivation

12.1.2 Estimation

12.1.3 Interpretation

12.2 The Negative Binomial Model

12.2.1 The Problem of Over-Dispersion

What Over-Dispersion Means

The Cameron-Trivedi Test

12.2.2 Derivation

12.2.3 Estimation

12.2.4 Interpretation

12.2.5 An Alternative Test of Over-Dispersion

12.3 The Generalized Event Count Model*

12.4 Hurdle and Zero-Inflated Count Models

12.4.1 Hurdle Models [312]

Derivation

Estimation

Interpretation

12.4.2 Zero-Inflated Models

Derivation

Estimation

Chapter 13

Event Duration Models

Chapter 14

Multi-Level Models

Chapter 15

Probability Distributions

The following probability distribution underlie the applications that have been discussed in this report. For each distribution, the range of the random variable, the parameter(s), probability function, distribution function (where available), mean, variance, likelihood and log-likelihood functions, and special cases are noted.

15.1 Bernoulli Distribution

Range	$y \in \{0, 1\}$
Parameter	Probability parameter: $0 \leq \pi \leq 1$
Probability function	$f(y) = \pi^y (1-\pi)^{1-y}$
Distribution function	$F(y) = (1-\pi)^{1-y}$
Mean	π
Variance	$\pi(1-\pi)$
Likelihood function	$\mathcal{L} = \pi^{\sum_i y_i}(1-\pi)^{n-\sum_i y_i}$
Log-likelihood function	$\ell = \sum_i y_i \ln(\pi) + (n - \sum_i y_i)\ln(1-\pi)$
Applications	Linear probability model

15.2 Bivariate Normal Distribution

Range	$-\infty < y_1 < \infty$
	$-\infty < y_2 < \infty$
Parameters	Location parameters: μ_1, μ_2
	Scale parameters: σ_1, σ_2

	Correlation parameter: ρ
Probability function	$f(y_1, y_2) = \frac{1}{2\pi\sigma_1\sigma_2\sqrt{1-\rho^2}} \exp\left\{-\frac{\epsilon_1^2+\epsilon_2^2-2\rho\epsilon_1\epsilon_2}{2(1-\rho^2)}\right\}$
	where:
	$\epsilon_1 = \frac{y_1-\mu_1}{\sigma_1}$
	$\epsilon_2 = \frac{y_2-\mu_2}{\sigma_2}$
Mean	μ_1, μ_2
Variance	σ_1^2, σ_2^2
Covariance	$\sigma_{12} = \rho\sigma_1\sigma_2$
Likelihood function	$\mathcal{L} = [4\pi^2\sigma_1^2\sigma_2^2(1-\rho^2)]^{-.5n} \times$
	$\exp\left\{-\frac{1}{2(1-\rho^2)}\sum_i(\epsilon_{1i}^2+\epsilon_{2i}^2-2\rho\epsilon_{1i}\epsilon_{2i})\right\}$
Log-likelihood function	$\ell = -.5n\ln[4\pi^2\sigma_1^2\sigma_2^2(1-\rho^2)] -$
	$\frac{1}{2(1-\rho^2)}\sum_i(\epsilon_{1i}^2+\epsilon_{2i}^2-2\rho\epsilon_{1i}\epsilon_{2i})$
Applications	Heckman sample selection model

15.3 Burr-II Distribution

Range	$-\infty < y < \infty$
Parameter	Shape parameter: $\alpha > 0$
Probability function	$f(y) = \frac{\alpha[1+\exp(-y)]^{-\alpha}}{1+\exp(y)}$
Distribution function	$F(y) = [1+\exp(-y)]^{-\alpha}$
Likelihood function	$\mathcal{L} = \frac{\alpha^n\prod_i[1+\exp(-y_i)]^{-\alpha}}{\prod_i[1+\exp(y_i)]}$
Log-likelihood function	$\ell = n\ln(\alpha) - \alpha\sum_i\ln[1+\exp(y_i)]$
	$-\sum_i\ln[1+\exp(y_i)]$

15.4 Exponential Distribution

Range	$0 \le y < \infty$
Parameter	Scale parameter: $\beta > 0$
Probability function	$f(y) = \frac{1}{\beta}\exp\left(-\frac{y}{\beta}\right)$
	$f(y) = \lambda\exp(-\lambda y)$
Distribution function	$F(y) = 1-\exp\left(-\frac{y}{\beta}\right)$
	$F(y) = 1-\exp(-\lambda y)$
Mean	β

Variance	β^2
Likelihood function	$\mathcal{L} = \left(\frac{1}{\beta}\right)^n \exp\left(-\frac{\sum_i y_i}{\beta}\right)$
Log-likelihood function	$\ell = -n \ln \beta - \beta^{-1} \sum_i y_i$
Applications	Event duration models

15.5 Extreme Value (Gumbel) Distribution

Range	$-\infty < y < \infty$
Parameters	Location parameter: α Scale parameter: $\beta > 0$
Probability function	$f(y) = \frac{1}{\beta} \exp\left(-\frac{y-\alpha}{\beta}\right) \exp\left\{-\exp\left(-\frac{y-\alpha}{\beta}\right)\right\}$
Distribution function	$F(y) = \exp\left\{-\exp\left(-\frac{y-\alpha}{\beta}\right)\right\}$
Mean	$\alpha - .57722\beta$
Variance	$\frac{\beta^2\pi^2}{6}$
Likelihood function	$\mathcal{L} = \beta^{-n} \exp\left(\sum_i -\frac{y_i-\alpha}{\beta}\right) \times$ $\exp\left\{-\exp\left(\sum_i -\frac{y_i-\alpha}{\beta}\right)\right\}$
Log-likelihood function	$\ell = -n \ln \beta - \sum_i -\frac{y_i-\alpha}{\beta} - \exp\left(\sum_i -\frac{y_i-\alpha}{\beta}\right)$
Special cases:	
Standard Gumbel	$\alpha = 0$, $\beta = 1$
Applications	Binary regression models

15.6 Gamma Distribution

15.7 Logistic Distribution

Range	$-\infty < y < \infty$
Parameters	Location parameter: α Scale parameter: $\beta > 0$
Probability function	$f(y) = \frac{\exp\left(\frac{y-\alpha}{\beta}\right)}{\beta\left\{1+\exp\left(\frac{y-\alpha}{\beta}\right)\right\}^2}$
Distribution function	$F(y) = \frac{1}{1+\exp\left(-\frac{y-\alpha}{\beta}\right)}$
Mean	α
Variance	$\frac{\pi^2\beta^2}{3}$

Likelihood function	$\mathcal{L} = \frac{\exp\left\{\sum_i \frac{y_i-\alpha}{\beta}\right\}}{\beta^n \prod_i \left(1+\exp\left\{\frac{y_i-\alpha}{\beta}\right\}\right)^2}$
Log-likelihood function	$\ell = -n\ln\beta - 2\sum_i \ln\left(1+\exp\left\{\frac{y_i-\alpha}{\beta}\right\}\right) + \sum_i \frac{y_i-\alpha}{\beta}$
Special cases:	
Standard logistic	$\alpha = 0$, $\beta = 1$
Standardized logistic	$\alpha = 0$, $\beta = \frac{\sqrt{3}}{\pi}$
Applications	Logit model

15.8 Log-Normal Distribution

15.9 Normal Distribution

Range	$-\infty < y < \infty$
Parameters	Location parameter: μ
	Scale parameter: $\sigma > 0$
Probability function	$f(y) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{1}{2\sigma^2}\left(y_i-\mu\right)^2\right\}$
Mean	μ
Variance	σ^2
Likelihood function	$\mathcal{L} = \left(2\pi\sigma^2\right)^{-.5n} \exp\left\{-\frac{1}{2\sigma^2}\sum_i \left(y_i-\mu\right)^2\right\}$
Log-likelihood function	$\ell = -.5n\ln(2\pi) - .5n\ln(\sigma^2) - \frac{1}{2\sigma^2}\sum_i \left(y_i-\mu\right)^2$
Special cases:	
Standard normal	$\mu = 0$, $\sigma = 1$
Applications	Regression model; probit model

15.10 Negative Binomial Distribution

15.11 Poisson Distribution

Range	$y = 0, 1, 2\cdots$
Parameters	Location parameter: μ
Probability function	$f(y) = \frac{\mu^y \exp(-\mu)}{t!}$
Distribution function	$F(y) = \sum_{i=0}^{y} \frac{\mu^i \exp(-\mu)}{i!}$
Mean	μ
Variance	μ

Likelihood function	$\mathcal{L} = \frac{\mu^{\sum_i y_i} \exp(-n\mu)}{\prod_i y_i!}$
Log-likelihood function	$\ell = \ln(\mu) \sum_i y_i - n\mu - \sum_i \ln(y_i!)$
Applications	Event count models

15.12 Weibull Distribution