FIRST-ORDER DYNAMICS OF THE LOGIT-MLE PROCEDURE

By

SCOTT R. KEGLER

Bachelor of Science
University of Minnesota
Minneapolis, Minnesota
1979

Bachelor of Arts
University of Minnesota
Minneapolis, Minnesota
1990

Master of Operations Research
North Carolina State University
Raleigh, North Carolina
1992

Master of Science
North Carolina State University
Raleigh, North Carolina
1995

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
DOCTOR OF PHILOSOPHY
May, 1999

FIRST-ORDER DYNAMICS OF THE

LOGIT-MLE PROCEDURE

Dissertation Approved:

Barry Kurt Moser

Dissertation Advisor

Melinda H. McC

Mk E Payt

J Chandler

Wayne B. Powell

Dean of the Graduate College

PREFACE

Numerous statistical procedures for conducting sequential experiments have been developed over the past five decades. Many such procedures have been analyzed more or less completely in terms of asymptotic performance. However, some of the more recent model-based procedures involving principles of maximum likelihood have proven somewhat less tractable to such analyses. The focus of this dissertation is the asymptotic behavior of the Logit-MLE procedure for binary response data first proposed by Wu (1985). Under weak assumptions it is shown that a first-order approximation of the dynamics of this procedure produces consistent sequential estimates. Subsequent simulation shows that the accuracy of the approximation improves as the sequential trial count increases, thus giving weight to the argument that the Logit-MLE procedure *per se* also produces consistent estimates.

Each of my committee members contributed to this research effort in his or her own unique way. Foremost thanks go to my advisor, Dr. Barry Moser, for providing me with a very stimulating problem to investigate, great assistance with the literature in this area, and careful guidance throughout the process. My committee members from the Department of Statistics, Dr. Moser, Dr. Mark Payton, and Dr. Melinda McCann, all contributed to an outstanding environment, both inside the classroom and out, in which to carry out this research. Thanks are also due to Dr. John Chandler of the Department of Computer Science for serving as my outside committee member, and for providing advice (and source code) for my simulations.

Finally, I would like to thank my wife, Dr. Michelle Kegler, for her patience and support throughout my studies.

TABLE OF CONTENTS

LIST OF FIGURES AND TABLES

NOMENCLATURE / LIST OF SYMBOLS

Term / Symbol	Definition
a_n	multiplier used in the stochastic approximation iteration form $x_{n+1} = x_n - a_n \cdot (y_n - p^0)$
β	scale parameter in the two-parameter logit function $\psi(x;\mu,\beta)$
$\hat{\beta}_n$	maximum likelihood estimate of the parameter β after n Logit-MLE trials
EQRP	estimated quantal response problem
$\mathbf{H}_n$	Hessian matrix for the log-likelihood function $\log \mathcal{L}_n$ after n Logit-MLE trials
$\mathcal{L}_n$	likelihood function for the parameters μ and β after n Logit-MLE trials
μ	location parameter in the two-parameter logit function $\psi(x;\mu,\beta)$
$\hat{\mu}_n$	maximum likelihood estimate of the parameter μ after n Logit-MLE trials
n	trial index in a sequential experiment
n_E	terminal trial index of the exact phase of the *approximating* Logit-MLE procedure
n_I	terminal trial index of the startup phase in the Logit-MLE procedure
n_T	terminal trial index of the sequential phase in the Logit-MLE procedure
p^0	target probability of positive response in the EQRP
$\psi(x;\mu,\beta)$	two-parameter logit function with location parameter μ and scale parameter β
R(x)	true probability of positive response at control variable level x in the EQRP
w_n	estimate of x^0 on trial n in the *approximating* Logit-MLE procedure
x	control variable in the quantal response function R(x)
x^0	target control variable setting corresponding to probability p^0 in the EQRP
x_n	estimate of x^0 on trial n of a sequential experiment
y_n	binary response on trial n of a sequential experiment

CHAPTER 1

SEQUENTIAL ESTIMATION

1.1 The Estimated Quantal Response Problem

Abstractly, the object of study is some phenomenon involving a binary response. Without loss of generality the response levels can be labeled *success* and *failure*. It is assumed that the probability p of success on any given trial is influenced in a systematic way by the level of some variable x under control of the experimenter. The functional relationship between p and x, although unknown, will be assumed throughout the treatment to be continuous and strictly increasing. Additional assumptions may be imposed depending on the context. The underlying relationship will be denoted by the function R(x):

$$R(x) \equiv \Pr\{ \text{success} \mid x \} .$$

The experimental objective is to estimate the unique level $x = x^0$ at which the probability of success assumes a particular value $0 < p^0 < 1$. Figure 1.1 below illustrates a typical situation:

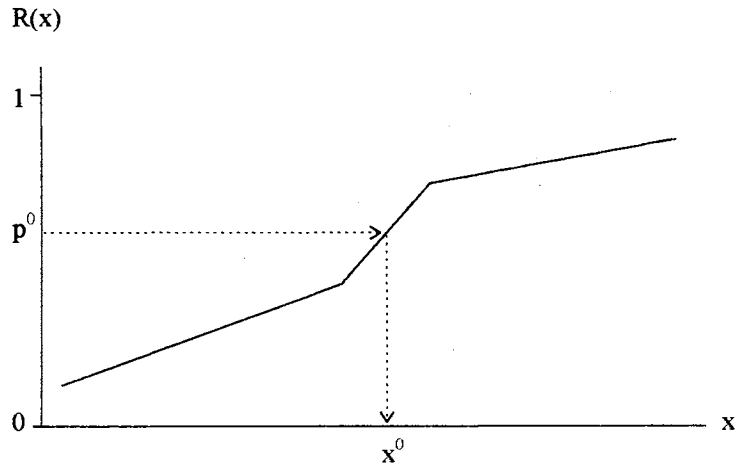

Figure 1.1 The Estimation Problem

The curve characterizing the underlying response relationship is sometimes called the *quantal response*

curve. The problem described will thus be referred to as the *estimated quantal response problem* (EQRP). This problem is generally considered part of the much broader domain of *stochastic approximation*.

1.2 Sequential Estimation Methods

This section surveys important developments applicable to the EQRP defined in §1.1. The focus is confined to *sequential* estimation methods. The simplest sequential estimation method involves making a single observation at some level of the control variable x believed to be close to the target level x^0. Based on the observed response (and possibly prior observations) a new level of x is chosen and the cycle repeats. The process terminates when some stopping rule (often a given sample size) is satisfied. More complex designs may involve taking multiple observations at x on each iteration or taking observations at several levels of x simultaneously. The essential feature is the sequential manner in which the data are collected.

Many of the results to be described below are in fact applicable to a class of problems much broader than the EQRP framework. Although the conditions required to support the general application of a particular theorem may be numerous, many such conditions are automatically satisfied in the more limited context of the EQRP. Only those conditions not implied by the ones already set forth above are specified when referencing any particular theorem. On occasion this may seem to diminish the accomplishments of the researchers mentioned, but when considered in a more general stochastic approximation framework the progression of results is more marked.

1.2.1 Consistency Results

Robbins and Monro (1951) are generally credited with initiating the main line of research concerning methods for treating various stochastic approximation problems. Their paper presented an estimation method applicable to the EQRP as a special case. Their idea was to form a sequence of estimates of x^0 based on responses observed over successive trials. Letting x_n denote the setting of the control variable on trial n the corresponding response $y_n \sim \text{BIN}(1,R(x_n))$. In their analysis the quantal response function must

satisfy the additional condition that it is differentiable at x^0. The authors defined a positive sequence of real numbers $\{a_n\}$ to be a *sequence of type 1/n* if it satisfies:

$$\sum_{n=1}^{\infty} a_n^2 < \infty$$

$$\sum_{n=2}^{\infty} a_n / (a_1 + \ldots + a_{n-1}) = \infty . \qquad (1.2.1)$$

It can be shown that any sequence satisfying $c'/n \leq a_n \leq c''/n$ where $0 < c' \leq c''$ is such a sequence. A common example is the sequence $\{1/n\}$. Based on any sequence of type 1/n it was demonstrated that the sequence of estimates defined by:

$$x_{n+1} = x_n - a_n \cdot (y_n - p^0) \qquad (1.2.2)$$

with x_1 arbitrary will converge in probability to x^0 as $n \to \infty$. Iterations such as (1.2.2) as well as numerous variants have come to be known as *Robbins-Monro processes*. No further properties were attributed to such processes by Robbins and Monro in their original paper, but their work ushered in a very productive period during which their results were refined and extended by numerous researchers.

Applying the findings of Wolfowitz (1952) to the EQRP provides the same convergence result as that of Robbins and Monro with the differentiability condition on R(x) at x^0 removed. Moreover, the conditions (1.2.1) on the multiplier sequence $\{a_n\}$ can be replaced with the slightly simpler conditions:

$$\sum_{n=1}^{\infty} a_n^2 < \infty$$

$$\sum_{n=1}^{\infty} a_n = \infty . \qquad (1.2.3)$$

This class of sequences $\{a_n\}$ is at least as large as the class defined in (1.2.1). The conditions (1.2.3) on $\{a_n\}$ are the ones most commonly cited in subsequent literature.

Blum (1954) provided a stronger convergence result. Without assuming differentiability of R(x) at x^0 it was shown that a sequence $\{x_n\}$ defined as in (1.2.2) with $\{a_n\}$ satisfying (1.2.3) will converge to x^0 almost surely.

In the proof of each convergence result thus far the sequence $\{a_n\}$ is generally taken to be comprised of terms which do not depend on the random variables x and y. Robbins and Siegmund (1971) greatly relaxed this limitation. Their result shows that each multiplier a_n may be a nonnegative (measurable) function of (x_1,y_1) , ... , (x_{n-1},y_{n-1}). Provided that the resulting sequence $\{a_n\}$ satisfies conditions (1.2.3) then $x_n \rightarrow x^0$ almost surely. Although the values of x_n and y_n will be known at the time a_n is used in the iteration (1.2.2) they are excluded from its formulation.

1.2.2 Asymptotic Normality Results

Apart from the issue of convergence the other principal focus of research concerned the asymptotic distribution of iterates x_n of the form (1.2.2). Using differing approaches Chung (1954) and Sacks (1958) derived conditions under which the standardized estimate:

$$z_n \equiv \sqrt{n} \cdot (x_n - x^0) \tag{1.2.4}$$

is asymptotically normal with an asymptotic mean of zero and asymptotic variance to be given below. The assumptions required for asymptotic normality are somewhat more involved than those required for the convergence results. To fix ideas the result of Sacks (1958) will be considered here since it appeared latest and seems to be the most general. There are several conditions on R(x) not automatically satisfied in the EQRP as defined. It must be assumed that the response function can be expressed in the form:

$$R(x) = p^0 + \alpha_1 \cdot (x - x^0) + \delta(x,x^0) \tag{1.2.5}$$

where $\alpha_1 > 0$ and $\delta(x,x^0) = o(|x-x^0|)$ as $x \rightarrow x^0$. It must also be assumed that there exists a positive constant K such that:

$$|R(x) - p^0| \leq K \cdot |x - x^0| \qquad \forall\, x\,. \tag{1.2.6}$$

Note that condition (1.2.5) implies differentiability of R(x) at x^0 with $R'(x^0) = \alpha_1$. Consider any constant A such that:

$$A \cdot \alpha_1 > \tfrac{1}{2}\,. \tag{1.2.7}$$

If the sequence of multipliers $\{a_n\} = \{A/n\}$ then the standardized iterates z_n will be asymptotically normal with asymptotic mean zero and asymptotic variance:

$$\sigma_z^2 = A^2 \cdot p^0(1 - p^0) \,/\, (2A\alpha_1 - 1)\,. \tag{1.2.8}$$

This result has several practical implications. If a lower bound $\alpha_1' > 0$ on $R'(x^0)$ can be reasonably assumed then iterates x_n generated using $\{a_n\} = \{A/n\}$ with $A > 1/(2\alpha_1')$ will be asymptotically normally distributed with asymptotic mean x^0 and asymptotic variance σ_z^2/n where σ_z^2 is as given in (1.2.8). Since the form specified for $\{a_n\}$ makes it a sequence of type 1/n it also follows that x_n converges almost surely to x^0 due to results already presented.

1.2.3 Asymptotically Optimal Processes

The variance formula (1.2.8) stimulated additional research. Differentiation wrt A reveals that selecting $A = 1/\alpha_1 \equiv 1/R'(x^0)$ minimizes σ_z^2 at $p^0(1-p^0)/\alpha_1^2$ while simultaneously satisfying condition (1.2.7).

Therefore a sequence $\{x_n\}$ based on $\{a_n\} = \{ 1/(n\cdot\alpha_1) \}$ will possess the convergence and asymptotic normality properties already discussed along with minimum asymptotic variance among all sequences using multipliers of form $\{a_n\} = \{A/n\}$ with $A > 1/(2\alpha_1)$. A Robbins-Monro process possessing these properties is usually regarded as *asymptotically optimal*. The obvious limitation is that R(x) is unknown. This led to attempts to incorporate various estimates of $R'(x^0)$ into the sequential estimation process. One of the earliest such methods was the *extended Robbins-Monro process* proposed by Venter (1967). It is assumed that the response function can be expressed over some interval $x^0-\varepsilon < x < x^0+\varepsilon$ in the form:

$$R(x) = p^0 + \alpha_1\cdot(x - x^0) + \alpha_2\cdot(x - x^0)^2 + \ldots + \alpha_s\cdot(x - x^0)^s + \delta(x,x^0) \qquad (1.2.9)$$

where $\alpha_1 > 0$ and $\delta(x,x^0) = o(|x-x^0|^s)$ as $x \to x^0$. To achieve all of the important asymptotic results the index limit s in (1.2.9) must be an integer in $\{2,3,4,\ldots\} \cup \{\infty\}$. As will become evident below a large value of s is beneficial. Condition (1.2.9) again implies differentiability of R(x) at x^0 with $R'(x^0) = \alpha_1$. The proposed procedure also assumes that $R'(x^0) \in (a,b)$ where a and b are known positive constants. Given a current estimate x_n of x^0 the next iteration of the procedure involves a pair of experimental responses. The response y_n^{lower} is observed at the setting $x_n^{lower} = x_n - c_n$ and the response y_n^{upper} is observed at the setting $x_n^{upper} = x_n + c_n$ where $\{c_n\}$ is a sequence of gradually diminishing positive constants. Any sequence of the form $\{c_n\} = \{c/n^\gamma\}$ where $c > 0$ and $\gamma \in (¼,½)$ is suitable for purposes here, although the class of acceptable sequences is defined more broadly. An estimate of $R'(x^0)$ is then defined by:

$$B_n = (1/n) \sum_{k=1}^{n} (y_k^{upper} - y_k^{lower}) / (2\cdot c_k) \qquad (1.2.10)$$

and subsequently truncated to:

$$A_n = \max \{ a , \min\{B_n , b\} \} . \qquad (1.2.11)$$

A typical application of this slope estimate in forming the next estimate of x^0 is then:

$$x_{n+1} = x_n - (1/(n \cdot A_n)) \cdot ((y_n^{lower} + y_n^{upper})/2 - p^0) . \tag{1.2.12}$$

If $s \geq 2$ in (1.2.9) and:

$$s \cdot \alpha_1 > \gamma \cdot b \tag{1.2.13}$$

then $x_n \rightarrow x^0$ almost surely as $n \rightarrow \infty$ and $A_n \rightarrow \alpha_1 \equiv R'(x^0)$ almost surely as $n \rightarrow \infty$, and:

$$\sqrt{n} \cdot (x_n - x^0) \xrightarrow{d} N(0 , p^0(1-p^0)/(2\alpha_1^2)) . \tag{1.2.14}$$

The asymptotic variance given in (1.2.14) is one-fourth the minimum asymptotic variance previously given for Robbins-Monro processes involving multiplier sequences of the form $\{a_n\} = \{A/n\}$. This is attributable to the fact that two observations are made per iteration in the extended Robbins-Monro process. Hence the extended process is no less efficient and this was indeed mentioned by Venter as a defense against the criticism that it requires two observations per iteration.

Anbar (1978) proposed using a least squares regression estimate of $R'(x^0)$ based on just one experimental response per iteration. Under this approach condition (1.2.5) is assumed for R(x). Another assumption concerning R(x) bearing a similarity to (1.2.6) is the existence of constants $0 < K_1 < K_2 < \infty$ such that:

$$K_1 \cdot |x - x^0| < |R(x) - p^0| < K_2 \cdot |x - x^0| \qquad \forall\, x \neq x^0 . \tag{1.2.15}$$

Moreover, the control variable x must be restricted to some interval $[x_{min} , x_{max}]$ where x_{min} and x_{max} are assumed to satisfy:

$$0 \;<\; R(x_{min}) \;<\; R(x_{max}) \;<\; 1\,. \tag{1.2.16}$$

Given observations (x_1,y_1) , … , (x_n,y_n) the least squares estimator is then defined in the usual way:

$$B_n \;=\; \sum_{k=1}^{n} (x_k - \bar{x}_n)\cdot y_k \;/\; \sum_{k=1}^{n} (x_k - \bar{x}_n)^2 \qquad n \geq 2 \tag{1.2.17}$$

with B_0 and B_1 represented by arbitrary initial guesses. In an approach similar to that taken by Venter it is assumed that constants a and b satisfying $K_1 \leq a < \alpha_1 < b \leq K_2$ are known. These constants are used to truncate the slope estimate as necessary:

$$A_n \;=\; \max\{\, a\,,\, \min\{B_{n-1}\,,\, b\}\,\}\,. \tag{1.2.18}$$

Notably the truncation is lagged by one iteration. This is evidently due to the need to appeal to the convergence result of Robbins and Siegmund (1971) previously described in §1.2.1. The truncated slope estimate is cycled back into the iteration process via:

$$x_{n+1} \;=\; x_n \;-\; (1/(n\cdot A_n))\cdot(y_n - p^0)\,. \tag{1.2.19}$$

Provided that:

$$b \;<\; 2\cdot K_1 \tag{1.2.20}$$

then $x_n \rightarrow x^0$ almost surely as $n \rightarrow \infty$ and $B_n \rightarrow \alpha_1 \equiv R'(x^0)$ almost surely as $n \rightarrow \infty$, and:

$$\sqrt{n}\cdot(x_n - x^0) \;\xrightarrow{d}\; N(0\,,\, p^0(1-p^0)/\alpha_1^2)\,. \tag{1.2.21}$$

Thus Anbar's method also achieves the minimum asymptotic variance discussed previously. Note that here the untruncated slope estimator is strongly consistent as opposed to the truncated estimator in Venter's results. In order to reduce potentially excessive bias associated with early observations both Venter and Anbar discuss the possibility of also truncating the data sequence used in forming the slope estimates; once n is sufficiently large only the observations (x_m,y_m) , ... , (x_n,y_n) would be used. Only Anbar provides specific limitations on the extent of truncation permissible.

Procedures such as those specified by Venter and Anbar rely on multipliers a_n which are influenced by all or most of the data observed up to any given moment. Such procedures are often called *adaptive* Robbins-Monro processes. By comparison, a *nonadaptive* Robbins-Monro process based on a sequence $\{a_n\}$ fixed in advance selects each successive level of x based solely on the current level x_n, the response y_n observed at that level, and the preselected multiplier value a_n.

1.2.4 Model-Based Approaches

The adaptive approaches discussed in §1.2.3 make almost no assumptions about the form of the underlying response function R(x). They tend to concentrate on using local information about R(x) as opposed to possibly more global properties. Wu (1985) proposed an approach whereby a parametric model form suitable to the binary response situation serves as a working model of R(x). The stated purpose was to take better advantage of the information contained in the data, a goal regarded as especially important given the small sample sizes often accompanying sequential experiments. Much of Wu's presentation centers specifically on the two-parameter logit function $\psi(x;\mu,\beta)$ as a model for the underlying R(x):

$$\psi(x;\mu,\beta) \equiv e^{\mu+\beta x} / (1 + e^{\mu+\beta x}) . \qquad (1.2.22)$$

This function is S-shaped and strictly increasing in x when $\beta > 0$. The parametric model form $\psi(x;\mu,\beta)$ has been empirically shown to accommodate the mean response data for a wide variety of phenomena

involving binary responses. A typical graph of $\psi(x)$ with $\beta > 0$ appears in Figure 1.2:

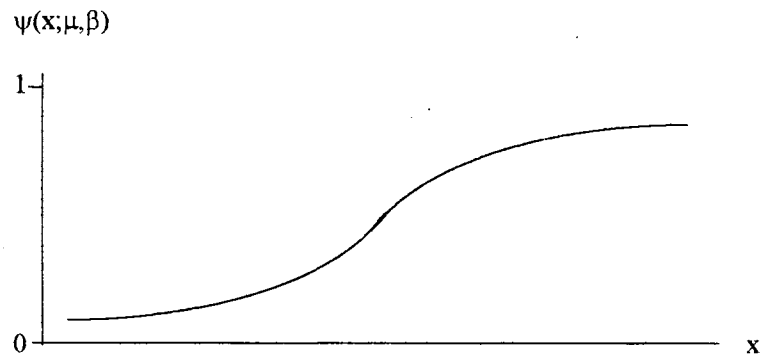

Figure 1.2 Two-parameter Logit Function

Wu's method involves calculating maximum likelihood estimates (MLEs) for the model parameters μ and β based on all of the data observed through the current iteration n. Let $\hat{\mu}_n$ and $\hat{\beta}_n$ denote the MLEs based on the data available after completion of n trials. These parameter estimates determine the fitted model $\hat{\psi}_n(x) \equiv \psi(x;\hat{\mu}_n,\hat{\beta}_n)$. The next level for the control variable x is then determined by solving:

$$\hat{\psi}_n(x) = p^0 \tag{1.2.23}$$

for $x = x_{n+1}$. A new response y_{n+1} is observed with $x = x_{n+1}$ and the cycle is repeated. Wu called this particular model-based approach the *Logit-MLE procedure*. For experiments involving small to medium sample sizes simulation studies suggest that such an approach is quite effective compared to other methods previously considered. However, only under quite strong assumptions was Wu able to demonstrate any asymptotic properties for this procedure. Treatment of the asymptotic behavior of the Logit-MLE procedure is the central focus of the research proposal to follow.

In all of the findings discussed thus far the control variable x is permitted to vary continuously. Other methods confine x to a discrete set $\{d_1,d_2,\ldots,d_N\}$ of *design points* fixed in advance of the experiment.

Shen and O'Quigley (1996) considered such an approach based on one-parameter models of R(x). Let $\varphi(x;\theta)$ represent such a model. As in the approach taken by Wu the MLE of the model parameter θ is reestimated as each additional data point is observed. Let $\hat{\theta}_n$ denote the MLE based on the data available after n trials. This parameter estimate determines the fitted model $\hat{\varphi}_n(x) \equiv \varphi(x;\hat{\theta}_n)$. The next setting for the control variable x is then determined by selecting the design point which minimizes the difference between the currently fitted model and the target probability level:

$$x_{n+1} = \arg\min_{x \in \{d_1,d_2,\ldots,d_N\}} |\hat{\varphi}_n(x) - p^0| \ . \tag{1.2.24}$$

In such a procedure the *target design point* x^T is the value in $\{d_1,d_2,\ldots,d_N\}$ that minimizes the difference between the true response function R(x) and the target probability level:

$$x^T = \arg\min_{x \in \{d_1,d_2,\ldots,d_N\}} |R(x) - p^0| \ . \tag{1.2.25}$$

Let θ^T designate the parameter value that causes the curve $\varphi(x;\theta^T)$ to pass through the point $(x^T,R(x^T))$. That is, θ^T is the parameter value that causes the model curve to agree with the true response curve at the target design point x^T. Under certain assumptions the authors obtained the following results which share the convergence property for $\{x_n\}$ cited earlier for procedures estimating x^0 over a continuous range:

$$x_n \to x^T \text{ almost surely as } n \to \infty \tag{1.2.26}$$

$$\hat{\theta}_n \to \theta^T \text{ almost surely as } n \to \infty \ . \tag{1.2.27}$$

Moreover, the MLE $\hat{\theta}_n$ is asymptotically normal with a known asymptotic variance.

Moser and Faries (1996) investigated a variant of Wu's Logit-MLE procedure whereby simultaneous observations are taken on each iteration at the control variable settings $x_{n+1,1}$ and $x_{n+1,2}$ satisfying:

$$\hat{\psi}_n(x_{n+1,1}) = p^1$$
$$\hat{\psi}_n(x_{n+1,2}) = p^2 \qquad (1.2.28)$$

where $0 < p^1 < p^2 < 1$. This procedure simultaneously estimates the two roots x^1 and x^2 of the underlying response function R(x) satisfying:

$$R(x^1) = p^1$$
$$R(x^2) = p^2 \ . \qquad (1.2.29)$$

In simulation studies this procedure has been shown to perform favorably. It also has been shown to provide relatively accurate extrapolated estimates for roots x^* corresponding to values p^* other than p^1 and p^2. Extrapolation is accomplished via the fitted model:

$$\hat{\psi}_n(x^*) = p^* \ . \qquad (1.2.30)$$

Extrapolated estimates tend to be more accurate using this approach than for the original Logit-MLE procedure given the same number of total observations in each. One possible viewpoint concerning this difference is that by estimating roots at $p^1 < p^2$ greater heterogeneity in the control variable settings is induced. This appears to effect close agreement between the fitted model $\hat{\psi}_n(x)$ and the true response function R(x) over a wider range of values for *x*.

As in the case of Wu's original procedure the asymptotic behavior for the Moser-Faries variant remains an open issue. However, by following the discretization approach of Shen and O'Quigley the authors were able to demonstrate similar asymptotic results.

In the work done by Shen and O'Quigley as well as by Moser and Faries it is precisely the restriction of x to a discrete set of points which facilitates proof of various asymptotic results. Discretization of x does, however, have shortcomings. There is no assurance that any of the available design points are "close" to the desired root(s) of the true response function R(x). The convergence results can also be sensitive to *model misspecification*, that is, too great a disagreement between the working model form φ or ψ and R(x). Simulations show that only slight model misspecification can lead to convergence to the wrong design point or a failure to converge altogether. Nonetheless, by selecting a sufficiently fine array of design points estimates are often satisfactory.

1.2.5 Summary

There has been significant advancement in the area of sequential estimation since the advent of the Robbins-Monro procedure. The model-based procedures developed in the 1980s and 1990s represent the "state of the art" in terms of sophistication and finite-sample accuracy in simulations. However, the asymptotic behavior of estimates generated using many model-based procedures is not understood as thoroughly as for earlier procedures. Sequential estimation procedures are intended for use in situations involving limited sample sizes and this might raise the issue of undue concern over asymptotic performance. Nevertheless, when applying any procedure it is reassuring to know that it possesses good large-sample properties so that reasonable finite-sample behavior can be anticipated on grounds other than empirical.

The purpose of this dissertation is to extend what is known about the asymptotic performance of model-based estimation procedures when the control variable x is permitted to vary continuously. In many

experimental settings such an approach seems more natural than requiring x to be discrete. Moreover, any asymptotic results are then more directly comparable to results associated with earlier methods. The particular focus of study is Wu's Logit-MLE procedure. This procedure is described in much greater detail in §1.3. The associated research questions are then briefly presented in §1.4.

1.3 The Adaptive Logit-MLE Procedure

The Logit-MLE procedure proposed by Wu (1985) uses the two-parameter logit function (1.2.22) to model the underlying response function R(x). Various authors refer to μ as the *shift* or *location* parameter and β as the *slope* or *scale* parameter. When β is fixed the model reduces to a one-parameter logit function and the associated estimation procedure is considered nonadaptive. When μ and β are both subject to estimation the procedure is considered adaptive. The present treatment focuses on the adaptive case.

1.3.1 The Startup Phase

The *startup phase* of an experiment relying on the adaptive Logit-MLE procedure involves conducting preliminary trials over various levels of the control variable x until the response pattern fulfills conditions which are necessary and sufficient for existence and uniqueness of the MLEs of the model parameters μ and β. (Determination of the MLEs will be discussed in §2.2.) Letting:

$x_i \equiv$ level of x on trial i

$y_i \equiv$ observed response { failure = 0 , success = 1 } on trial i

the key condition is the presence of overlap between the range of x values associated with "0" responses and the range of x values associated with "1" responses. Based on general results determined by Silvapulle (1981) the overlap conditions for the current model were characterized precisely by Wu. Assuming sample points $\{x_1,\ldots,x_n\}$ and observed responses $\{y_1,\ldots,y_n\}$ with not all y_i equal let:

$$x^-_{min} \equiv \min \{ x_i : y_i = 0 \} \qquad x^-_{max} \equiv \max \{ x_i : y_i = 0 \}$$

$$x^+_{min} \equiv \min \{ x_i : y_i = 1 \} \qquad x^+_{max} \equiv \max \{ x_i : y_i = 1 \} \ .$$

Figure 1.3 identifies these quantities for a hypothetical response pattern:

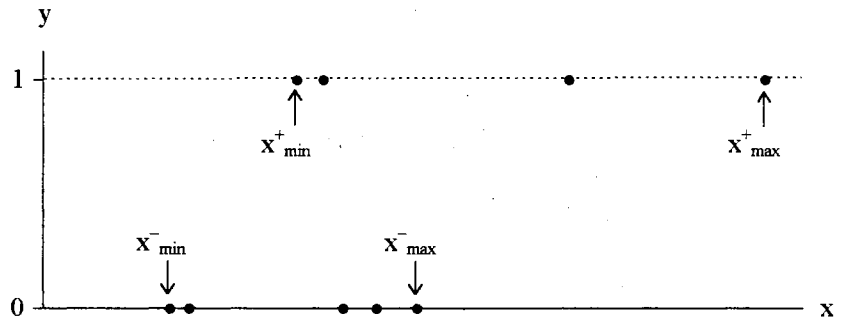

Figure 1.3 Control Variable Ranges for "0" and "1" Responses

The MLEs $\hat{\mu}$ and $\hat{\beta}$ exist (and are unique) if and only if one of the following conditions is satisfied:

(i) $\quad (x^-_{min} , x^-_{max}) \cap (x^+_{min} , x^+_{max}) \neq \varnothing$

(ii) $\quad x^+_{min} < x^-_{min} = x^-_{max} < x^+_{max}$

(iii) $\quad x^-_{min} < x^+_{min} = x^+_{max} < x^-_{max} \ .$

Note that if all "0" responses fall at a single point x^- and all "1" responses fall at a single point x^+ the MLE existence and uniqueness condition cannot be satisfied. Once satisfied, the overlap condition will persist for the remainder of the experiment and the startup phase is terminated. The startup phase is not of principal concern here, and will henceforth be referred to only peripherally.

1.3.2 The Sequential Estimation Phase

After completion of the startup phase all further trials are to be conducted sequentially by letting the fitted curve $\hat{\psi}(x) \equiv \psi(x;\hat{\mu},\hat{\beta})$ guide the selection of sample points in an iterative manner. In this *sequential phase* of the estimation process, let:

$\hat{\mu}_n \equiv$ MLE of μ based on observed data points (x_1,y_1) , ... , (x_n,y_n)

$\hat{\beta}_n \equiv$ MLE of β based on observed data points (x_1,y_1) , ... , (x_n,y_n) .

After completion of trial n the value of the next sample point will generally be determined by solving:

$$\psi(x;\hat{\mu}_n,\hat{\beta}_n) = p^0 \qquad (1.3.1)$$

for $x = x_{n+1}$. The next sample point x_{n+1} is thus determined by finding the value of the control variable x at which the currently fitted function assumes the value p^0. See Figure 1.4 below:

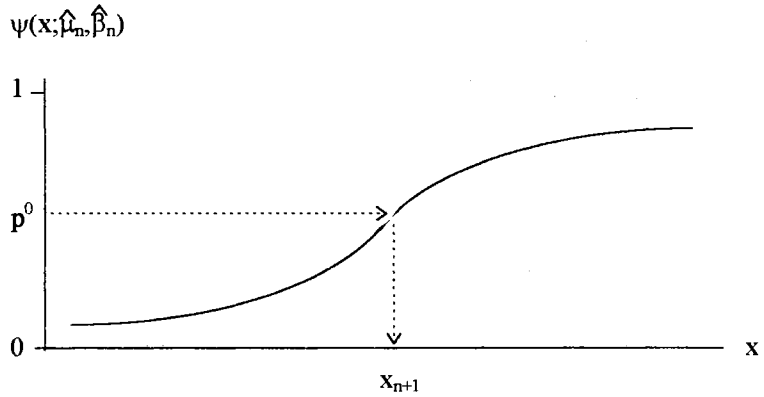

Figure 1.4 Sample Point Selection in the Adaptive Logit-MLE Procedure

If $\hat{\beta}_n \neq 0$ the solution to (1.3.1) is easily shown to be:

$$x_{n+1} = [\log(p^0) - \log(1-p^0) - \hat{\mu}_n] / \hat{\beta}_n . \qquad (1.3.2)$$

If some stopping criterion for the experiment is satisfied then x_{n+1} represents the terminal estimate of x^0. Otherwise one trial is performed with $x = x_{n+1}$ and the resulting binary response y_{n+1} is recorded. Then based on all previously observed data and the new observation:

$$\{ (x_1,y_1) , \ldots , (x_n,y_n) \} \cup \{ (x_{n+1},y_{n+1}) \}$$

revised MLEs $\hat{\mu}_{n+1}$ and $\hat{\beta}_{n+1}$ are calculated and the cycle is repeated.

1.3.3 The Adaptive Logit-MLE Procedure

Sections 1.3.1 and 1.3.2 present the essential ideas underlying the proposed estimation approach. To remove any ambiguity the procedure described in Figure 1.5 on the following page will henceforth represent the implementation of the adaptive Logit-MLE procedure unless stated otherwise. The following definitions are necessary:

$$n \equiv \text{current trial index}$$

$$n_I \equiv \text{index of final trial of the startup phase} \qquad (1.3.3)$$

$$n_T \equiv \text{index of terminal trial of sequential phase} .$$

As a practical matter it may happen that the MLE existence and uniqueness conditions described in §1.3.1 are not satisfied in a particular experiment or simulation run with a fixed sample size for any $n \leq n_T$. In an actual experiment it is anticipated that the experimenter's knowledge in choosing the initial levels for x combined with a sufficiently large allowance for n_T should preclude this possibility. In simulation studies

1. Conduct the startup phase. Initialize $n = n_I$.
2. Calculate the MLEs $\hat{\mu}_n$ and $\hat{\beta}_n$.
3. If $\hat{\beta}_n \neq 0$ calculate x_{n+1} according to (1.3.2). Otherwise set $x_{n+1} = x_n$.
4. If $n = n_T$ stop. The terminal estimate of x^0 is x_{n+1}.
5. Conduct trial n+1 by observing the response y_{n+1} at the sample point x_{n+1}.
6. Increment the trial counter: $n = n+1$.
7. Return to step 2.

Figure 1.5 Formal Specification of the Adaptive Logit-MLE Procedure

the most expedient measure is to simply discard such runs. At any rate, the present concern is the performance of the procedure as $n_T \rightarrow \infty$ and it will therefore be assumed that the startup conditions can always be satisfied.

1.3.4 Known Asymptotic Properties

If the scale parameter is fixed at some constant β^0 the Logit-MLE procedure is considered nonadaptive. Wu demonstrated that the nonadaptive Logit-MLE procedure is equivalent to a Robbins-Monro process having iteration form:

$$x_{n+1} = x_n - (d_n/n)\cdot(y_n - p^0) .$$

The term d_n is naturally bounded from below by a positive constant, but must be artificially bounded from above to attain convergence. The lower bound d can be determined analytically and by choosing an upper bound $D > d$ and defining:

$$D_n = \min \{d_n , D\}$$

Wu argued that the modified Robbins-Monro iteration:

$$x_{n+1} = x_n - (D_n/n)\cdot(y_n - p^0)$$

will result in $x_n \rightarrow x^0$ almost surely. No results on asymptotic normality or an optimal rate of convergence are implied.

In the adaptive procedure both of the parameters μ and β are estimated. Wu demonstrated that by assuming $(\hat{\mu}_n,\hat{\beta}_n)$ converges uniformly to *some* constant (μ^*,β^*) where $\beta^* \neq 0$ it follows that $x_n \rightarrow x^0$ almost surely. Such consistency implies that most of the sample points will eventually be in close proximity, and assuming this the following first-order approximation to the adaptive procedure was derived:

$$x_{n+1} = x_n - (1/(n\cdot C_n)) \cdot (y_n - p^0) \qquad (1.3.4)$$

where:

$$C_n = \sum_{k=1}^{n} (x_k - \bar{x}_n)\cdot y_k \; / \; \sum_{k=1}^{n} (x_k - x_n)^2 \; . \qquad (1.3.5)$$

Because x_n rather than $\bar{x}_n$ appears in the denominator of (1.3.5) this expression does not represent the usual regression slope estimator B_n. Wu provided the argument, however, that the consistency of x_n also implies that $C_n \rightarrow B_n$ as $n \rightarrow \infty$ and that therefore the first-order approximation (1.3.4) will be asymptotically equivalent to the adaptive Robbins-Monro process proposed by Anbar (1978). Under conditions similar to those required by Anbar for asymptotic optimality (see §1.2.3) this approximation to the adaptive Logit-MLE procedure should therefore also be asymptotically optimal.

1.4 Research Questions

By assuming that the MLEs $(\hat{\mu}_n,\hat{\beta}_n)$ converge to some constant (μ^*,β^*) in the adaptive Logit-MLE procedure Wu was able to show that $x_n \to x^0$ almost surely. This consistency result was then used to justify the first-order approximation (1.3.4) to the adaptive procedure. By applying the consistency result again it was argued further that the approximation will be asymptotically equivalent to an adaptive Robbins-Monro process that is asymptotically optimal. Several questions are left open:

1. *Does x_n actually converge to the correct root x^0 in the adaptive procedure*?

   Because of the nonlinearity of the two-parameter logit model the analysis of the adaptive procedure is aided by first-order approximations. The particular approximation (1.3.4) derived by Wu may not be well-suited for studying consistency of the adaptive procedure, however, since it relies on consistency in its derivation.

   An alternative first-order approximation will be formulated in §2. The formulation will rely on the Newton-Raphson method as applied in the determination of the MLEs $(\hat{\mu}_n,\hat{\beta}_n)$. The approximation will then be used to define a modified adaptive procedure to serve as the basis for further study. In the discussion which follows the sequence of estimates generated by this approximating procedure will be denoted by $\{w_n\}$ in order to distinguish it from the sequence $\{x_n\}$ generated by the original procedure.

   In §3 assumptions will be presented which support a conclusion of almost sure convergence of the sequence $\{w_n\}$ to the root value x^0. It should be possible to reach this conclusion without imposing assumptions concerning convergence of the MLEs $(\hat{\mu}_n,\hat{\beta}_n)$. The convergence proof will appeal to the *almost supermartingale theorem* of Robbins and Siegmund (1971).

2. *To what extent is it reasonable to study the adaptive Logit-MLE procedure by way of first-order approximations*?

This question will be addressed in §4 by means of simulation. Comparison of the original adaptive Logit-MLE procedure and the approximating procedure can be accomplished using simulation runs based on identical streams of pseudo-random numbers. By addressing the accuracy of the first-order approximation the utility of the results associated with question 1 may be evaluated accordingly. The specific method is detailed in §4.1.

CHAPTER 2

FIRST-ORDER APPROXIMATION OF THE LOGIT-MLE PROCEDURE

2.1 A Rationale for Approximation

The adaptive Logit-MLE procedure defined in §1.3 determines successive sample points by means of the inverse of the fitted two-parameter logit function:

$$x_{n+1} = [\log(p^0) - \log(1-p^0) - \hat{\mu}_n] / \hat{\beta}_n \qquad (2.1.1)$$

provided that $\hat{\beta}_n \neq 0$. The MLEs $\hat{\mu}_n$ and $\hat{\beta}_n$ are not the result of any closed-form function of the observed data points (x_1,y_1) , ... , (x_n,y_n) and hence no closed-form expression in (x_1,y_1) , ... , (x_n,y_n) is available by which to directly analyze the actions of the procedure. The goal of this chapter is to develop an approximation to the adaptive Logit-MLE procedure which is more amenable to analysis than (2.1.1). This approximation will be expressed in terms of the observed data but will not be completely free of the MLEs. The major advantage it will offer is its conformity to an adaptive Robbins-Monro iteration of the form:

$$x_{n+1} = x_n - a_n((x_1,y_1),\ldots,(x_{n-1},y_{n-1})) \cdot (y_n - p^0) \ . \qquad (2.1.2)$$

The benefit of working with this form is that one may appeal to the vast collection of results applicable to Robbins-Monro processes. In this particular adaptive form the multiplier a_n is not fixed in advance but is instead a function of the previously observed pairs (x_1,y_1) , ... , (x_{n-1},y_{n-1}) thereby making it suitable for application of the *almost supermartingale theorem* of Robbins and Siegmund (1971). Application of this theorem to obtain a convergence result will be the main topic in §3.

2.2 The MLE Normal Equations

This section details the formulation of the normal equations which determine the MLEs $\hat{\mu}_n$ and $\hat{\beta}_n$ on each iteration of the sequential phase of the adaptive Logit-MLE procedure. Given the assumed two-parameter logit model (1.2.22) and a set of observations recorded over n independent trials:

$$(x_1,y_1)\ ,\ \ldots\ ,\ (x_n,y_n)$$

the likelihood function for the parameters μ and β is:

$$\mathcal{L}_n(\mu,\beta) \equiv \prod_{i=1}^{n} [\psi(x_i;\mu,\beta)]^{y_i} \cdot [1-\psi(x_i;\mu,\beta)]^{1-y_i} \equiv \prod_{i=1}^{n} [\, e^{\mu+\beta x_i}\,]^{y_i} \,/\, [\,1+e^{\mu+\beta x_i}\,]\ .$$

The log-likelihood function is then given by:

$$\log \mathcal{L}_n(\mu,\beta) \equiv \sum_{i=1}^{n} [\, y_i(\mu+\beta x_i) - \log(1+e^{\mu+\beta x_i})] \ . \tag{2.2.1}$$

The partial derivatives of the log-likelihood function wrt μ and β are:

$$\partial \log \mathcal{L}_n(\mu,\beta) / \partial\mu = \sum_{i=1}^{n} [\, y_i - e^{\mu+\beta x_i} / (1+e^{\mu+\beta x_i})] = \sum_{i=1}^{n} [\, y_i - \psi(x_i;\mu,\beta)]$$

and:

(2.2.2)

$$\partial \log \mathcal{L}_n(\mu,\beta) / \partial\beta = \sum_{i=1}^{n} x_i\cdot[y_i - e^{\mu+\beta x_i} / (1+e^{\mu+\beta x_i})] = \sum_{i=1}^{n} x_i\cdot[y_i - \psi(x_i;\mu,\beta)]\ .$$

Setting both partial derivatives to zero results in the following *normal equations*:

$$N_n^1(\mu,\beta) \equiv \sum_{i=1}^{n} [y_i - \psi(x_i;\, \mu,\, \beta)] = 0 \,,$$

$$N_n^2(\mu,\beta) \equiv \sum_{i=1}^{n} x_i \cdot [y_i - \psi(x_i;\, \mu,\, \beta)] = 0 \,. \qquad (2.2.3)$$

Under conditions detailed in §1.3.1 there will exist a unique solution $(\hat{\mu}_n, \hat{\beta}_n)$ to system (2.2.3). The same conditions also imply that there will exist some minimal heterogeneity in the sample points $\{x_1,\ldots,x_n\}$. By Lemma A.1 of Appendix A the following Hessian matrix associated with the log-likelihood function:

$$\begin{bmatrix} \partial^2 \log \mathcal{L}_n(\mu,\beta) / \partial\mu^2 & \partial^2 \log \mathcal{L}_n(\mu,\beta) / \partial\mu\partial\beta \\ \partial^2 \log \mathcal{L}_n(\mu,\beta) / \partial\beta\partial\mu & \partial^2 \log \mathcal{L}_n(\mu,\beta) / \partial\beta^2 \end{bmatrix}$$

will then be negative definite. The solution to (2.2.3) therefore can be taken as maximizing the log-likelihood function. Solutions of system (2.2.3) must generally be determined numerically. A typical approach is to apply the Newton-Raphson method. For details the reader is referred to Agresti (1990).

Since the Hessian matrix will be negative definite then it will also have a nonvanishing determinant. As a consequence the solution $(\hat{\mu}_n, \hat{\beta}_n)$ to system (2.2.3) will vary continuously in the data values $x_1, y_1, \ldots, x_n, y_n$. This may be formally demonstrated by applying the implicit function theorem to system (2.2.3) at the solution $(x_1, y_1, \ldots, x_n, y_n, \hat{\mu}_n, \hat{\beta}_n)$ and noting that the appropriate Jacobian determinant will equal the Hessian determinant for the log-likelihood function.

2.3 Step Size Approximation

For any two successive estimates x_n and x_{n+1} generated by the adaptive Logit-MLE procedure define:

$$\Delta x_{n+1} \equiv x_{n+1} - x_n . \tag{2.3.1}$$

The term Δx_{n+1} will be referred to as the *step size*. The primary goal of this section is to develop some form for expressing the step size which will facilitate further analysis, in particular one which will lend itself to obtaining a convergence result. The treatment assumes that the startup phase described in §1.3.1 has been completed and that the MLEs $\hat{\mu}_n$ and $\hat{\beta}_n$ exist. Assuming that $\hat{\beta}_n \neq 0$ upon completion of a given trial n the next sample point will be determined according to (2.1.1). That is:

$$x_{n+1} = [\log(p^0) - \log(1-p^0) - \hat{\mu}_n] / \hat{\beta}_n .$$

Defining $C^0 \equiv \log(p^0) - \log(1-p^0)$ this can be written more compactly as:

$$x_{n+1} = [C^0 - \hat{\mu}_n] / \hat{\beta}_n .$$

By also defining:

$$\Delta\hat{\mu}_n \equiv \hat{\mu}_n - \hat{\mu}_{n-1}$$

$$\Delta\hat{\beta}_n \equiv \hat{\beta}_n - \hat{\beta}_{n-1}$$

the step size can be expressed as:

$$\Delta x_{n+1} = [C^0 - \hat{\mu}_n] / \hat{\beta}_n - [C^0 - \hat{\mu}_{n-1}] / \hat{\beta}_{n-1}$$

$$= [C^0 - (\hat{\mu}_{n-1} + \Delta\hat{\mu}_n)] / [\hat{\beta}_{n-1} + \Delta\hat{\beta}_n] \quad - \quad [C^0 - \hat{\mu}_{n-1}] / \hat{\beta}_{n-1}$$

$$= - [\Delta\hat{\mu}_n \cdot \hat{\beta}_{n-1} + \Delta\hat{\beta}_n \cdot (C^0 - \hat{\mu}_{n-1})] / [\hat{\beta}_{n-1} \cdot (\hat{\beta}_{n-1} + \Delta\hat{\beta}_n)]$$

$$= - [\Delta\hat{\mu}_n \cdot \hat{\beta}_{n-1} + \Delta\hat{\beta}_n \cdot (C^0 - \hat{\mu}_{n-1})] / [\hat{\beta}_{n-1} \cdot \hat{\beta}_n]$$

$$= - (1/\hat{\beta}_n) \cdot [\Delta\hat{\mu}_n + \Delta\hat{\beta}_n \cdot (C^0 - \hat{\mu}_{n-1})/\hat{\beta}_{n-1}]$$

$$= - (1/\hat{\beta}_n) \cdot [\Delta\hat{\mu}_n + \Delta\hat{\beta}_n \cdot x_n] \,. \qquad (2.3.2)$$

Based on (2.3.2) the behavior of Δx_{n+1} may be studied through the behavior of $\Delta\hat{\mu}_n$ and $\Delta\hat{\beta}_n$. Note that the lhs of the system of normal equations (2.2.3) can be written:

$$N_n^{\,1}(\mu,\beta) \equiv N_{n-1}^{\,1}(\mu,\beta) + [y_n - \psi(x_n; \mu, \beta)] \quad ,$$

$$N_n^{\,2}(\mu,\beta) \equiv N_{n-1}^{\,2}(\mu,\beta) + x_n \cdot [y_n - \psi(x_n; \mu, \beta)] \,.$$

At the beginning of trial n the values $\hat{\mu}_{n-1}$ and $\hat{\beta}_{n-1}$ are known and these quantities satisfy:

$$N_{n-1}^{\,1}(\hat{\mu}_{n-1}, \hat{\beta}_{n-1}) = 0 \,,$$

$$N_{n-1}^{\,2}(\hat{\mu}_{n-1}, \hat{\beta}_{n-1}) = 0 \,.$$

Therefore:

$$N_n^1(\hat{\mu}_{n-1},\hat{\beta}_{n-1}) = [y_n - \psi(x_n ; \hat{\mu}_{n-1}, \hat{\beta}_{n-1})] \quad ,$$

$$N_n^2(\hat{\mu}_{n-1},\hat{\beta}_{n-1}) = x_n \cdot [y_n - \psi(x_n ; \hat{\mu}_{n-1}, \hat{\beta}_{n-1})] \ .$$

Since x_n will have been selected so that $\psi(x_n ; \hat{\mu}_{n-1}, \hat{\beta}_{n-1}) = p^0$ this last system further simplifies to:

$$N_n^1(\hat{\mu}_{n-1},\hat{\beta}_{n-1}) = y_n - p^0 \quad ,$$

$$N_n^2(\hat{\mu}_{n-1},\hat{\beta}_{n-1}) = x_n \cdot [y_n - p^0] \ .$$

Extracting the common element on the rhs and writing the system in vector form results in:

$$\begin{bmatrix} N_n^1(\hat{\mu}_{n-1},\hat{\beta}_{n-1}) \\ \\ N_n^2(\hat{\mu}_{n-1},\hat{\beta}_{n-1}) \end{bmatrix} = (y_n - p^0) \cdot \begin{bmatrix} 1 \\ \\ x_n \end{bmatrix} \ . \tag{2.3.3}$$

Both sides of (2.3.3) represent the gradient of the log-likelihood function $\log \mathcal{L}_n(\mu,\beta)$ evaluated at the prior MLEs $(\hat{\mu}_{n-1},\hat{\beta}_{n-1})$. A single iteration of the Newton-Raphson procedure applied to the functions N_n^1 and N_n^2 starting at $(\hat{\mu}_{n-1},\hat{\beta}_{n-1})$ provides an approximation of $(\hat{\mu}_n,\hat{\beta}_n) - (\hat{\mu}_{n-1},\hat{\beta}_{n-1}) \equiv (\Delta\hat{\mu}_n,\Delta\hat{\beta}_n)$. Based on the rhs of (2.3.3) this approximation is:

$$\begin{bmatrix} \Delta\hat{\mu}_n \\ \\ \Delta\hat{\beta}_n \end{bmatrix} \cong \begin{bmatrix} \widetilde{\Delta\mu}_n \\ \\ \widetilde{\Delta\beta}_n \end{bmatrix} = -[\mathbf{H}(x_1,\ldots,x_n,\hat{\mu}_{n-1},\hat{\beta}_{n-1})]^{-1} \cdot (y_n - p^0) \cdot \begin{bmatrix} 1 \\ \\ x_n \end{bmatrix} \tag{2.3.4}$$

with the 2×2 Hessian matrix $\mathbf{H}_n \equiv \mathbf{H}(x_1,\ldots,x_n,\hat{\mu}_{n-1},\hat{\beta}_{n-1})$ given explicitly by:

$$\mathbf{H}_n \equiv \mathbf{H}(x_1,\ldots,x_n,\hat{\mu}_{n-1},\hat{\beta}_{n-1}) \equiv \begin{bmatrix} -\sum_{i=1}^{n} z_i & -\sum_{i=1}^{n} x_i \cdot z_i \\ -\sum_{i=1}^{n} x_i \cdot z_i & -\sum_{i=1}^{n} x_i^2 \cdot z_i \end{bmatrix} \tag{2.3.5}$$

where the scalar elements z_i are defined as:

$$z_i \equiv z(x_i;\hat{\mu}_{n-1},\hat{\beta}_{n-1}) \equiv \psi(x_i;\hat{\mu}_{n-1},\hat{\beta}_{n-1}) \cdot [1 - \psi(x_i;\hat{\mu}_{n-1},\hat{\beta}_{n-1})] \; . \tag{2.3.6}$$

Given any heterogeneity among the sample points $\{x_1,\ldots,x_n\}$ this Hessian will be negative definite by Lemma A.1 and the indicated inverse exists. Substituting the inverse of (2.3.5) into (2.3.4) yields the following expression for the approximation:

$$\begin{bmatrix} \widetilde{\Delta\mu}_n \\ \widetilde{\Delta\beta}_n \end{bmatrix} = \frac{y_n - p^0}{\det(\mathbf{H}_n)} \begin{bmatrix} \sum_{i=1}^{n} x_i^2 \cdot z_i & -\sum_{i=1}^{n} x_i \cdot z_i \\ -\sum_{i=1}^{n} x_i \cdot z_i & \sum_{i=1}^{n} z_i \end{bmatrix} \begin{bmatrix} 1 \\ x_n \end{bmatrix} .$$

Carrying out the matrix-vector multiplication on the rhs provides a somewhat simpler form:

$$\begin{bmatrix} \widetilde{\Delta\mu}_n \\ \widetilde{\Delta\beta}_n \end{bmatrix} = \frac{y_n - p^0}{\det(\mathbf{H}_n)} \begin{bmatrix} \sum_{i=1}^{n} x_i \cdot (x_i - x_n) \cdot z_i \\ \sum_{i=1}^{n} (x_n - x_i) \cdot z_i \end{bmatrix} . \tag{2.3.7}$$

Even for moderate n approximation (2.3.7) empirically gives excellent estimates of $\Delta\hat{\mu}_n$ and $\Delta\hat{\beta}_n$. Applying this approximation the quantity $\Delta\hat{\mu}_n + \Delta\hat{\beta}_n \cdot x_n$ can in turn be approximated by:

$$\Delta\hat{\mu}_n + \Delta\hat{\beta}_n \cdot x_n \cong \widetilde{\Delta\mu_n} + \widetilde{\Delta\beta_n} \cdot x_n$$

$$= [y_n - p^0] \cdot \sum_{i=1}^{n} [x_i \cdot (x_i - x_n) \cdot z_i + x_n \cdot (x_n - x_i) \cdot z_i] \; / \; \det(\mathbf{H}_n)$$

$$= [y_n - p^0] \cdot \sum_{i=1}^{n} [x_i \cdot (x_i - x_n) \cdot z_i - x_n \cdot (x_i - x_n) \cdot z_i] \; / \; \det(\mathbf{H}_n)$$

$$= [y_n - p^0] \cdot \sum_{i=1}^{n} (x_i - x_n)^2 \cdot z_i \; / \; \det(\mathbf{H}_n) \; . \qquad (2.3.8)$$

Substituting approximation (2.3.8) into (2.3.2) gives the following approximation for the step size:

$$\Delta x_{n+1} \cong \widetilde{\Delta x_{n+1}} = [y_n - p^0] \cdot [- (1/\hat{\beta}_n) \cdot \sum_{i=1}^{n} (x_i - x_n)^2 \cdot z_i \; / \; \det(\mathbf{H}_n)] \; . \qquad (2.3.9)$$

Equivalently (2.3.9) may be written:

$$x_{n+1} \cong x_n + \widetilde{\Delta x_{n+1}}$$

$$= x_n - [(1/\hat{\beta}_n) \cdot \sum_{i=1}^{n} (x_i - x_n)^2 \cdot z_i \; / \; \det(\mathbf{H}_n)] \cdot [y_n - p^0] \; . \qquad (2.3.10)$$

It has already been mentioned that the 2×2 Hessian $\mathbf{H}_n$ is negative definite and therefore $\det(\mathbf{H}_n) > 0$. Provided that the MLE $\hat{\beta}_n$ becomes persistently positive at some point in the process then (2.3.10) has the form of an adaptive Robbins-Monro iteration in which the multiplier a_n is defined by:

$$a_n \equiv (1/\hat{\beta}_n) \cdot \sum_{i=1}^{n} (x_i - x_n)^2 \cdot z_i \; / \; \det(\mathbf{H}_n) \; . \tag{2.3.11}$$

The approximation developed in this section provides the basis for an approximating Logit-MLE procedure to be defined in the next section. Under reasonably mild assumptions the approximating procedure will be shown to generate a sequence of estimates which converge to x^0 almost surely.

2.4 An Approximating Procedure

Approximation (2.3.10) will be used to modify the adaptive Logit-MLE procedure defined in §1.3.3. The sequence of estimates of x^0 generated by the modified procedure will be denoted by $\{w_n\}$ to clearly distinguish it from the sequence $\{x_n\}$ generated by the original adaptive procedure. The startup phase of the modified procedure will be identical in structure to that of the original procedure. The sequential phase in the modified procedure will also have the same structure as in the original procedure up to some predetermined trial index n_E. That is, for sequential trials $n_I < n \leq n_E$ the sample points will be selected using the inverse of the fitted two-parameter logit function just as in the original adaptive procedure:

$$w_{n+1} = [\log(p^0) - \log(1-p^0) - \hat{\mu}_n] \; / \; \hat{\beta}_n \tag{2.4.1}$$

it being understood that the MLEs are now based on data points labeled as (w_i, y_i). After trial n_E of the sequential phase, however, sample points will be determined according to the approximation:

$$w_{n+1} = w_n - [(1/\hat{\beta}_{n-1}) \cdot \sum_{i=1}^{n} (w_i - w_n)^2 \cdot z_i \; / \; \det(\mathbf{H}_n)] \cdot [y_n - p^0] \tag{2.4.2}$$

where the terms $z_1,\ldots, z_n$ and $\mathbf{H}_n$ are similarly understood to rely on the data points (w_i,y_i):

$$z_i \equiv z(w_i; \hat{\mu}_{n-1}, \hat{\beta}_{n-1})$$

$$\mathbf{H}_n \equiv \mathbf{H}(w_1,\ldots,w_n, \hat{\mu}_{n-1}, \hat{\beta}_{n-1}) \ . \tag{2.4.3}$$

Exact definitions of z_i and $\mathbf{H}_n$ appear in §2.3. The sequential estimation procedure obtained through the use of this approximation scheme will henceforth be referred to as the *approximating* Logit-MLE procedure.

Note that the multiplier a_n in (2.4.2) is defined by:

$$a_n \equiv (1/\hat{\beta}_{n-1}) \cdot \sum_{i=1}^{n} (w_i - w_n)^2 \cdot z_i \ / \ \det(\mathbf{H}_n) \ . \tag{2.4.4}$$

This multiplier has the same form as in (2.3.11) with the exception that $\hat{\beta}_{n-1}$ appears in place of $\hat{\beta}_n$. The reason is that $\hat{\beta}_n$ depends explicitly on the binary response y_n while $\hat{\beta}_{n-1}$ does not. In §3.4 the *almost supermartingale theorem* of Robbins and Siegmund (1971) will be applied to show convergence of the approximating procedure and this theorem technically does not permit functional dependency between a_n and the pair (w_n,y_n). It may be noted that a_n still appears to involve a dependency on the sample point w_n directly through the terms $(w_i - w_n)^2$ and indirectly through z_n and $\mathbf{H}_n$. However, w_n is completely determined by $(w_1,y_1),\ldots,(w_{n-1},y_{n-1})$ via the estimation procedure and therefore a_n as defined in (2.4.4) depends only on $(w_1,y_1),\ldots,(w_{n-1},y_{n-1})$. Early adaptive procedures such as that of Anbar (1978) abided by this technicality; in papers appearing later it seems to have often been dismissed. A conservative approach will be taken here.

The framework used to present the original adaptive Logit-MLE procedure in Figure 1.5 of §1.3.3 is shown in Figure 2.1 on the next page with appropriate modifications to define the approximating procedure. The trial index definitions (1.3.3) are also repeated below with the addition of an index reference value dividing the sequential trials into the exact phase and the subsequent approximating phase:

$$
\begin{aligned}
n &\equiv \text{current trial index} \\
n_I &\equiv \text{index of final trial of the startup phase} \\
n_E &\equiv \text{index of final sequential trial prior to commencing approximation} \\
n_T &\equiv \text{index of terminal trial of sequential phase} .
\end{aligned}
\tag{2.4.5}
$$

It is assumed that $n_I < n_E < n_T$. As the terminal trial index $n_T \to \infty$ the asymptotic performance of the approximating procedure becomes relevant. The approximating procedure is intended only for use in analyzing the original adaptive procedure. As stated in §1.4 the principal objective is to prove almost sure convergence of the approximating sequence $\{w_n\}$ to the root x^0. Treatment of convergence will be the subject of §3.

Once convergence has been proven the next issue is the extent to which the step size in the approximating procedure corresponds to the step size in the true Logit-MLE procedure. Simulation will be employed for this purpose and the index reference value n_E will be useful in this regard. By using identical streams of pseudo-random variates the approximating and original procedures can be forced to maintain identical states through the index value n_E. Only after trial n_E can the sequences $\{w_n\}$ and $\{x_n\}$ differ. The idea is to set n_E large enough to allow the estimation process to "quiet down" and thereafter compare the approximated and actual step sizes. Treatment of this issue is the subject of §4.

1. Conduct the startup phase. Initialize $n = n_I$.

Sequential Trials: *Exact Phase*

2. Calculate the MLEs $\hat{\mu}_n$ and $\hat{\beta}_n$.
3. If $\hat{\beta}_n \neq 0$ calculate w_{n+1} according to (2.4.1). Otherwise set $w_{n+1} = w_n$.
4. Conduct trial n+1 by observing response y_{n+1} at the sample point w_{n+1}.
5. Increment the trial counter: $n = n+1$.
6. If $n < n_E$ return to step 2.

Sequential Trials: *Approximating Phase*

7. Calculate the MLEs $\hat{\mu}_{n-1}$ and $\hat{\beta}_{n-1}$.
8. If $\hat{\beta}_{n-1} \neq 0$ calculate w_{n+1} according to (2.4.2). Otherwise set $w_{n+1} = w_n$.
9. If $n = n_T$ stop. The terminal estimate of x^0 is w_{n+1}.
10. Conduct trial n+1 by observing response y_{n+1} at the sample point w_{n+1}.
11. Increment the trial counter: $n = n+1$.
12. Return to step 7.

Figure 2.1 Approximating Procedure

CHAPTER 3

ANALYSIS OF THE APPROXIMATING LOGIT-MLE PROCEDURE

The present chapter analyzes the asymptotic properties of the approximating Logit-MLE procedure defined in §2.4. The issue of foremost interest is whether the sequence $\{w_n\}$ generated by the approximating procedure will converge to x^0.

The iteration (2.4.2) is the basis of the approximating procedure. Under certain assumptions the *almost supermartingale theorem* of Robbins and Siegmund (1971) can be applied to this iteration to demonstrate that $w_n \rightarrow x^0$ almost surely. Section 3.1 below details the framework in which this theorem may be applied to stochastic approximation problems in general. The remaining sections of this chapter concern the application of this framework to the approximating Logit-MLE procedure.

3.1 Applying the Almost Supermartingale Theorem

Application 2 of Robbins and Siegmund (1971) pertains to problems such as the EQRP defined in §1.1. The iteration studied by Robbins and Siegmund has the generic form:

$$w_{n+1} = w_n - a_n((w_1,t_1),\ldots,(w_{n-1},t_{n-1})) \cdot t_n \tag{3.1.1}$$

where $a_n(\cdot)$ is a nonnegative measurable function of the n-1 previous pairs (w_1,t_1) , ... , (w_{n-1},t_{n-1}) with $a_1(\cdot)$ arbitrary. Conditional on these n-1 pairs t_n is a random variable parameterized by $w = w_n$ such that:

$$\begin{aligned} E[t_n] &= M(w_n) \\ \\ Var[t_n] &\equiv \sigma^2(w_n) < \infty . \end{aligned} \tag{3.1.2}$$

The functions $M(\cdot)$ and $\sigma(\cdot)$ must also be measurable and there must exist constants $a,b > 0$ such that:

$$\sigma(w) + |M(w)| \leq a + b\cdot|w| \tag{3.1.3}$$

for all $-\infty < w < \infty$. Moreover, there must exist a real number θ such that for every $0 < \varepsilon < 1$:

$$\inf_{\varepsilon < |w-\theta| < 1/\varepsilon} M(w)\cdot(w-\theta) > 0 . \tag{3.1.4}$$

Then, if for every sequence (w_1,t_1), (w_2,t_2), (w_3,t_3), ... it holds that:

$$\sum_{n=1}^{\infty} a_n((w_1,t_1),\ldots,(w_{n-1},t_{n-1})) = \infty \qquad \text{(at least whenever } \sup_n |w_n| < \infty) \tag{3.1.5}$$

and:

$$\sum_{n=1}^{\infty} a_n((w_1,t_1),\ldots,(w_{n-1},t_{n-1}))^2 < \infty \tag{3.1.6}$$

it follows from the *almost supermartingale theorem* that $w_n \rightarrow \theta$ a.s.

3.2 Assumptions Supporting Proof of Convergence of the Approximating Procedure

This section details the assumptions under which a proof of convergence of the sequence $\{w_n\}$ generated by the approximating Logit-MLE procedure can be obtained. Four assumptions have already been made in earlier chapters, characterizing the estimation problem and the preliminary conditions which must be satisfied prior to commencement of the sequential phase of the procedure:

(A1) The target probability p^0 is confined to the open interval (0,1).

(A2) The underlying quantal response function $R(\cdot)$ is continuous and strictly increasing.

(A3) Given the control variable level w_n on trial n the response variable $y_n \sim BIN(1,R(w_n))$.

(A4) Upon completion of the startup phase the pattern of sample points and responses will exhibit the overlap necessary for existence of the MLEs as discussed in §1.3.1.

Satisfying the "overlap conditions" automatically implies that the sample points used during the startup phase cannot all be equal. Specifically, if n_I is the index of the final startup trial there will exist $w_i \neq w_j$ for some $1 \leq i < j \leq n_I$.

When n_T of (2.4.5) is made arbitrarily large the following additional assumptions become meaningful:

(A5) There will exist some index value N_1 such that on any trial $n \geq N_1$ the control variable w_n will lie in some bounded interval of $\mathbb{R}$ characterized by $[w_{min},w_{max}]$.

(A6) There will exist some index value N_2 such that upon completion of any trial $n \geq N_2$ the MLEs $(\hat{\mu}_n,\hat{\beta}_n)$ lie in some bounded subset of $\mathbb{R}^2$ characterized as $[\mu_{min},\mu_{max}] \times [\beta_{min},\beta_{max}]$ where $\beta_{min} > 0$.

The restriction on the MLE $\hat{\beta}_n$ means that any fitted curve $\psi(w;\hat{\mu}_n,\hat{\beta}_n)$ must be strictly increasing following completion of trial N_2.

(A7) There will exist some index value N_3 and some constant M such that on every trial $n \geq N_3$ it will hold that $(w_n - \overline{w}_n)^2 \leq M^2 \cdot \hat{\sigma}_{w,n}^2$ where $\overline{w}_n$ and $\hat{\sigma}_{w,n}^2$ denote the empirical mean and variance (see Appendix A) of $w_1,\ldots, w_n$. In other words, on any trial $n \geq N_3$ the *current* sample point w_n will be assumed to fall within $\pm M$ empirical standard deviations of the current sample point mean.

3.3 Asymptotically Invariant Bounds

In this section several quantities are defined which are important in the asymptotic analysis of the approximating Logit-MLE procedure.

By assumption (A5) of §3.2 the iterates w_n generated by the approximating procedure may vary over an unrestricted range until trial N_1. On trial N_1 and all subsequent trials w_n will be confined to the interval $[w_{min},w_{max}]$ specified in (A5). As of trial N_1 there will thus exist an asymptotically invariant bound:

$$w_{abs} \equiv \max \{ |w_1|,\ldots, |w_{N_1}|, |w_{min}|, |w_{max}| \}$$

and it will hold that $w_i \in [-w_{abs}, w_{abs}]$ for all iterates $w_1,w_2,w_3, \ldots$.

An asymptotically invariant bound on the terms z_i defined in (2.3.6) will also exist. Based on the bound w_{abs} defined above and the intervals specified in assumptions (A5) and (A6) of §3.2 the following bound may be defined:

$$\varepsilon_z \equiv \min_{\substack{w \in [-w_{abs}, w_{abs}] \\ \mu \in [\mu_{min}, \mu_{max}] \\ \beta \in [\beta_{min}, \beta_{max}]}} \{ \psi(w;\mu,\beta)\cdot[1 - \psi(w;\mu,\beta)] \}$$

Referring again to assumptions (A5) and (A6) let $N \equiv \max\{N_1,N_2\}$. The terms z_i in (2.4.4) are not static; *each one* changes after each trial as the MLEs $\hat{\mu}_n$ and $\hat{\beta}_n$ are updated. However, on any trial $n > N$ it will hold that $0 < \varepsilon_z \le z_i < 1$ for all $i = 1,2,3,\ldots,n$. In this way ε_z acts as a "retroactive" lower bound for *all* z_i. Moreover, the terms $\varepsilon_{z,n}$ defined in (A.3) of Appendix A will satisfy $\varepsilon_z \le \varepsilon_{z,n}$ for all $n > N$.

3.4 Convergence of the Approximating Sequence

This section culminates the analysis of the asymptotic behavior of the approximating Logit-MLE

procedure. The objective is to show that under the assumptions of §3.2 the approximating procedure will generate a sequence of estimates $\{w_n\}$ such that $w_n \to x^0$ almost surely.

Let $N \equiv \max\{N_1, N_2, N_3\}$ where the N_i are the threshold index values in assumptions (A5) - (A7) of §3.2. Let $N' \equiv \max\{N, n_E\}$ where n_E indexes the last trial of the exact phase of the approximating procedure. Upon completion of trial $n = N'$ the estimate of x^0 is $w_{N'+1}$. On all trials $n > N'$ the properties specified in assumptions (A1) - (A7) may be considered in force and sample points will be determined according to the approximation (2.4.2). It was argued in §2.4 that the approximation can be expressed in the form:

$$w_{n+1} = w_n - a_n((w_1,y_1),\ldots,(w_{n-1},y_{n-1})) \cdot (y_n - p^0) \quad . \tag{3.4.1}$$

The precise objective is to demonstrate that starting from the estimate $w_{N'+1}$ the sequence of estimates $\{w_n\}$ produced by the approximating phase of the procedure will converge (a.s.) to the root x^0 as the terminal index value n_T becomes arbitrarily large. In order to apply the result of Robbins and Siegmund (1971) detailed in §3.1 an equivalent problem will be considered, defined by translating the response variable y and the quantal response function $R(\cdot)$ as follows:

$$\begin{aligned} t_n &\equiv y_n - p^0 \\ M(w_n) &\equiv R(w_n) - p^0 \quad . \end{aligned} \tag{3.4.2}$$

Under this translation the iteration form (3.4.1) is then equivalent to:

$$\begin{aligned} w_{n+1} &= w_n - a_n((w_1,t_1+p^0),\ldots,(w_{n-1},t_{n-1}+p^0)) \cdot t_n \\ &\equiv w_n - a_n^*((w_1,t_1),\ldots,(w_{n-1},t_{n-1})) \cdot t_n \end{aligned} \tag{3.4.3}$$

which matches the generic form specified in (3.1.1). It will be demonstrated that this equivalent problem satisfies the remaining conditions of §3.1.

Under assumption (A3):

$$y_n \sim \text{BIN}(1,R(w_n)) \ .$$

From this and (3.4.2) it follows that:

$$E[t_n] = M(w_n)$$

$$\text{Var}[t_n] \equiv \sigma^2(w_n) = R(w_n)\cdot(1-R(w_n)) < 1$$

and thus (3.1.2) is satisfied.

The measurability requirements on $M(\cdot)$ and $\sigma(\cdot)$ will be satisfied due to continuity of these functions; both are continuous functions of $R(\cdot)$ which is continuous by assumption (A2) of §3.2. Moreover, $0 \le R(w_n) \le 1$ and $0 < p^0 < 1$ so:

$$|M(w_n)| < 1$$

and therefore (3.1.3) may be satisfied by choosing $a = 2$ with $b > 0$ arbitrary.

In assumption (A2) of §3.2 the function $R(\cdot)$ is also assumed to be strictly increasing with $R(x^0) = p^0$. Hence $M(\cdot)$ is also strictly increasing with $M(x^0) = 0$. Thus for any $0 < \varepsilon < 1$:

$$\inf_{\varepsilon < |w-x^0| < 1/\varepsilon} M(w)\cdot(w-x^0) \quad \geq \quad \min\{\, |M(x^0-\varepsilon)|\cdot\varepsilon\,,\, M(x^0+\varepsilon)\cdot\varepsilon\} \quad > \quad 0$$

so by letting $\theta = x^0$ it follows that (3.1.4) is satisfied.

The remaining conditions pertain to the sequence of multipliers $\{a_n^*\}$ for the approximating procedure, defined via the correspondence (3.4.3). Note that based on this correspondence:

$$a_n^*((w_1,t_1),\ldots,(w_{n-1},t_{n-1})) \quad \equiv \quad a_n((w_1,t_1+p^0),\ldots,(w_{n-1},t_{n-1}+p^0)) \quad \equiv \quad a_n((w_1,y_1),\ldots,(w_{n-1},y_{n-1}))$$

and therefore it is acceptable to work with the multipliers $\{a_n\}$ exactly as they are defined in the approximating procedure. These multipliers were given in (2.4.4) to have the form:

$$a_n \quad \equiv \quad (1/\hat{\beta}_{n-1})\cdot\sum_{i=1}^{n}(w_i - w_n)^2\cdot z_i \;/\; \det(\mathbf{H}_n)\ . \tag{3.4.4}$$

Under assumption (A6) the MLE $\hat{\beta}_{n-1}$ will be persistently positive for $n > N'$. Hence by the discussion at the end of §2.3 the multipliers $a_n(\cdot)$ will also be persistently positive for $n > N'$.

The measurability requirement for the multipliers $a_n(\cdot)$ may be addressed demonstrating continuity. Specifically, it needs to be established that $a_n(\cdot)$ is continuous in its arguments $w_1,y_1,\ldots,w_{n-1},y_{n-1}$. At the close of §2.2 an argument was presented that the MLEs $\hat{\mu}_{n-1}$ and $\hat{\beta}_{n-1}$ are continuous in $w_1,y_1,\ldots,w_{n-1},y_{n-1}$. The terms $(w_i - w_n)^2$ are clearly continuous in $w_1,\ldots,w_n$. From (2.3.6) it follows that each z_i for $i = 1,\ldots,n$ is continuous in its arguments w_i, $\hat{\mu}_{n-1}$, and $\hat{\beta}_{n-1}$. From the proof of Lemma A.2 in Appendix A it is evident that $\det(\mathbf{H}_n)$ is continuous in $w_1,\ldots,w_n$ and $z_1,\ldots,z_n$. Hence, if w_n is a continuous function of $w_1,y_1,\ldots,w_{n-1},y_{n-1}$ then $a_n(\cdot)$ is a composition of continuous functions of $w_1,y_1,\ldots,w_{n-1},y_{n-1}$ and therefore continuous in these arguments. It will now be argued by induction that this is indeed the case. Consider that the last estimate of x^0 provided by the exact phase of the approximating procedure is given by:

$$w_{n_E} = [\log(p^0) - \log(1-p^0) - \hat{\mu}_{n_E-1}] / \hat{\beta}_{n_E-1}$$

this being a continuous function of the MLEs which are themselves continuous in $w_1,y_1,\ldots,w_{n_E-1},y_{n_E-1}$. Now suppose that w_n is a continuous function of $w_1,y_1,\ldots,w_{n-1},y_{n-1}$ for arbitrary $n \geq n_E$. Then:

$$w_{n+1} = w_n - a_n \cdot (y_n - p^0)$$

with a_n as in (3.4.4). Clearly w_{n+1} is directly continuous in w_n, a_n, and y_n. But by the induction hypothesis and the discussion preceding it, a_n is a function of $w_1,y_1,\ldots,w_{n-1},y_{n-1}$ and is continuous in these arguments. Hence w_{n+1} is continuous in $w_1,y_1,\ldots,w_n,y_n$ and the induction argument is complete.

Finally, it must be shown that conditions (3.1.5) and (3.1.6) are satisfied. This involves demonstrating the existence of constants $0 < c' \leq c''$ such that the multipliers satisfy $c'/n \leq a_n(\cdot) \leq c''/n$ for all $n > N'$. First the lower bound will be established. Assuming that $n > N'$:

$$\begin{aligned}
a_n &\equiv (1/\hat{\beta}_{n-1}) \cdot \sum_{i=1}^{n} (w_i - w_n)^2 \cdot z_i \;/\; \det(\mathbf{H}_n) && \{ \text{ from (3.4.4) } \} \\
&\geq (1/\beta_{max}) \cdot \sum_{i=1}^{n} (w_i - w_n)^2 \cdot z_i \;/\; \det(\mathbf{H}_n) && \{ \text{ assumption (A6) } \} \\
&\geq (\varepsilon_z/\beta_{max}) \cdot \sum_{i=1}^{n} (w_i - w_n)^2 \;/\; \det(\mathbf{H}_n) && \{ \text{ invariant bound } \varepsilon_z \text{ of §3.3 } \} \\
&\geq (\varepsilon_z/\beta_{max}) \cdot \sum_{i=1}^{n} (w_i - \bar{w}_n)^2 \;/\; \det(\mathbf{H}_n) && \{ \bar{w}_n \text{ minimizes the SS } \} \\
&= (\varepsilon_z/\beta_{max}) \cdot n\hat{\sigma}_{w,n}^2 \;/\; \det(\mathbf{H}_n) && \{ \text{ definition of } \hat{\sigma}_{w,n}^2 \}
\end{aligned}$$

$$\geq \quad (\varepsilon_z / \beta_{max}) \cdot n \cdot \hat{\sigma}_{w,n}^2 \; / \; n^2 \cdot \hat{\sigma}_{w,n}^2 \qquad \{ \text{Lemma A.2} \}$$

$$= \quad (\varepsilon_z / \beta_{max}) / n \; .$$

Thus $c' = \varepsilon_z / \beta_{max}$ where $c' > 0$.

Next, the upper bound will be established. Assuming that $n > N'$:

$$a_n \quad \equiv \quad (1/\hat{\beta}_{n-1}) \cdot \sum_{i=1}^{n} (w_i - w_n)^2 \cdot z_i \; / \; \det(\mathbf{H}_n) \qquad \{ \text{from (3.4.4)} \}$$

$$\leq \quad (1/\beta_{min}) \cdot \sum_{i=1}^{n} (w_i - w_n)^2 \cdot z_i \; / \; \det(\mathbf{H}_n) \qquad \{ \text{assumption (A6)} \}$$

$$\leq \quad (1/\beta_{min}) \cdot \sum_{i=1}^{n} (w_i - w_n)^2 \; / \; \det(\mathbf{H}_n) \qquad \{ \; 0 < z_i < 1 \;\; \forall i \; \}$$

$$= \quad (1/\beta_{min}) \cdot \Big[\sum_{i=1}^{n} (w_i - \overline{w}_n)^2 \; + \; n \cdot (w_n - \overline{w}_n)^2 \Big] \; / \det(\mathbf{H}_n)$$

$$= \quad (1/\beta_{min}) \cdot n \cdot \big[\hat{\sigma}_{w,n}^2 \; + \; (w_n - \overline{w}_n)^2 \big] \; / \det(\mathbf{H}_n) \qquad \{ \text{definition of } \hat{\sigma}_{w,n}^2 \}$$

$$\leq \quad (1/\beta_{min}) \cdot n \cdot \big[\hat{\sigma}_{w,n}^2 \; + \; (w_n - \overline{w}_n)^2 \big] \; / \; \big[\varepsilon_{z,n}^2 \cdot n^2 \cdot \hat{\sigma}_{w,n}^2 \big] \qquad \{ \text{Lemma A.2} \}$$

$$= \quad \big[1/(\beta_{min} \cdot \varepsilon_{z,n}^2) \big] \cdot \big[\hat{\sigma}_{w,n}^2 \; + \; (w_n - \overline{w}_n)^2 \big] \; / \; \big[n \cdot \hat{\sigma}_{w,n}^2 \big]$$

$$= \quad \big[1/(\beta_{min} \cdot \varepsilon_{z,n}^2) \big] \cdot \big[1 \; + \; (w_n - \overline{w}_n)^2 / \hat{\sigma}_{w,n}^2 \big] \; / \; n$$

$$\leq \quad \big[1/(\beta_{min} \cdot \varepsilon_{z,n}^2) \big] \cdot \big[1 \; + \; M^2 \big] \; / \; n \qquad \{ \text{assumption (A7)} \}$$

$$\leq \quad [1/(\beta_{min}\cdot\varepsilon_z^{\,2})]\cdot[1 + M^2] \;/\; n \qquad \{ \text{ invariant } \varepsilon_z \text{ of } \S 3.3 \}$$

$$= \quad [(1+M^2) / (\beta_{min}\cdot\varepsilon_z^{\,2})] \;/\; n \;.$$

Thus $c'' = (1 + M^2) / (\beta_{min}\cdot\varepsilon_z^{\,2})$.

Since $0 < c'/n \leq a_n(\cdot) \leq c''/n$ for all $n > N'$ then due to familiar results associated with the series $\sum 1/n$ and the series $\sum 1/n^2$ it follows that:

$$\sum_{n=1}^{\infty} a_n(\cdot) \quad = \quad \infty \,,$$

$$\sum_{n=1}^{\infty} a_n^{\,2}(\cdot) \quad < \quad \infty \,.$$

Hence conditions (3.1.5) and (3.1.6) are met and it is proven that $w_n \rightarrow \theta = x^0$ a.s. as $n_T \rightarrow \infty$.

CHAPTER 4

A SIMULATION STUDY

The paper by Wu (1985) introducing the Logit-MLE procedure and the present dissertation both offer a convergence proof based on some form of first-order approximation. The key idea common to both is that eventually the true step size (2.3.1) becomes almost linear in the quantity $y_n - p^0$, that is:

$$\Delta x_{n+1} \quad \cong \quad -a_n \cdot (y_n - p^0) \tag{4.1}$$

where a_n is adaptive. There is, however, a critical distinction between the formulations of the two approximations. The approximation (1.3.4) formulated by Wu calls for the prior iterates $x_1, \ldots, x_n$ to lie in close proximity. It does not support an explanation of how such a state is reached. The approximation (2.4.2) formulated here does not rely on such a proximity assumption, thus providing a more reasonable means by which to argue convergence.

In §4.1 the basic accuracy of the first-order approximations will be evaluated by studying the relative error:

$$| -a_n \cdot (y_n - p^0) \quad - \quad \Delta x_{n+1} | \quad / \quad | \Delta x_{n+1} | \tag{4.2}$$

arising out of the competing approximation schemes during simulated experiments.

4.1 Relative Error Analysis

A desirable property for either first-order approximation is that greater accuracy be realized as the trial count n increases. Sequential experiments may be simulated in order to evaluate the accuracy of the approximations in n, but when implemented in the usual way such simulations tend to generate sequences of estimates which gradually cluster. Thus, an (apparently) increasing accuracy is likely to be confounded

with the process of convergence. We would like to know whether accuracy increases independent of convergence, and this can be evaluated across simulation runs involving various parameter settings.

Each simulation run will consist of some number of startup trials distributed evenly over fixed levels of x, followed by some number of sequential trials carried out as prescribed by the adaptive Logit-MLE procedure. Let the sequence of iterates generated up to this point be denoted by $x_1 , \ldots , x_n$. An identical sequence $w_1 , \ldots , w_n$ will be maintained in parallel to $x_1 , \ldots , x_n$ such that $w_1 = x_1 , \ldots , w_n = x_n$. The next iterate x_{n+1} will again be generated using the Logit-MLE procedure but the next iterate w_{n+1} will be determined using a first-order approximation. Wu's approximation is calculated by applying (1.3.4) to the data $(w_1,y_1) , \ldots , (w_n,y_n)$. The competing approximation is calculated by applying (2.4.2) to the same data. Whichever approximation is used, the relative error measure is:

$$\text{r.e.} \quad = \quad | x_{n+1} - w_{n+1} | \quad / \quad | x_{n+1} - x_n | \tag{4.1.1}$$

which is equivalent to (4.2). To maintain consistency with the approximating procedure of §2.4 the same trial index definitions will be employed here. For example, in a simulated experiment involving 40 startup trials, followed by 10 sequential trials carried out using the Logit-MLE iteration, followed by one trial with the Logit-MLE iteration and the approximation carried out in parallel, the appropriate index settings would be:

$$n_I = 40 ,$$

$$n_E = 50 ,$$

$$n_T = 51 .$$

4.1.1 Simulation Setup

Simulated binary responses will be generated using streams of IID U(0,1) pseudo-random variates and a piecewise-linear quantal response function $R(\cdot)$. The response function $R(\cdot)$ is graphed in Figure 4.1

below with important reference points labeled on the axes.

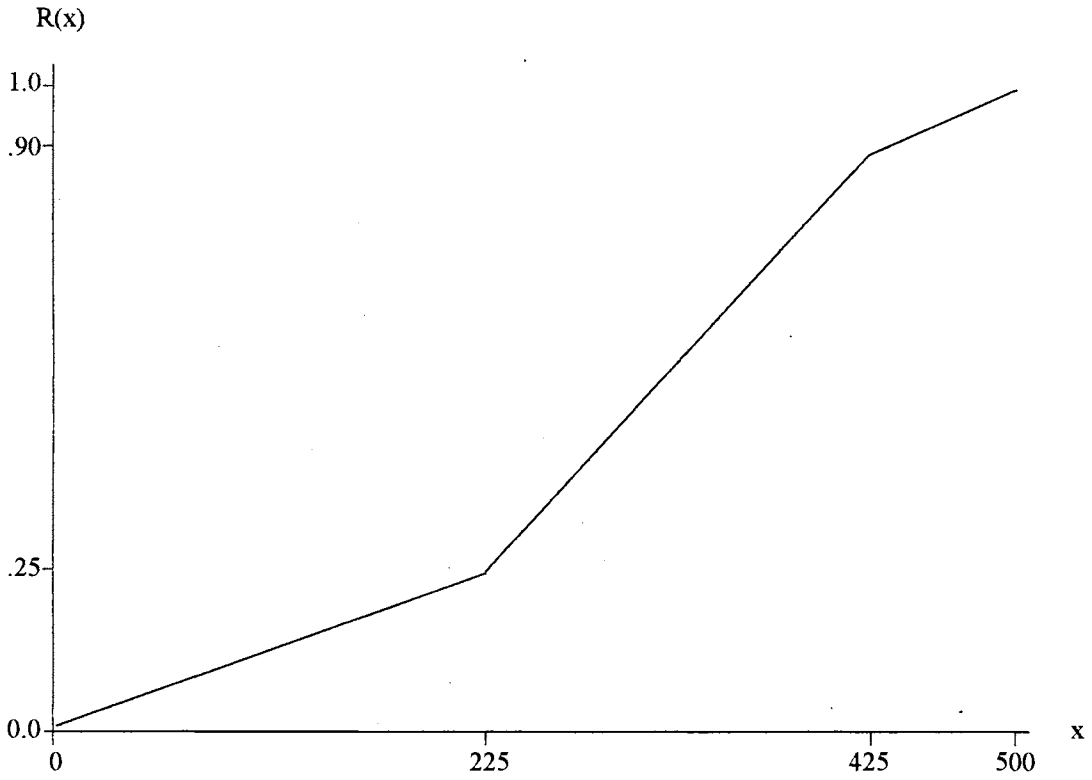

Figure 4.1 Quantal Response Curve for Simulations

This form of quantal response function was selected for two reasons. First, it allows root estimation to be simulated for a target point x^0 where the function is differentiable or at a point where it is nondifferentiable. Second, it is different enough from the shape of a two-parameter logit function that the simulation results may reasonably be considered independent of correct model specification.

For a given simulation run let $\{u_n\}$ denote a stream of IID U(0,1) pseudo-random variates. On the n^{th} trial of the experiment the value u_n represents the "response profile" of the current experimental unit. The simulated binary response y_n associated with a control variable setting of $x = x_n$ will then be generated in the usual way:

$$y_n = \begin{bmatrix} 1 & , & u_n \le R(x_n) \\ 0 & , & \text{otherwise} \end{bmatrix}. \qquad (4.1.2)$$

The startup trials for each simulated experiment will be evenly allocated over the control variable settings x = 100, x = 200, x = 300, x = 400. If the response pattern for the startup phase fails to satisfy the existence and uniqueness conditions (described in §1.3.1) for the MLEs of the logit function parameters the simulation run will be abandoned.

4.1.2 Simulation Results

Table 4.2 below displays the relative error analysis for various settings of the index limits n_I, n_E, n_T. The estimation target was x^0 corresponding to $p^0 = 0.25$. (Note incidentally that the quantal response function is not differentiable at this point.) Each table entry resulted from N = 10,000 simulation runs. All of the simulation runs involved $n_I = 40$ startup trials. The number of subsequent Logit-MLE sequential trials was 1, 10, 60, 160, and 460 in ascending order for the five different index settings.

	Approximation (2.4.2)		Approximation (1.3.4)	
n_I, n_E, n_T	average	worst-case	average	worst-case
40, 41, 42	.01180	.33435	.29883	.69545
40, 50, 51	.00627	.25839	.25032	.34165
40, 100, 101	.00302	.05996	.14395	.46144
40, 200, 201	.00234	.04589	.10245	.86573
40, 500, 501	.00170	.03802	.07475	1.38266

Table 4.2 Relative Error of Approximation with Sequences Converging ($p^0 = 0.25$)

The results in Table 4.2 suggest that both approximations improve on average as the trial count increases. In the worst-case analysis only approximation (2.4.2) improves. (At least one case was encountered where approximation (1.3.4) yielded a directional error.) The sequences generated in these simulations were permitted to naturally converge as the trial count increased. The gradual improvement in approximation accuracy is therefore potentially confounded with simultaneous convergence. The results reported in Table 4.3 below address the issue of whether accuracy improves due to amassing data or due to sample points gradually clustering. The simulation runs summarized in this table involved the same total trial counts as the simulations reported in the last four lines of Table 4.2 but with nearly all trials allocated to the startup phase, thereby artificially forcing dispersion among the x-values used to calculate the approximations.

	Approximation (2.4.2)		Approximation (1.3.4)	
n_I, n_E, n_T	average	worst-case	average	worst-case
49, 50, 51	.00782	.22271	.30345	.51591
99, 100, 101	.00262	.03435	.31575	.46160
199, 200, 201	.00099	.01608	.32737	.42444
499, 500, 501	.00030	.00334	.33509	.40733

Table 4.3 Relative Error of Approximation with Sample Points Dispersed ($p^0 = 0.25$)

Although such an allocation would not be used in practice, it does strongly suggest that the accuracy of approximation (2.4.2) is attributable primarily to an increasing trial count. Approximation (1.3.4) shows essentially no improvement in n under these conditions, which is consistent with its reliance on close proximity of sample points.

Tables 4.4 and 4.5 parallel the analyses of Tables 4.2 and 4.3, respectively, with the estimation target switched to x^0 corresponding to $p^0 = 0.10$. The quantal response function is differentiable at this point and moreover the estimation objective tends to the extreme. The conclusions with respect to approximation (2.4.2) do not change appreciably. However, approximation (1.3.4) shows no improvement in this setting even when the sequences are permitted to converge.

	Approximation (2.4.2)		Approximation (1.3.4)	
n_I, n_E, n_T	average	worst-case	average	worst-case
40, 41, 42	.01281	.25105	.44298	.70395
40, 50, 51	.00582	.19320	.46984	.64720
40, 100, 101	.00214	.06540	.47871	.60478
40, 200, 201	.00128	.04399	.48006	2.23299
40, 500, 501	.00069	.02671	.47793	2.20608

Table 4.4 Relative Error of Approximation with Sequences Converging ($p^0 = 0.10$)

	Approximation (2.4.2)		Approximation (1.3.4)	
n_I, n_E, n_T	average	worst-case	average	worst-case
49, 50, 51	.00893	.15153	.44197	.71034
99, 100, 101	.00353	.06877	.43617	.70772
199, 200, 201	.00178	.03593	.43424	.64860
499, 500, 501	.00074	.01067	.43353	.58804

Table 4.5 Relative Error of Approximation with Sample Points Dispersed ($p^0 = 0.10$)

4.1.3 Simulation Program

All of the simulations reported in §4.1.2 were carried out using the C++ program listed in Appendix B with the parameters appearing in the first section of the program properly set. The program implements a "shuffle" generator to produce the U(0,1) pseudo-random streams $\{u_n\}$. In all cases identical seeds were used to initialize the streams for the simulations comparing approximations (2.4.2) and (1.3.4).

CHAPTER 5

SUMMARY OF FINDINGS

A first-order approximation to the movement of the adaptive Logit-MLE procedure may be formulated with few special assumptions beyond those usually made in the quantal response setting. An estimation procedure based on this first-order approximation converges with probability one to the correct value x^0 under mild assumptions, which may be incorporated into the estimation procedure if desired.

In simulated experiments, the relative error of the first-order approximation decreases steadily as the trial count n increases, independent of whether the sequence of estimates converges or is dispersed. This empirical result in combination with the proven convergence of the first-order procedure provides a strong argument for convergence of the true Logit-MLE procedure.

REFERENCES

Agresti, A. (1990). *Categorical Data Analysis*. New York: Wiley.

Anbar, D. (1978). A stochastic Newton-Raphson method. *Journal of Statistical Planning and Inference* **2**: 153-163.

Blum, J.R. (1954). Approximation methods which converge with probability one. *Annals of Mathematical Statistics* **25**: 382-386.

Chung, K.L. (1954). On a stochastic approximation method. *Annals of Mathematical Statistics* **25**: 463-483.

Moser, B.K. and D. Faries (1996). A new sequential approximation method. *Journal of Statistical Planning and Inference* **55**: 175-189.

Robbins, H. and S. Monro (1951). A stochastic approximation method. *Annals of Mathematical Statistics* **22**: 400-407.

Robbins, H. and D. Siegmund (1971). A convergence theorem for nonnegative almost supermartingales and some applications. *Optimizing Methods in Statistics*. New York: Academic Press.

Sacks, J. (1958). Asymptotic distribution of stochastic approximation procedures. *Annals of Mathematical Statistics* **29**: 373-405.

Shen, L.Z. and J. O'Quigley (1996). Consistency of continual reassessment method under model misspecification. *Biometrika* **83**: 395-405.

Silvapulle, M.J. (1981). On the existence of maximum likelihood estimators for the binomial response models. *Journal of the Royal Statistical Society* **B43**: 310-313.

Venter, J.H. (1967). An extension of the Robbins-Monro procedure. *Annals of Mathematical Statistics* **38**: 181-190.

Wolfowitz, J. (1952). On the stochastic approximation method of Robbins and Monro. *Annals of Mathematical Statistics* **23**: 457-461.

Wu, C.F.J. (1985). Efficient sequential designs with binary data. *Journal of the American Statistical Association* **80**: 974-984.

APPENDIX A

THE LOG-LIKELIHOOD HESSIAN

Several properties of the Hessian matrix associated with the log-likelihood function (2.2.1) are established in this appendix. A simple derivation shows that the first-order partial derivatives of the two-parameter logit function (1.2.22) wrt μ and β are:

$$\partial\, \psi(x;\mu,\beta) / \partial\mu \;=\; \psi(x;\mu,\beta) \cdot [\, 1 - \psi(x;\mu,\beta)]$$

$$\partial\, \psi(x;\mu,\beta) / \partial\beta \;=\; x \cdot \psi(x;\mu,\beta) \cdot [\, 1 - \psi(x;\mu,\beta)] \;.$$

Applying these results in conjunction with the first-order partials (2.2.2) of the log-likelihood function, the following second-order partial derivatives can be quickly determined:

$$\partial^2 \log \mathcal{L}_n(\mu,\beta) / \partial\mu^2 \;=\; -\sum_{i=1}^{n} \psi(x_i;\mu,\beta) \cdot [\, 1 - \psi(x_i;\mu,\beta)]$$

$$\partial^2 \log \mathcal{L}_n(\mu,\beta) / \partial\mu\partial\beta \;=\; -\sum_{i=1}^{n} x_i \cdot \psi(x_i;\mu,\beta) \cdot [\, 1 - \psi(x_i;\mu,\beta)]$$

$$\partial^2 \log \mathcal{L}_n(\mu,\beta) / \partial\beta^2 \;=\; -\sum_{i=1}^{n} x_i^2 \cdot \psi(x_i;\mu,\beta) \cdot [\, 1 - \psi(x_i;\mu,\beta)] \;.$$

Adopting the shorthand notation of §2.3 let:

$$z_i \;\equiv\; z(x_i;\mu,\beta) \;\equiv\; \psi(x_i;\mu,\beta) \cdot [1 - \psi(x_i;\mu,\beta)] \;.$$

Note that for any finite arguments $0 < z_i < 1$.

The Hessian matrix associated with the log-likelihood function $\log \mathcal{L}_n(\mu,\beta)$ may then be expressed as:

$$\mathbf{H}_n \equiv \mathbf{H}(x_1,\ldots,x_n,\mu_{n-1},\beta_{n-1}) \equiv \begin{bmatrix} -\sum_{i=1}^{n} z_i & -\sum_{i=1}^{n} x_i \cdot z_i \\ -\sum_{i=1}^{n} x_i \cdot z_i & -\sum_{i=1}^{n} x_i^2 \cdot z_i \end{bmatrix}.$$

Two useful results will now be stated and proved.

Lemma A.1

Given sample points $x_1,\ldots,x_n$ such that not all x_i are equal, the matrix $\mathbf{H}(x_1,\ldots,x_n,\mu,\beta)$ is negative definite.

Proof

We may equivalently demonstrate that $\mathbf{A} \equiv -\mathbf{H}(x_1,\ldots,x_n,\mu,\beta)$ is positive definite.

Clearly the leading submatrix $\mathbf{A}_1 = \sum_{i=1}^{n} z_i$ has a positive determinant.

It remains to show that $\det(\mathbf{A}) > 0$. Note that:

$$\det(\mathbf{A}) = \left[\sum_{i=1}^{n} x_i^2 \cdot z_i\right]\left[\sum_{i=1}^{n} z_i\right] - \left[\sum_{i=1}^{n} x_i \cdot z_i\right]^2.$$

Rewriting:

$$\det(\mathbf{A}) = \left[\sum_{i=1}^{n} x_i^2 \cdot z_i\right]\left[\sum_{j=1}^{n} z_j\right] - \left[\sum_{i=1}^{n} x_i \cdot z_i\right]\left[\sum_{j=1}^{n} x_j \cdot z_j\right]$$

$$= \sum_{i=1}^{n}\sum_{j=1}^{n} x_i^2 \cdot z_i \cdot z_j - \sum_{i=1}^{n}\sum_{j=1}^{n} x_i \cdot z_i \cdot x_j \cdot z_j$$

$$= \sum_{i=1}^{n}\sum_{j=1}^{n} x_i \cdot (x_i - x_j) \cdot z_i \cdot z_j$$

$$= \sum_{i<j}^{n}\sum^{n} [x_i \cdot (x_i - x_j) \cdot z_i \cdot z_j + x_j \cdot (x_j - x_i) \cdot z_j \cdot z_i]$$

$$= \sum_{i<j}^{n}\sum^{n} [x_i \cdot (x_i - x_j) \cdot z_i \cdot z_j - x_j \cdot (x_i - x_j) \cdot z_i \cdot z_j]$$

$$= \sum_{i<j}^{n}\sum^{n} (x_i - x_j)^2 \cdot z_i \cdot z_j \ .$$

Since $x_i \neq x_j$ for some $i \neq j$ and since all products $z_i \cdot z_j > 0$, it follows that $\det(\mathbf{A}) > 0$. Therefore $\mathbf{A}$ is positive definite. ■

The next lemma requires some further definitions. Given the sample points $x_1,\ldots,x_n$ recall that the *empirical mean* and *empirical variance* are defined respectively as:

$$\bar{x}_n \equiv \sum_{i=1}^{n} x_i / n \ , \tag{A.1}$$

$$\hat{\sigma}_{x,n}^2 \equiv \sum_{i=1}^{n} (x_i - \bar{x}_n)^2 / n \ . \tag{A.2}$$

Given $x_1,\ldots,x_n$ and any fixed μ and β also define:

$$\varepsilon_{z,n} \equiv \min_i \{ z_i \} \equiv \min_i \{ z(x_i;\mu,\beta) \} \qquad (A.3)$$

noting that $\varepsilon_{z,n} > 0$.

Lemma A.2

Let $\mathbf{H} \equiv \mathbf{H}(x_1,\ldots,x_n,\mu,\beta)$. Then $\varepsilon_{z,n}^2 \cdot n^2 \cdot \hat{\sigma}_{x,n}^2 \leq \det(\mathbf{H}) < n^2 \cdot \hat{\sigma}_{x,n}^2$.

Proof

Since $\mathbf{H}$ is 2×2 then $\det(\mathbf{H}) = \det(-\mathbf{H})$. In the proof of Lemma A.1 note that $\det(-\mathbf{H}) = \det(\mathbf{A})$. Thus:

$$\det(\mathbf{H}) = \sum_{i<j}^{n\ n} (x_i - x_j)^2 \cdot z_i \cdot z_j$$

Since $0 < \varepsilon_{z,n} \leq z_i < 1$ for $i = 1,\ldots,n$ then:

$$\varepsilon_{z,n}^2 \cdot \sum_{i<j}^{n\ n} (x_i - x_j)^2 \leq \det(\mathbf{H}) < \sum_{i<j}^{n\ n} (x_i - x_j)^2 . \qquad (A.4)$$

By symmetry:

$$\sum_{i<j}^{n\ n} (x_i - x_j)^2 = (1/2) \sum_{i=1}^{n} \sum_{j=1}^{n} (x_i - x_j)^2$$

$$= (1/2) \sum_{i=1}^{n} \sum_{j=1}^{n} [(x_i - \bar{x}_n) - (x_j - \bar{x}_n)]^2$$

$$= (1/2) \sum_{i=1}^{n} \sum_{j=1}^{n} [(x_i-\bar{x}_n)^2 + (x_j-\bar{x}_n)^2 - 2(x_i-\bar{x}_n)(x_j-\bar{x}_n)]$$

$$= (1/2) [\, n \sum_{i=1}^{n} (x_i-\bar{x}_n)^2 + n \sum_{j=1}^{n} (x_j-\bar{x}_n)^2 - 2 \sum_{i=1}^{n} (x_i-\bar{x}_n) \sum_{j=1}^{n} (x_j-\bar{x}_n)] \quad (\text{the last sum} = 0)$$

$$= (1/2) [\, n \cdot n \hat{\sigma}_{x,n}^2 + n \cdot n \hat{\sigma}_{x,n}^2 \,]$$

$$= (1/2) [\, 2n^2 \cdot \hat{\sigma}_{x,n}^2 \,]$$

$$= n^2 \cdot \hat{\sigma}_{x,n}^2 \,.$$

Substituting this result into (A.4) gives:

$$\varepsilon_{z,n}^2 \cdot n^2 \cdot \hat{\sigma}_{x,n}^2 \;\leq\; \det(\mathbf{H}) \;<\; n^2 \cdot \hat{\sigma}_{x,n}^2 \,. \quad \blacksquare$$

APPENDIX B

SIMULATION PROGRAM

```
/*=========================================================================*/
/* PROGRAM ENVIRONMENT SECTION                                             */
/*=========================================================================*/

#include <iostream.h>
#include <iomanip.h>
#include <math.h>
#include <values.h>




/*=========================================================================*/
/* PROGRAM CONSTANTS                                                       */
/*=========================================================================*/

const int    NRUNS =  10000;  // number of simulations to run

const int    NS    =     40;  // terminal trial index for startup phase

const int    NSDX  =      4;  // number of distinct x-values in startup phase

const int    NE    =     41;  // terminal trial index for exact phase

const int    NT    =     42;  // terminal trial index

const double XMAX  =  500.0;  // defines effective range for x

const double P0    =   0.25;  // target "success" probability

const double EPS   = 1.0e-8;  // general ε for computational precision

const char   ATYPE =    'N';  // switches type of first-order approximation:
                              //  ■ 'N' = Newton-Raphson first step
                              //  ■ 'W' = Wu



/*=========================================================================*/
/* MATHEMATICAL/SIMULATION ROUTINES                                        */
/*=========================================================================*/

/*-------------------------------------------------------------------------*/
/* This function returns the probability of success "p" corresponding to   */
/* a given value of the control variable "x" based on a piecewise-linear   */
/* quantal response function.  (This is the data generator.)               */
/*-------------------------------------------------------------------------*/

double Quantal(double x)
 {

  /* Establish breakpoints of the piecewise-linear quantal response curve: */

  const int  NBP = 4;

  double     RCx[NBP] = {   0, .45*XMAX, .85*XMAX, XMAX};
```

```
  double     RCp[NBP] = {0.00,     0.25,     0.90, 1.00};

  /* Declare other variables: */

  int       i;
  double    p;

  /* Calculate the probability of success at x: */

  if (x <= RCx[0])
    {
     p = RCp[0];
    }
  else if (x >= RCx[NBP-1])
    {
     p = RCp[NBP-1];
    }
  else
    {
     i = 0;
     while (x > RCx[i+1]) i++;
     p = RCp[i]  +  ((x - RCx[i]) / (RCx[i+1] - RCx[i])) * (RCp[i+1] -
RCp[i]);
    }

  return p;

 }


/*-------------------------------------------------------------------------*/
/* This function returns the probability of success "p" corresponding to   */
/* a given value of the control variable "x" using a two-parameter logit   */
/* curve to model the quantal response function.                           */
/*-------------------------------------------------------------------------*/

double Logit(double mu, double beta, double x)
 {

  double z, p;

  /* Calculate the exponent: */

  z = mu + beta*x;

  /* Calculate the probability of success at x: */

  if (z < -500.0)
    {
     p = 0.0;
    }
  else if (z > 500.0)
    {
     p = 1.0;
    }
  else
    {
     p =  exp(z) / (1.0 + exp(z));
    }

  return p;
```

```
}

/*---------------------------------------------------------------------------*/
/* This function implements a U(0,1) PRNG based on a shuffling algorithm.    */
/*                                                                           */
/* Argument descriptions:                                                    */
/*                                                                           */
/* ■ seeda -- generator seed "a" to be set initially to an integer value     */
/*            in the range [1.0,2147483646.0] and thereafter not changed     */
/*            by the user                                                    */
/* ■ seedb -- generator seed "b" to be set initially to an integer value     */
/*            in the range [1.0,2147483710.0] and thereafter not changed     */
/*            by the user                                                    */
/* ■ reset -- initialization flag to be set to 1 on the first call           */
/*---------------------------------------------------------------------------*/

double Uniform(double& seeda, double& seedb, int& reset)
 {

  const double  MODA = 2147483647.0;
  const double  MODB = 2147483711.0;
  const double  MULT = 16807.0;

  const int     TABLE_SIZE = 128;

  static double table[TABLE_SIZE];

  int           j;

  double        u;


  /* Initialize the shuffle table on the first call: */

  if (reset)
    {
     for (j = 0; j < TABLE_SIZE; j++)
        {
         seeda    = fmod(MULT*seeda, MODA);
         seedb    = fmod(MULT*seedb, MODB);
         table[j] = (seeda + seedb/MODB) / MODA;
        }
     reset = 0;
    }


  /* Randomly draw and regenerate shuffle table entries */
  /* until an entry on the interval (0,1) is extracted: */

  do
    {
     /* Draw an element from the table at random: */

     do
       {
        seeda = fmod(MULT*seeda, MODA);
        seedb = fmod(MULT*seedb, MODB);
        j     = int(fmod(seedb, double(TABLE_SIZE)));
       }
     while ((j < 0)  ||  (j >= TABLE_SIZE));

     u = table[j];

     /* Replace the table element with a new value: */
```

```
      table[j] = (seeda + seedb/MODB) / MODA;
    }
  while ((u <= 0.0)  ||  (u >= 1.0));

  /* Return the new U(0,1) PRN: */

  return u;

 }



/*=========================================================================*/
/* ESTIMATION ROUTINES                                                     */
/*=========================================================================*/

/*-------------------------------------------------------------------------*/
/* Based on a sequence of observed (x,y) pairs this function determines if */
/* the MLEs exist for the parameters (μ,ß) in a two-parameter logit model. */
/*-------------------------------------------------------------------------*/

int MLEs_Exist(double x[], double y[], int n)
 {

  double   minimum_0, maximum_0, minimum_1, maximum_1;

  int      i;


  /* Determine the x-range over which "0" responses have occurred and the */
  /* x-range over which "1" responses have occurred:                      */

  minimum_0  =  MAXDOUBLE;
  maximum_0  =  MINDOUBLE;
  minimum_1  =  MAXDOUBLE;
  maximum_1  =  MINDOUBLE;

  for (i = 0; i < n; i++)
     {
      if (y[i] == 0)
        {
         minimum_0  =  x[i] < minimum_0  ?  x[i] : minimum_0;
         maximum_0  =  x[i] > maximum_0  ?  x[i] : maximum_0;
        }
      else
        {
         minimum_1  =  x[i] < minimum_1  ?  x[i] : minimum_1;
         maximum_1  =  x[i] > maximum_1  ?  x[i] : maximum_1;
        }
     }


  /* Check whether the MLE existence conditions are met: */

  if (maximum_0 <= minimum_1)
    {
     return 0;
    }
  else if (maximum_1 <= minimum_0)
    {
     return 0;
    }
  else if ((minimum_0 == maximum_0) && (minimum_1 == maximum_1))
    {
```

```
    return 0;
   }
  else
   {
    return 1;
   }

}



/*-------------------------------------------------------------------------*/
/* This function implements the Newton-Raphson procedure to find the MLEs  */
/* of the parameters characterizing a two-parameter logit curve, based on  */
/* a sequence of observed (x,y) pairs.  It is assumed that the MLEs exist; */
/* this can be verified using the function 'MLEs_Exist' defined elsewhere  */
/* in this program.                                                        */
/*                                                                         */
/* The starting values for the MLEs are set externally.  This allows MLEs  */
/* from earlier calculations to be used as the starting values, generally  */
/* resulting in quicker convergence.                                       */
/*-------------------------------------------------------------------------*/

void MLEs(double x[], double y[], int n, double& mu, double& beta)
 {

  int         i;

  double      p, z, g1, g2, H11, H12, H22, detH, D1, D2;


  /* Calculate the MLEs: */

  do
    {
     g1  = 0.0;
     g2  = 0.0;
     H11 = 0.0;
     H12 = 0.0;
     H22 = 0.0;

     for (i = 0; i < n; i++)
        {
         p     =  Logit(mu, beta, x[i]);

         g1   +=  (y[i]-p);
         g2   +=  x[i]*(y[i]-p);

         z     =  p*(1.0-p);

         H11  -=  z;
         H12  -=  x[i]*z;
         H22  -=  x[i]*x[i]*z;
        }

     detH  = H11*H22 - H12*H12;

     D1    = g1*H22 - g2*H12;
     D2    = g2*H11 - g1*H12;

     mu   -= D1/detH;
     beta -= D2/detH;
    }
  while (g1*g1 + g2*g2 > EPS);

 }
```

```
/*--------------------------------------------------------------------------*/
/* This routine determines the next sample point x  based on the adaptive  */
/* Logit-MLE procedure.                            n                        */
/*--------------------------------------------------------------------------*/

void Logit_MLE(double x[], double y[], int n)
 {

  double  mu, beta;


  /* Calculate the MLEs based on currently available data: */

  mu   = 0.0;
  beta = 0.0;

  MLEs(x, y, n, mu, beta);


  /* Calculate the next adaptive Logit-MLE sample point: */

  if (fabs(beta) > EPS)
    {
     x[n] = (log(P0) - log(1.0 - P0) - mu) / beta;
    }
  else
    {
     x[n] = x[n-1];
    }

 }



/*--------------------------------------------------------------------------*/
/* This routine determines the next sample point w  using a Newton-Raphson */
/* first-step approximation.                       n                        */
/*--------------------------------------------------------------------------*/

void NR_First_Step(double w[], double y[], int n)
 {

  double  mu_prior, beta_prior;

  int     i;

  double  p, z, S, H11, H12, H22, detH, a;


  /* Calculate the lagged MLEs based on currently available data: */

  mu_prior   = 0.0;
  beta_prior = 0.0;

  MLEs(w, y, n-1, mu_prior, beta_prior);


  /* Determine the next sample point: */

  if (fabs(beta_prior) > EPS)
    {
     /* Calculate the coefficient 'a': */

     S   = 0.0;
```

```
      H11 = 0.0;
      H12 = 0.0;
      H22 = 0.0;

      for (i = 0; i < n; i++)
         {
          p    = Logit(mu_prior, beta_prior, w[i]);
          z    = p*(1.0-p);
          S   += (w[i]-w[n-1])*(w[i]-w[n-1]) * z;
          H11 -= z;
          H12 -= w[i]*z;
          H22 -= w[i]*w[i]*z;
         }

      detH = H11*H22 - H12*H12;

      a    = (1.0 / beta_prior) * (S / detH);

      /* Calculate the sample point: */

      w[n] = w[n-1] - a*(y[n-1] - P0);
     }

   else
     {
      w[n] = w[n-1];
     }

 }


/*-------------------------------------------------------------------------*/
/* This routine determines the next sample point w  using the first-order  */
/* approximation of Wu.                          n                         */
/*-------------------------------------------------------------------------*/

void Wu_First_Order(double w[], double y[], int n)
 {

  int            i;

  double         sumw, sumy, sumwy, sumsq, c;


  /* Calculate the coefficient c: */

  sumw  = 0.0;
  sumy  = 0.0;
  sumwy = 0.0;
  sumsq = 0.0;

  for (i = 0; i < n; i++)
     {
      sumw  += w[i];
      sumy  += y[i];
      sumwy += w[i]*y[i];
      sumsq += (w[i]-w[n-1])*(w[i]-w[n-1]);
     }

  c = sumsq / (sumwy - sumw*sumy/n);


  /* Calculate the sample point: */

  w[n] = w[n-1] - (c/n)*(y[n-1] - P0);
```

```
}


/*-------------------------------------------------------------------------*/
/* This routine determines the next sample point w  using the specified    */
/* type of first-order approximation.              n                       */
/*-------------------------------------------------------------------------*/

void Approximate(double w[], double y[], int n, char type)
 {

  switch (type)
        {
         case 'N': /* Newton-Raphson first-step approximation: */
                   NR_First_Step(w,y,n);
                   break;

         case 'W': /* Wu's first-order approximation: */
                   Wu_First_Order(w,y,n);
                   break;

         default:  w[n] = w[n-1];
                   break;
        }

 }




/*=========================================================================*/
/* OUTPUT ROUTINES                                                         */
/*=========================================================================*/

/*-------------------------------------------------------------------------*/
/* This routine prints a heading for a table of simulation run summaries.  */
/*-------------------------------------------------------------------------*/

void Print_Heading(void)
 {

  cout << "    ** SIMULATIONS FOR ESTIMATION OF L("
       << setprecision(2)
       << setw(4)
       << P0
       << ") **"
       << endl
       << endl;

  cout << "Run #    Error/Movement" << endl;
  cout << "-----    --------------" << endl;

 }


/*-------------------------------------------------------------------------*/
/* This routine prints a footing for a table of simulation run summaries.  */
/*-------------------------------------------------------------------------*/

void Print_Footing(double avg_error, double max_error)
 {

  cout << endl
       << endl
```

```
        << "    AVERAGE ERROR WAS "
        << setprecision(5)
        << setw(8)
        << avg_error
        << endl
        << "    HIGHEST ERROR WAS "
        << setprecision(5)
        << setw(8)
        << max_error
        << endl;

 }


/*-------------------------------------------------------------------------*/
/* This routine prints a summary line for one simulation run.              */
/*-------------------------------------------------------------------------*/

void Run_Summary(int run, double error)
 {

  cout << setiosflags(ios::fixed)
       << setiosflags(ios::showpoint)
       << setw(5)
       << run
       << "       "
       << setprecision(5)
       << setw(8)
       << error
       << endl;

 }




/*=========================================================================*/
/* MAIN PROGRAM                                                            */
/*=========================================================================*/

void main()
 {

  /* Arrays for tallying trial outcomes for Logit-MLE procedure: */

  double x[NT], ye[NT];


  /* Arrays for tallying trial outcomes for approximating procedure: */

  double w[NT], ya[NT];


  /* Simulation run counter: */

  int run;


  /* Completed simulation run counter: */

  int completed;


  /* Trial index: */
```

```
int trial;


/* Quantal response probability variable: */

double p;


/* PRNG variables: */

double seeda, seedb, u;
int    reset;


/* Approximation error measurement variables: */

double minimum_x, maximum_x, x_movement;
double error, sum_error, max_error;



/* Initialize the PRNG once at the beginning of the entire simulation: */

seeda = 1247483131.0;
seedb =  951711238.0;
reset = 1;


/* Initialize the overall error measurements: */

sum_error = 0.0;
max_error = 0.0;


/* Print the simulation table heading: */

Print_Heading();


/* Initialize the count of completed simulation runs: */

completed = 0;


/* Perform simulation runs: */

for (run = 1; run <= NRUNS; run++)
   {
    /* Initialize the trial index: */

    trial = 0;


    /* Perform startup trials.  Startup trials are evenly allocated */
    /* across NSDX sample points equally spaced between 0 and XMAX: */

    while (trial < NS)
         {
          x[trial]  = (trial/(NS/NSDX) + 1) * XMAX/(NSDX+1);
          w[trial]  = x[trial];
          p         = Quantal(x[trial]);
          u         = Uniform(seeda,seedb,reset);
          ye[trial] = (u < p)  ?  1 : 0;
          ya[trial] = ye[trial];
          trial++;
         }
```

```
/* If the MLEs don't exist after startup abandon the simulation run: */

if (!MLEs_Exist(x,ye,trial))
  {
   cout << "Startup phase did not yield MLEs -- run abandoned." << endl;
   continue;
  }


/* Perform trials for the exact sequential phase: */

while (trial < NE)
     {
      Logit_MLE(x,ye,trial);
      w[trial]  = x[trial];
      p         = Quantal(x[trial]);
      u         = Uniform(seeda,seedb,reset);
      ye[trial] = (u < p)  ?  1 : 0;
      ya[trial] = ye[trial];
      trial++;
     }


/* Initialize extremes of x movement during the approximation phase: */

minimum_x = x[trial-1];
maximum_x = x[trial-1];


/* Perform trials for the approximating sequential phase: */

while (trial < NT)
     {
      /* Determine the Logit-MLE sample point: */

      Logit_MLE(x,ye,trial);

      /* Determine the approximated sample point: */

      Approximate(w,ya,trial,ATYPE);

      /* Update the x movement variables: */

      minimum_x = (x[trial] < minimum_x) ? x[trial] : minimum_x;
      maximum_x = (x[trial] > maximum_x) ? x[trial] : maximum_x;

      /* Calculate the error on final trial of approximation phase: */

      if (trial == NT-1)
        {
         x_movement = maximum_x - minimum_x;
         error      = fabs(w[trial]-x[trial]) / x_movement;
        }

      /* Generate the binary responses for the current trial: */

      u         = Uniform(seeda,seedb,reset);
      p         = Quantal(x[trial]);
      ye[trial] = (u < p)  ?  1 : 0;
      p         = Quantal(w[trial]);
      ya[trial] = (u < p)  ?  1 : 0;

      trial++;
     }
```

```
        /* Increment the count of completed simulations: */

        completed++;


        /* Print the summary line for the current run: */

        Run_Summary(run, error);


        /* Update the overall error measurements: */

        sum_error += error;
        max_error  = (error > max_error) ? error : max_error;
       }

    /* Print the simulation table footing: */

    Print_Footing(sum_error/completed, max_error);

}
```

VITA

Scott R. Kegler

Candidate for the Degree of

Doctor of Philosophy

Dissertation: FIRST-ORDER DYNAMICS OF THE LOGIT-MLE PROCEDURE

Major Field: Statistics

Biographical:

Education: Graduated from Frank B. Kellogg High School, Roseville, Minnesota in June 1974; received a Bachelor of Science in Economics and a Bachelor of Arts in Statistics from the University of Minnesota in 1979 and 1990, respectively; received a Master of Operations Research and a Master of Science in Applied Mathematics from North Carolina State University in 1992 and 1995, respectively. Completed the requirements for the Doctor of Philosophy degree with a major in Statistics at Oklahoma State University in May, 1999.

Experience: Employed as a research analyst at the Minnesota Department of Corrections from 1980 to 1981; employed as a programmer, programmer/analyst and systems analyst at the Star Tribune Newspaper in Minneapolis, Minnesota from 1981 to 1990; employed as a software engineer at Barr-Mullin, Inc. in Raleigh, North Carolina from 1994 to 1995.

Professional Memberships: Institute for Management Sciences and Operations Research, American Statistical Association.