A NEW METHOD FOR THE ROBBINS-MONRO STOCHASTIC APPROXIMATION PROCEDURE

By

WUCHEN FEI

Bachelor of Business Administration
National Defense Management College
Taipei, Taiwan, ROC
1978

Master of Business Adminstration
Tamkang University
Taipei, Taiwan, ROC
1982

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
July, 1989

A NEW METHOD FOR THE ROBBINS-MONRO

STOCHASTIC APPROXIMATION

PROCEDURE

Thesis Approved:

Barry Kurt Moser
Thesis Adviser

D. Kenny Folks

Ronald W. McNew

M. P. Terrell

Norman N. Durham
Dean of the Graduate College

ACKNOWLEDGMENTS

I wish to express my appreciation to my adviser, Dr. Barry K. Moser, for suggesting this topic and for his valuable advice and encouragement. I also wish to thanks my committee chairman, Dr. Leroy J. Folks, and my committee members, Dr. Ronald M. McNew, Dr. Gary R. Stevens and Dr. Kenneth E. Case, for their helpful suggestions and assistance in my qualifying examination. I also like to appreciate my colleague at National Defense Management College for their encouragement.

I want to acknowledge the financial support I have received in the past three years from my government, Republic of China.

Special gratitude is to my parents, Tzu-Guin Fei, Hsiang-Wei Liu, my wife, Yi-Fang Lo, and my son, Kwang-Hwa Fei for their gracious acceptance of my long absence.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

In this chapter, a literature review of the Robbins-Monro stochastic approximation procedure is presented. A new stochastic approximation procedure is also introduced.

Literature Review

In 1951, Robbins and Monro introduced a method for finding the root of an increasing regression function by successive approximations. They considered the model

$$Y(x_i) = M(x_i) + Z(x_i) \qquad i=1,2,... \tag{1.1}$$

where $Y(x_i)$ denotes the response at level x_i, M is a regression function, and $Z(x_i)$ represents the random error at level x_i with $EZ(x_i)=0$ and $EZ^2(x_i)=\sigma^2$.

In the deterministic case (where $Z(x_i) = 0$ for all i), the Newton-Raphson method for finding the root L_p of the equation $M(x) = p$ is a sequential scheme defined by the recursive formula

$$X_{n+1} = X_n - (Y_n - p)/M'(X_n) \tag{1.2}$$

where $M'(x)$ is the tangent slope of M at x.

In the stochastic model (where $Z(x_i)$ are random variables), the Robbins-Monro (RM) scheme is defined by the recursive formula

$$X_{n+1} = X_n - a_n(Y_n - p), \tag{1.3}$$

where a_n are positive constants such that $\Sigma a_n = \infty$ and $\Sigma a_n^2 < \infty$. Robbins and Monro showed that X_n converges to L_p in L^2. Blum (1954), Dvoretzky (1956), and Robbins and Siegmund (1971) proved that X_n converges to L_p almost surely (a.s.) under certain conditions.

Chung (1954) and Sacks (1958) defined $a_n = n^{-1}A$ where A is a positive constant. Under some assumptions on Z and M, they established that $n^{1/2}(X_n - L_p)$ has an asymptotic normal distribution with mean zero and variance $A^2\sigma^2/(2A\alpha-1)$, where $\sigma^2 = \lim_{n\to\infty} \mathrm{Var}Y(x_n)$ and $\alpha > 0$ is the tangent slope of M at $x = L_p$. A minimum asymptotic variance σ^2/α^2 is obtained when $A = \alpha^{-1}$.

In practice, without knowledge of M, α is unknown. Thus, for a certain parametric function M with unknown parameters, defining an efficient procedure such that X_n having the minimum asymptotic variance is natural. This problem was considered first by Sakrison (1965) and then by Albert and Gardner (1967). Both Sakrison and Albert and Gardner replaced the constant α by a stochastic sequence estimating α. In both cases, the estimating sequence depends on the function M. The case where M is unknown was considered by Venter (1967).

Venter's method requires two observations Y'_n and Y''_n at $x_n - c_n$ and $x_n + c_n$ where x_n is the nth approximation and $\{c_n\}$ is a sequence of positive constants which converges to zero. Although Venter's method is asymptotically efficient, Anbar (1978) noted that taking two observations at a time may not be feasible in situations where the total number of experiments allowed is small. Anbar suggested the following procedure:

$$X_{n+1} = X_n - A_{mn} n^{-1}(Y_n - p), \qquad n > m(n) \tag{1.4}$$

where

$$A_{mn}^{-1} = \begin{cases} \delta_1 & \text{if } b_{mn} \leq \delta_1 \\ b_{mn} & \text{if } \delta_1 < b_{mn} < \delta_2 \\ \delta_2 & \text{if } \delta_2 \leq b_{mn}, \end{cases} \tag{1.5}$$

$$0 < \delta_1 < \delta_2 < \infty,$$

and b_{mn} is the least squares estimator of $M'(L_p)$ at stage n and defined by:

$$b_{mn} = \sum_m^n (X_i - \bar{X}_{mn})(Y_i - p) \Big/ \sum_m^n (X_i - \bar{X}_{mn})^2 \tag{1.6}$$

$$\bar{X}_{mn} = \sum_{m+1}^n X_i/(n-m) \tag{1.7}$$

$$m = m(n) = o((\log n)^{1/2+\varepsilon}) \quad \text{for every } \varepsilon > 0$$

$$\lim_{x\to\infty} o(x)/x = 0$$

Under some assumptions on M and Z, Anbar proved that X_n in (1.4) converges to L_p a.s., b_{mn} converges to $M'(L_p)$

a.s., and $n^{1/2}(X_n - L_p)$ converges in law to a normal random variable with mean zero and variance σ^2/α^2. Since Anbar's procedure attains the optimal asymptotic variance σ^2/α^2, it is an efficient procedure. Lai and Robbins (1981) have proven similar results under the assumption that $Z(x_i)$ are i.i.d. random variables. They also demonstrated the convergence speed of x_n. In both Venter's and Anbar's procedure, X_n is a function of $x_1, y_1, \ldots, y_{n-1}$. Because these procedures estimate α at each stage, they are called adaptive RM procedures. Adaptive RM procedure are often applied in situations where Y(x) is a dichotomous random variable. However, dichotomous random variables are only one type of random variable that applied to the adaptive RM procedure.

In many fields of research, the outcomes of an experiment are assumed to be dichotomous (response or nonresponse). In testing the strength of materials, the stimulus level may be the level of impact energy applied to a piece of material, and the response is either "fail" or "not fail" (Wetherill [1963]). In testing explosives, the stimulus level may be the height from which a weight is dropped or the pressure directly applied to the explosive, and the response is "explode" or "not explode" (Dixon and Mood [1948]). In biology, a test animal either lives or dies at a given dose level (Finney [1978]). In an educational experiment, one may want to study the item characteristic curve that relates the difficulty level of the test item to the probability of a right or a wrong answer (Lord [1971]).

The main interest of this type of research is to estimate the percentiles of the response curve F(x), the distribution function of the binary random variable Y at a given stimulus level x. The 100pth percentile L_p is defined as:

$$F(L_p) = p. \tag{1.8}$$

That is, L_p is the root of the equation F(x) = p. The median $L_{0.5}$ of F is the most commonly used measure of the response curve. In some cases, however, it may be more relevant to study the extreme percentiles. For example, in finding the impact energy level for which the material fails 10% of the times. On the other hand, $L_{0.9}$ may be more relevant in explosive research.

Let y_n = 1 or 0 when the nth observation is a response or nonresponse. For estimating L_p by a RM procedure, the stimulus level X_{n+1} is chosen according to the formula:

$$X_{n+1} = X_n - An^{-1}(Y_n - p). \tag{1.9}$$

The small-sample behavior of the RM procedure depends heavily on a good initial guess x_1 (Wetherill [1963]). However, a good guess of L_p is also hard to achieve. Poor choices of A and x_1 will make (1.9) an inefficient procedure for small or moderate samples.

Wu (1985) proposed another sequential design procedure. He wanted to have a good estimate $\hat{F}_n$ of the whole curve F, from which the next point x_{n+1} is chosen to be the 100pth percentile of $\hat{F}_n$, that is $\hat{F}_n(x_{n+1}) = p$. He also noted that a smooth nonparametric estimate of F(x) was not feasible without a large number of observations. Therefore, he adopted the approach of assuming a parametric form for the distribution function of the random variable Y. Let $F(x) = P(Y=1|x)$ be the distribution function of binary random variable Y at the level x, and let

$$F(x) = H(x|\theta), \quad H \text{ is continuous in } x$$

$$\lim_{x \to \infty} H(x|\theta) = 1, \quad \lim_{x \to -\infty} H(x|\theta) = 0$$

where θ is a vector of unknown parameters.

Wu's sequential design procedure for estimating L_p is as follows:

1. Find an efficient estimate $\hat{\theta}_n = \hat{\theta}((x_1,y_1),\ldots(x_n,y_n))$ of θ (Wu uses the maximum likelihood estimator MLE).
2. Define the estimated quantal response curve by $\hat{F}_n(x) = H(x|\hat{\theta}_n)$ and choose the next design x_{n+1} such that $\hat{F}_n(x_{n+1}) = p$.

He noted that the change from x_n to x_{n+1} via the MLE version of his method may be unduly large when the problem is ill posed. This can happen in the first few runs after the existence, and uniqueness of the MLE is first satisfied. Thus, he proposed a truncated version of his procedure as follows. Define d_n as the solution of

$$x_{n+1} = x_n - n^{-1}d_n(y_n - p) \qquad (1.10)$$

where $x_{n+1} = \hat{F}_n^{-1}(p)$. For example, consider the distribution function of the logit model, $F(x) = [1 + \exp(-\mu-\beta x)]^{-1}$. Define $x_{n+1} = [\log \frac{p}{1-p} - \hat{\mu}_n]/\ \hat{\beta}_n$ where $\hat{\mu}_n$ and $\hat{\beta}_n$ are the MLE's of μ and β at stage n. Then, the (n+1)th design level is chosen to be

$$x_{n+1} = x_n - n^{-1}d_n^*(y_n - p) \qquad (1.11)$$

where

$$d_n^* = \max\ [\delta_1, \min(\delta_2, d_n)], \qquad 0 < \delta_1 < \delta_2 < \infty$$

Wu did show that his procedure was consistent for the one parameter logit model. Assuming consistency, he also proved that his procedure is asymptotically equivalent in first order to the efficient RM procedure for the two parameter logit model. However, Wu was unable to prove the asymptotic normality of x_n and d_n^*. Thus, he could not establish the asymptotic normality of $\hat{L}_{p*}$(the estimator of the root of $M(L_{p*}) = P^*$ for any $0 < p^* < 1$). The most negative aspect of Wu's procedure is that the Newton-Raphson method must be used repeatedly to estimate the MLE's of the parameters for each step in the stochastic procedure and the Newton-Raphson method is a time consuming procedure.

A New Adaptive RM Procedure

The purpose of this research is to define an efficient stochastic procedure for estimating the entire curve of an increasing function M(x), the expectation of random variable Y(x). All the procedures discussed previously are designed to estimate a single root. The objective of this new procedure is to estimate all the roots of M(x), that is, to estimate the entire curve M(x). The idea of this new method is very simple. In Chapters 1 to 4, it is assumed that M(x) is a two parameter increasing function. The general form for M with r parameters will be developed in Chapter 5. Let

$$Y(X_i) = M(X_i) + Z(X_i) \tag{1.12}$$

where

$$M(X_i) = E\ Y(X_i). \tag{1.13}$$

Let λ be the slope of the line through (L_p, p) and $(L_{p'}, p')$, and $M'(x) = \frac{\partial}{\partial x} M(x)$ be the tangent slope of M at x. Let $\alpha = M'(L_p)$ and $\alpha' = M'(L_{p'})$. By Figure 1.1, it is found that $\lambda = (p'-p)/(L_{p'} - L_p)$, $\alpha = c\lambda$, and $\alpha' = c'\lambda$, where c and c′ are positive constants which depend on the assumed parametric form of M(x). The relationship between c, c′ and M(x) for different models will be discussed in Chapter 2.

The new adaptive RM procedure for estimating $(L_p\ ,\ L_{p'})$ is given by

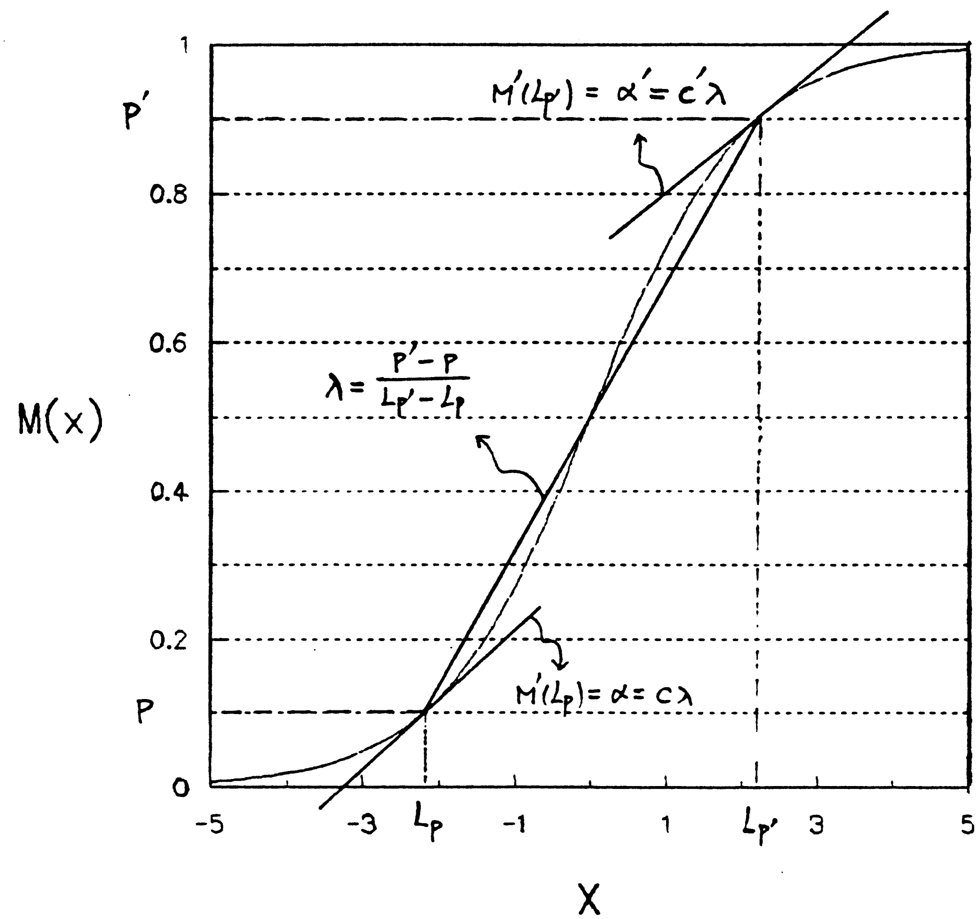

Figure 1.1 Relationship Between α and c_i for Two Parameters Case

$$\begin{pmatrix} X_{n+1} \\ X'_{n+1} \end{pmatrix} = \begin{pmatrix} X_n \\ X'_n \end{pmatrix} - \begin{pmatrix} a_n(Y_n - p) \\ a'_n(Y'_n - p') \end{pmatrix} \tag{1.14}$$

where

$$a_n = [n\hat{\alpha}_n]^{-1} = [nc\hat{\lambda}_n]^{-1} \tag{1.15}$$

$$a'_n = [n\hat{\alpha}'_n]^{-1} = [nc'\hat{\lambda}_n]^{-1}. \tag{1.16}$$

The bounded versions of a_n and a'_n of this new procedure are defined by

$$a_n = n^{-1}A_n \tag{1.17}$$

and

$$a'_n = n^{-1}A'_n \tag{1.18}$$

where

$$A_n^{-1} = \begin{cases} \delta_1 & \text{if } \hat{\alpha}_n \le \delta_1 \\ \hat{\alpha}_n & \text{if } \delta_1 < \hat{\alpha}_n < \delta_2 \\ \delta_2 & \text{if } \delta_2 \le \hat{\alpha}_n \end{cases} \tag{1.19}$$

$$A'^{-1}_n = \begin{cases} \delta_1 & \text{if } \hat{\alpha}'_n \le \delta_1 \\ \hat{\alpha}'_n & \text{if } \delta_1 < \hat{\alpha}'_n < \delta_2 \\ \delta_2 & \text{if } \delta_2 \le \hat{\alpha}'_n \end{cases} \tag{1.20}$$

$$0 < \delta_1 < \delta_2 < \infty$$

Since (x_n, x'_n) is used to estimate $(L_p, L_{p'})$, a natural estimator of λ^{-1} is

$$\hat{\lambda}_n^{-1} = (x'_n - x_n)/(p' - p). \tag{1.21}$$

Other estimators of λ such as the LSE $\hat{\lambda}_n = (cA_{mn})^{-1}$

(where A_{mn} and c are defined in (1.5) and (1.15) respectively) and the MLE $\hat{\lambda}_n = (cd_n^*)^{-1}$ (where d_n^* is defined in (1.11)) exist. However, calculations for $\hat{\lambda}_n^{-1}$ in (1.21) is easier and faster than for the LSE and MLE. It will be shown that $\hat{\lambda}_n^{-1}$ in (1.21) has desirable asymptotic and small sample properties.

Silvapulle (1981) mentioned that the MLE of $M'(L_p)$ for binary data exists only when certain conditions are satisfied. Frequently, these conditions are not satisfied for small samples (Wu [1985]). However, the estimator $\hat{\lambda}_n^{-1}$ in (1.21) always exists provided initial estimates (x_1, x_1') are available. Moreover, the convergence, asymptotic normality for the estimator of any root L_{p*} is easily obtained by the linear combination of $\hat{L}_p$ and $\hat{L}_{p'}$.

The convergence and asymptotic normality theorems for the estimators $\hat{L}_p$, $\hat{L}_{p'}$, and their linear combinations generated by the new procedure will be derived in Chapter 2. In Chapter 3, some examples of binary data models with two parameters are presented. The robustness of the root estimators from the new procedure is also discussed. In Chapter 4, the root estimators from Robbins-Monro procedure, Anbar's procedure, Wu's procedure and this new procedure are compared in a Monte Carlo simulation study. The general form and conclusions of the new procedure are drawn in Chapter 5.

CHAPTER II

CONVERGENCE AND ASYMPTOTIC NORMALITY

In this chapter, the convergence and asymptotic normality of the estimators of L_p and $L_{p'}$ from the new procedure will be discussed.

Assumptions

For the purpose of easy reference, all assumptions which will be needed in this procedure are listed below.

(M1) M(x) is a Borel-measurable function satisfying

$(M(x)-P)(x-L_p) > 0$ for all $x \neq L_p$

(M2) $\inf_{\varepsilon<|x-L_p|<1/\varepsilon} |M(x)-P| > 0$ for every $0 < \varepsilon < 1$

(M3) $M(x) = p + \alpha(x-L_p) + o(x-L_p)$

where $\lim_{x\to 0} o(x)/x = 0$ and $0 < \alpha < \infty$

(M4) There exists finite positive number K such that

$|M(x)-p| \le K|x-L_p|$ for all $x \neq L_p$

(Z1) (i) $\sup_x EZ^2(x) < \infty$

(ii) $\inf_x EZ^2(x) > 0$

(Z2) $\lim_{x\to L_p} EZ^2(x) = \sigma^2(p) < \infty$

(Z3) $$\lim_{R\to\infty} \lim_{\varepsilon\to 0+} \sup_{|x-L_p|<\varepsilon} \int_{\{|Z(x)|>R\}} Z^2(x)\, dM = 0$$

Convergence

In this section, the convergence theorems of (x_n, x'), A_n, A'_n and the linear combinations of (x_n, x'_n) are derived, where these terms are defined in (1.14) to (1.20).

Let $\{Y(x), -\infty < x < \infty\}$ be a family of random variables with $EY(x)=M(x)$ and $VarY(x)=\sigma^2(x) < \infty$. The new procedure is designed to find the roots $x=L_p$ and $x=L_{p'}$ of the equations $M(L_p)=p$ and $M(L_{p'})=p'$. Starting with an arbitrary random variable (X_1,X'_1) and defining successively (X_2, X'_2), (X_3, X'_3),... by (1.14) to (1.20), a_n and a'_n are non-negative functions of (x_1, x'_1), (y_1, y'_1), ..., (y_{n-1}, y'_{n-1}). Conditional on (x_1, x'_1), (y_1,y'_1), ..., (y_{n-1},y'_{n-1}), the random variables Y_n and Y'_n have distributions of $Y(x_n)$ and $Y(x'_n)$ which depend only on the values of x_n and x'_n, respectively. This implies that random variables $Z(x_n)$ and $Z(x'_n)$ defined by (1.12) and conditional on (x_1, x'_1), (y_1, y'_1), ..., (y_{n-1}, y'_{n-1}) are independent.

The following lemma, which is adopted from Robbins and Siegmund (1981) Application 2 p.242, will be used in Theorem 2.1 to prove the almost surely convergence of $(X_n, X'_n)'$ in (1.14) to $(L_p, L_{p'})'$.

<u>Lemma 2.1</u>: If σ and M are measurable and for some $0 < a,b < \infty$

$$\sigma(x) + |M(x)| \leq a + b(x), \qquad (2.1)$$

there exists a real number θ such that

$$\inf_{\varepsilon < |x-\theta| < 1/\varepsilon} |M(x)-p| > 0 \quad \text{for all } 0 < \varepsilon < 1 \qquad (2.2)$$

Define the recursive formula by

$$X_{n+1} = X_n - a_n(Y_n - p).$$

If $\Sigma_1^\infty a_n = \infty$ and $\Sigma_1^\infty a_n^2 < \infty$ for every sequence x_1, y_1, y_2, ... such that

$$\sup |x_n| < \infty \qquad (2.3)$$

then $\lim_{n\to\infty} x_n = \theta$ with probability one.

<u>Theorem 2.1:</u> If (M1), (M2), (M4), and (Z1)(i) are satisfied, then $\begin{pmatrix} X_n \\ X'_n \end{pmatrix}$ defined in (1.14) converges to $\begin{pmatrix} L_p \\ L_{p'} \end{pmatrix}$ almost surely (a.s.).

<u>Proof:</u> The recursive formula

$$\begin{pmatrix} X_{n+1} \\ X'_{n+1} \end{pmatrix} = \begin{pmatrix} X_n \\ X'_n \end{pmatrix} - \begin{pmatrix} a_n(Y_n - p) \\ a'_n(Y'_n - p') \end{pmatrix} \qquad (2.4)$$

implies

$$X_{n+1} = X_n - a_n(Y_n - p)$$

and

$$X'_{n+1} = X'_n - a'_n(Y'_n - p').$$

By assumptions (M1), (M4) and (Z1)(i), equation (2.1) is satisfied. By assumption (M2), equation (2.2) is satisfied. Moreover, by (1.17) and (1.19),

$$\infty = \Sigma\, \alpha_1 n^{-1} \le \Sigma\, a_n = \Sigma\, A_n n^{-1} \le \Sigma\, \alpha_2 n^{-1} = \infty$$

and

$$0 \le \Sigma\, a_n^2 = \Sigma\, A_n^2 n^{-2} \le \Sigma\, \alpha_2^2 n^{-2} < \infty.$$

By (1.18) and (1.20) we have

$$\infty = \Sigma\, \alpha_1 n^{-1} \le \Sigma\, a'_n = \Sigma\, A'_n n^{-1} \le \Sigma\, \alpha_2 n^{-1} = \infty$$

and

$$0 \le \Sigma\, a'^2_n = \Sigma\, A'^2_n n^{-2} \le \Sigma\, \alpha_2^2 n^{-2} < \infty.$$

Since all a_n, a'_n, $y_n - p$, and $y'_n - p'$ are finite, equation (2.3) is satisfied. By Lemma 2.1, it follows that X_n converges to L_p and X'_n converges to $L_{p'}$ a.s.

Q.E.D.

The following lemma, which is adoptive from Serfling (1980) p.25, will be used in Theorem 2.2 to prove the convergence of (A_n , A'_n) in (1.19) and (1.20) to $(\alpha^{-1}, \alpha'^{-1})$.

<u>Lemma 2.2</u>: Suppose that the k-vector $\mathbf{X}_n$ converges to the k-vector $\mathbf{X}$ almost surely, in probability, or in distribution. Let $B_{m\times k}$ be a constant matrix. Then $B\mathbf{X}_n$ converges to $B\mathbf{X}$ in the given mode of convergence.

<u>Theorem 2.2</u>: If (M1), (M2), (M4) and (Z1)(i) are satisfied, then A_n converges to α^{-1} a.s. and A'_n converges to α'^{-1} a.s., where A_n and A'_n are defined by (1.19) and (1.20).

<u>Proof</u>: Let λ be the slope of the line through (L_p , p) and ($L_{p'}$, p'). Thus

$$\lambda = (p' - p)/(L_{p'} - L_p) \tag{2.5}$$

and

$$\alpha = \frac{\partial}{\partial x} M(L_p) = c\lambda \tag{2.6}$$

$$\alpha' = \frac{\partial}{\partial x} M(L_{p'}) = c'\lambda \tag{2.7}$$

where c and c′ are positive constants depending on the distribution used (see Figure 1.1).

By Theorem 2.1, X_n and X'_n converge to L_p and $L_{p'}$ a.s., respectively. From Lemma 2.2, let $\mathbf{X}_n = (X_n\ ,\ X'_n)'$ and $B_{1x2} = \frac{1}{c(p'-p)}(-1\ ,\ 1)$. Thus, $A_n = \frac{(Xn'-Xn)}{c(p'-p)}$ converges to $\alpha^{-1} = \frac{Lp'-Lp}{c(p'-p)}$ almost surely. Similarly, $A'_n = \frac{(Xn'-Xn)}{c'(p'-p)}$ converges to $\alpha'^{-1} = \frac{Lp'-Lp}{c'(p'-p)}$ almost surely.

Q.E.D.

<u>Theorem 2.3</u>: If (M1), (M2), (M4) and (Z1)(i) are satisfied. For any p^*, the estimator of the root $x = L_{p*}$ of $M(x) = p^*$ can be presented as

$$\hat{L}_{p*} = kX_n + (1-k)X'_n \tag{2.8}$$

where $0 < k < \infty$. Then $\hat{L}_{p*}$ converges to

$$L_{p*} = kL_p + (1-k)L_{p'} \tag{2.9}$$

<u>Proof</u>: Since $\begin{pmatrix} X_n \\ X'_n \end{pmatrix}$ converges to $\begin{pmatrix} L_p \\ L_{p'} \end{pmatrix}$ a.s., by Lemma 2.2 , $\hat{L}_{p*}$ converges to L_{p*} a.s. Q.E.D.

<u>Remark</u>: Constant k defined in (2.8) and (2.9) depends on the function M(x). The relationship between k and M

will be discussed in the last section of this chapter.

Asymptotic Normality

In this section, the asymptotic normality of (X_n, X'_n), $\hat{L}_{p*}$, $\hat{\alpha}_n^{-1}$ and $\hat{\alpha}'^{-1}_n$ are derived.

The following lemma, which is adoptive from Sacks (1958) p.383, will be used in Theorem 2.4 to prove the asymptotic normality of $(X_n, X'_n)'$.

Lemma 2.3: Suppose (M1), (M3), (M4), (Z1), (Z2) and (Z3) are satisfied. Let $a_n = An^{-1}$ where A is a positive constant such that $2A\alpha > 1$. Then $n^{1/2}(X_n - L_p)$ is asymptotically normally distributed with mean zero and variance $A^2\sigma^2(2A\alpha - 1)^{-1}$.

Theorem 2.4: Suppose (M1) to (M4) and (Z1) to (Z3) are satisfied. Then

$$\sqrt{n}\begin{pmatrix} X_n - L_p \\ X'_n - L_{p'} \end{pmatrix} \sim AN_2\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \sigma^2/\alpha^2 & 0 \\ 0 & \sigma'^2/\alpha'^2 \end{pmatrix}\right)$$

Proof: By Theorem 2.2, A_n and A'_n converge to α^{-1} and α'^{-1} almost surely. Since α^{-1} and α'^{-1} are positive constants and both $2A\alpha$ and $2A'\alpha'$ greater than one, Lemma 2.3 implies that $\sqrt{n}(X_n - L_p)$ and $\sqrt{n}(X'_n - L_{p'})$ converge to Z_1 and Z_2 in distribution, where Z_1 and Z_2 are normal random variables with mean zero and variances σ^2/α^2 and σ'^2/α'^2, respectively. In equation (2.2), X'_n and X_n are correlated through A_n and A'_n. Note that A_n and A'_n converges to α^{-1} and α'^{-1}, and Y_n, Y'_n

are independent binary random variables. Thus, X'_n and X_n are asymptotically uncorrelated, and therefore, asymptotically independent. Q.E.D.

<u>Theorem 2.5</u>: If all assumptions in theorem 2.4 are satisfied, then $\sqrt{n}(\hat{L}_{p*} - L_{p*})$ is asymptotically normal with mean zero and variance $\sigma^2 k^2/\alpha^2 + \sigma'^2(1-k)^2/\alpha^2$.

<u>Proof</u>: Let B from Lemma 2.2 equal (k, 1-k). Since

$$\sqrt{n}\begin{pmatrix} X_n - L_p \\ X'_n - L_{p'} \end{pmatrix} \sim AN_2\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \sigma^2/\alpha^2 & 0 \\ 0 & \sigma'^2/\alpha'^2 \end{pmatrix}\right)$$

Lemma 2.2 implies

$$\sqrt{n}(\hat{L}_{p*} - L_{p*}) \sim AN(0,\ \sigma^2 k^2/\alpha^2 + \sigma'^2(1-k)^2/\alpha'^2)$$

Q.E.D.

<u>Theorem 2.6</u>: If all assumptions in theorem 2.4 are satisfied. Then $\sqrt{n}(\hat{\alpha}_n^{-1} - \alpha^{-1})$ is asymptotically normal with mean zero and variance $\sigma^2/((p'-p)\alpha)^2$. Similarly, $\sqrt{n}(\hat{\alpha}_n^{-1} - \alpha_n^{-1})$ is asymptotically normal with mean zero and variance $\sigma'^2/((p'-p)\alpha')^2$.

<u>Proof</u>: Note that

$$\sqrt{n}\begin{pmatrix} X_n - L_p \\ X'_n - L_{p'} \end{pmatrix} \sim AN_2\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \sigma^2/\alpha^2 & 0 \\ 0 & \sigma'^2/\alpha'^2 \end{pmatrix}\right)$$

and

$$\sqrt{n}(\hat{\alpha}_n^{-1} - \alpha^{-1}) = \sqrt{n}((X'_n - X_n) - (L_{p'} - L_p))(c(p'-p))^{-1}$$

$$= \left(\frac{\sqrt{n}}{c(p'-p)},\ \frac{\sqrt{n}}{c(p'-p)}\right)\begin{pmatrix} X_n - p \\ X'_n - p' \end{pmatrix}.$$

Thus, by Lemma 2.2, $\sqrt{n}(\hat{\alpha}_n^{-1} - \alpha^{-1})$ is asymptotically normally distributed with mean 0 and variance $(\sigma^2\alpha^{-2} + \sigma'^2\alpha'^{-2})/(c(p'-p))^2$. Also, $\sqrt{n}(\hat{\alpha}_n'^{-1} - \alpha'^{-1})$ is asymptotically normally distributed with mean zero and variance $(\sigma^2\alpha^{-2} + \sigma'^2\alpha'^{-2})/(c'(p'-p))^2$. Q.E.D.

Remark: Note that the new procedure defined in (1.14) to (1.20) assumes the values of c and c′ are known. The values of c and c′ are derived from the assumed parametric model form of M(x). If the assumed model is different from the true model, by Lemma 2.1, (X_n , X_n') will still converge to $(L_p , L_{p'})$. However, the minimal asymptotic variance defined in Theorem 2.4 will not be attained. Similar conclusion can also apply to Theorem 2.5 and Theorem 2.6. Details will be discussed in the Chapter 3.

Remark: Assumption (M1) implies that M(x) is an increasing function of x. Thus, M′(x) is greater than zero. It is natural, in practice, to restrict X_n' and X_n such that $X_n'-X_n > 0$ for all n. The truncated version of the random variables $\hat{\alpha}_n^{-1}$ and $\hat{\alpha}_n'^{-1}$ are used throughout the reminder of the paper.

Since $X_n'-X_n$ is asymptotically normally distributed with mean $L_{p'}-L_p$ and variance $\frac{1}{n}(\sigma^2/\alpha^2 + \sigma'^2/\alpha'^2)$, the distribution of $X_n'-X_n$ will concentrate around $L_{p'}-L_p$ as n increases. Thus, the probability that $X_n'-X_n \le 0$ converges to 0. Now, $\hat{\alpha}_n^{-1} = (X_n'-X_n)/c(p'-p)$, and $\hat{\alpha}_n'^{-1} =$

$(X'_n - X_n)/c'(p'-p)$ are functions of $X'_n - X_n$. Hence, the truncated version of the random variables $\hat{\alpha}_n^{-1}$ and $\hat{\alpha}'^{-1}_n$ will have the same asymptotic normal distributions as $\hat{\alpha}_n^{-1}$ and $\hat{\alpha}'^{-1}_n$. This conclusion also applies to random variables $\hat{\lambda}_n^{-1}$ and $\hat{\lambda}'^{-1}_n$.

Binary Data Distributions

Binary random variables, $Y(x)$, provide a major area of application for the new adaptive RM procedure defined in (1.14) to (1.20). In this section, four different parametric forms of $M(x)$ are discussed for binary data. They are the two parameter logit, skewed logit, log-log, and porbit models. The convergence and asymptotic normality for the estimators of the roots of these models are also discussed.

<u>Logit Model</u>: Let $M(x) = [1+\exp(-\mu-\beta x)]^{-1}$ where $-\infty < x, \mu < \infty$, and $0 < \beta < \infty$. Since M is an increasing function, there exists an unique percentile L_p for any $0 < p < 1$. Let L_p and $L_{p'}$ be the roots of $M(x)=p$ and $M(x)=p'$, respectively. Then,

$$L_p = \left(\ln\frac{p}{1-p} - \mu\right)/\beta \tag{2.10}$$

and

$$L_{p'} = \left(\ln\frac{p'}{1-p'} - \mu\right)/\beta. \tag{2.11}$$

The tangent slopes of M at $x=L_p$ and $x=L_{p'}$ are

$$\alpha = \beta p(1-p) \tag{2.12}$$

and

$$\alpha' = \beta p'(1-p'). \tag{2.13}$$

The slope of the line through (p, L_p) and $(p', L_{p'})$ is

$$\lambda = (p'-p)(L_{p'}-L_p)^{-1}. \tag{2.14}$$

In Chapter 1, it was mentioned that c and c′ are constants such that $\alpha = c\lambda$ and $\alpha' = c'\lambda$. For the logit model, substitute (2.10) through (2.14) into α and α'. Thus

$$c = p(1-p)\ \ln\left(\frac{p'(1-p)}{p(1-p')}\right)\ (p'-p)^{-1} \tag{2.15}$$

and

$$c' = p'(1-p')\ \ln\left(\frac{P'(1-P)}{P(1-P')}\right)\ (p'-p)^{-1}. \tag{2.16}$$

In the new procedure, (X_n, X'_n) are used to estimate $(L_p, L_{p'})$. Thus, substituting (X_n, X'_n) for $(L_p, L_{p'})$ in (2.10) and (2.11) yields the following estimators of the parameters μ and β:

$$\hat{\beta}_n = \ln\left(\frac{P'(1-P)}{P(1-P')}\right)\ (X'_n - X_n)^{-1} \tag{2.17}$$

$$\hat{\mu}_n = \left(\ln\left(\frac{P'P}{(1-P')(1-P)}\right) - \hat{\beta}_n (X_n + X'_n)\right)/2 \tag{2.18}$$

For any $0 < p^* < 1$, the estimator of the root L_{p*} can be presented as

$$\hat{L}_{p*} = \left(\ln\left(\frac{p*}{1-p*}\right) - \hat{\mu}_n\right)/\ \hat{\beta}_n \tag{2.19}$$

Now, replace $\hat{\mu}_n$, $\hat{\beta}_n$ by (2.18) and (2.17) to obtain

$$\hat{L}_{p*} = kX_n + (1-k)X'_n \tag{2.20}$$

where

$$k = \frac{\ln\left(\frac{p'(1-p*)}{(1-p')p*}\right)}{\ln\left(\frac{(1-p)p'}{p(1-p')}\right)} \tag{2.21}$$

By theorem 2.5, $\sqrt{n}(\hat{L}_{p*} - L_{p*})$ has a asymptotic normal distribution with mean zero and variance $\sigma^2(k/\alpha)^2 + \sigma'^2((1-k)/\alpha')^2$ where α, α', k are defined by (2.12), (2.13) and (2.21).

<u>Skewed Logit Model</u>: Let $M(x) = [1+\exp(-\mu-\beta x)]^{-2}$ where $-\infty < x, \mu < \infty$ and $0 < \beta < \infty$. Since M is an increasing function of x, there exists a unique root L_p of $M(x)=p$ for any $0 < p < 1$. Let L_p and $L_{p'}$ are the roots of $M(x)=p$ and $M(x)=p'$. Then

$$L_p = \left(\ln\frac{\sqrt{P}}{1-\sqrt{P}} - \mu\right)/ \beta \qquad (2.22)$$

and

$$L_{p'} = \left(\ln\frac{\sqrt{P'}}{1-\sqrt{P'}} - \mu\right)/ \beta. \qquad (2.23)$$

The tangent slopes of M at $x=L_p$ and $x=L_{p'}$ are

$$\alpha = 2\beta p(1-\sqrt{P}) \qquad (2.24)$$

and

$$\alpha' = 2\beta p'(1-\sqrt{P'}). \qquad (3.25)$$

The slope of the line through (p,L_p) and $(p',L_{p'})$ is defined by (2.14). The constants c and c' satisfying $\alpha = c\lambda$ and $\alpha' = c'\lambda$ can be obtained by substituting (2.22) through (2.25) into α and α'. Thus

$$c = \frac{2P(1-\sqrt{P})\ \ln\left(\sqrt{P'}(1-\sqrt{P})/\sqrt{P}(1-\sqrt{P'}\right)}{P' - P} \qquad (2.26)$$

and

$$c' = \frac{2P'(1-\sqrt{P'})\ \ln\left(\sqrt{P'}(1-\sqrt{P})/\sqrt{P}(1-\sqrt{P'})\right)}{P'-P}. \qquad (2.27)$$

Again, substitute (X_n, X'_n) for $(L_p, L_{p'})$ in (2.22) and (2.23) to obtain the following estimators of μ and β:

$$\hat{\beta} = \frac{\ln\left(\sqrt{P'}(1-\sqrt{P})/\sqrt{P}(1-\sqrt{P'})\right)}{X'_n - X_n} \qquad (2.28)$$

$$\hat{\mu} = \frac{1}{2}\left(\ln\left(\sqrt{P'P}/(1-\sqrt{P'})(1-\sqrt{P})\right) - \hat{\beta}_n(X'_n + X_n)\right). \qquad (2.29)$$

For any $0 < p^* < 1$, the estimator of the root $M(x)=p^*$ can be presented as

$$\hat{L}_{p*} = \left(\ln\left(\sqrt{p*}\ /(1-\sqrt{p*})\right) - \hat{\mu}_n\right)/\hat{\beta}_n. \qquad (2.30)$$

Now, replace $\hat{\mu}_n$, $\hat{\beta}_n$ by (2.29), (2.28) to obtain

$$\hat{L}_{p*} = kX_n + (1-k)X'_n \qquad (2.31)$$

where

$$k = \frac{\ln\left(\sqrt{P'}(1-\sqrt{p*})/(1-\sqrt{P'})\sqrt{p*}\ \right)}{\ln\left(\sqrt{P'}(1-\sqrt{P})/(1-\sqrt{P'})\sqrt{P}\ \right)}. \qquad (2.32)$$

By Theorem 2.5, $\sqrt{n}(\hat{L}_{p*}-L_{p*})$ is asymptotically normally distributed with mean zero and variance $\sigma^2k^2/\alpha^2 + \sigma'^2(1-k)^2/\alpha'^2$ where α, α', and k are defined by (2.24) , (2.25) and (2.32).

<u>Log-log Model</u>: Let $M(x) = 1-\exp[-\exp(\mu+\beta x)]$ where $-\infty < x, \mu < \infty$ and $0 < \beta < \infty$. Since M is an increasing function of x, there exist a unique root L_p of $M(x)=p$ for any $0 < p < 1$. Let L_p and $L_{p'}$ are the roots of $M(x)=p$ and $M(x)=p'$. Then

$$L_p = \left(\ln\left(\ln\frac{1}{1-P}\right) - \mu\right)/\ \beta \qquad (2.33)$$

and

$$L_{p'} = \left(\ln\left(\ln\frac{1}{1-p'}\right) - \mu\right) / \beta. \tag{2.34}$$

The tangent slopes of M at $x=L_p$ and $x=L_{p'}$ are

$$\alpha = \beta(1-p)\ \ln\frac{1}{1-p} \tag{2.35}$$

and

$$\alpha' = \beta(1-p')\ \ln\frac{1}{1-p'}. \tag{2.36}$$

The slope of the line through (p, L_p) and $(p', L_{p'})$ is defined by (2.14). The constants c and c′ satisfying $\alpha = c\lambda$ and $\alpha' = c'\lambda$ can be obtained by substituting (2.33) through (2.36) into α and α'. Thus

$$c = \frac{1-p}{p'-p}\ \ln\left(\frac{1}{1-p}\right)\ \ln\left(\frac{\ln(1-p')}{\ln(1-p)}\right) \tag{2.37}$$

and

$$c' = \frac{1-p'}{p'-p}\ \ln\left(\frac{1}{1-p}\right)\ \ln\left(\frac{\ln(1-p')}{\ln(1-p)}\right). \tag{2.38}$$

Now, substitute (X_n, X_n') for $(L_p, L_{p'})$ in (2.33) and (2.34) to obtain the following estimators of μ and β:

$$\hat{\beta}_n = \frac{\ln\left(\frac{\ln(1-p')}{\ln(1-p)}\right)}{X_n' - X_n} \tag{2.39}$$

$$\hat{\mu}_n = \frac{1}{2}\left(\ln\left(\ln(1-p)\ \ln(1-p')\right) - (x_n' + x_n)\right). \tag{2.40}$$

For any $0 < p^* < 1$, the estimator of the root $M(x)=p^*$ can be presented as

$$\hat{L}_{p*} = \left(\ln\left(\ln\frac{1}{1-p*}\right) - \hat{\mu}_n\right)/\hat{\beta}_n. \tag{2.41}$$

Now, replace $\hat{\mu}_n$, $\hat{\beta}_n$ by (2.40), (2.39) to obtain

$$\hat{L}_{p*} = kX_n + (1-k)X_n' \tag{2.42}$$

where

$$k = \frac{\ln\left(\frac{\ln(1-P')}{\ln(1-p*)}\right)}{\ln\left(\frac{\ln(1-P')}{\ln(1-p)}\right)}. \qquad (2.43)$$

By Theorem 2.5, $\sqrt{n}(\hat{L}_{p*}-L_{p*})$ is asymptotically normally distributed with mean zero and variance $\sigma^2k^2/\alpha^2 + \sigma'^2(1-k)^2/\alpha'^2$ where α, α', and k are defined by (2.35), (2.36) and (2.43).

<u>Probit Model</u>: Let $M(x) = F_z\left(\frac{x-\mu}{\beta}\right)$ where $0 < \beta < \infty$, $-\infty < x,\mu < \infty$ and $F_z(t) = \int_{-\infty}^{t} \frac{1}{\sqrt{2\pi}}\exp(-t^2/2)\,dt$. Since F_z be an strictly increasing function of t, F_z^{-1} exists for all $x \in R$. Thus,

$$L_p = \mu + \beta F_z^{-1}(P) \qquad (2.44)$$

and

$$L_{p'} = \mu + \beta F_z^{-1}(P'). \qquad (2.45)$$

The tangent slopes of M at $x=L_p$ and $x=L_{p'}$ are

$$\alpha = M'(L_p) = \frac{1}{\sqrt{2\pi}\,\beta}\exp\left(\frac{-1}{2}\left(\frac{Lp-\mu}{\beta}\right)^2\right) \qquad (2.46)$$

and

$$\alpha' = M'(L_{p'}) = \frac{1}{\sqrt{2\pi}\,\beta}\exp\left(\frac{-1}{2}\left(\frac{Lp'-\mu}{\beta}\right)^2\right). \qquad (2.47)$$

The slope of the line through (p, L_p) and $(p', L_{p'})$ is defined by (2.14). The constants c and c' satisfying $\alpha = c\lambda$ and $\alpha' = c'\lambda$ can be obtained by substituting (2.44) through (2.47) into α and α'. Thus,

$$c = \frac{Fz^{-1}(P') - Fz^{-1}(P)}{\sqrt{2\pi}\,(P'-P)}\exp\left(\frac{-1}{2}\left(F_z^{-1}(P)\right)^2\right) \qquad (2.48)$$

and

$$c' = \frac{F_Z^{-1}(P') - F_Z^{-1}(P)}{\sqrt{2\pi}\ (P'-P)} \exp\left(\frac{-1}{2}\left(F_z^{-1}(P')\right)^2\right). \quad (2.49)$$

Now, substitute (X_n, X_n') for $(L_P, L_{P'})$ in (2.44) and (2.45) to obtian the following estimators of μ and β:

$$\hat{\beta}_n = \frac{X_n' - X_n}{F_z^{-1}(P')-F_z^{-1}(P)} \quad (2.50)$$

$$\hat{\mu}_n = \frac{1}{2}\left((x_n'+x_n) - \hat{\beta}_n(F_z^{-1}(P')+F_z^{-1}(P))\right) \quad (2.51)$$

For any $0 < p^* < 1$, the estimator of the root $M(x)=p^*$ can be presented as

$$\hat{L}_{p*} = \hat{\mu}_n + \hat{\beta}_n F_z^{-1}(P^*) \quad (2.52)$$

Now, replace $\hat{\mu}_n$, $\hat{\beta}_n$ by (2.51), (2.50) to obtain

$$\hat{L}_{p*} = kX_n + (1-k)X_n' \quad (2.53)$$

where

$$k = \frac{F_Z^{-1}(P') - F_Z^{-1}(P^*)}{F_Z^{-1}(P') - F_Z^{-1}(P)} \quad (2.54)$$

By Theorem 2.5, $\sqrt{n}(\hat{L}_{p*}-L_{p*})$ is asymptotically normally distributed with mean zero and variance $\sigma^2 k^2/\alpha^2 + \sigma'^2(1-k)^2/\alpha'^2$ where α, α', and k are defined by (2.46), (2.47) and (2.54).

CHAPTER III

ROBUSTNESS

The asymptotic normality theorems discussed in Chapter 2 are derived under the assumption that the assumed model is the true model. It is now of interest to examine the asymptotic distribution of the estimator of a root if the true model is not the same as the assumed model.

Mean Square Error

By the results of Robbins and Monro (1951), $(X_n, X'_n)'$ from the new procedure will converge to $(L_p, L_{p'})'$, no matter what the true model is. Since a_n and a'_n in (1.15) and (1.16) are functions of X_n and X'_n, it can be proved that na_n and na'_n will converge to A and A', the inverse tangent slopes of the assumed model at $x=L_p$ and at $x=L_{p'}$.

By Lemma 2.3, $\sqrt{n}(X_n-L_p)$ is asymptotically normally distributed with mean zero and variance $A^2\sigma^2(2A\alpha-1)^{-1}$. If the assumed and the true models are identical, A and A' are equal to α and α'. Thus, the minimum asymptotic variances σ^2/α^2 and σ'^2/α'^2 are attained. However, if the assumed model is not equivalent to the true model,

$\sqrt{n}(X_n - L_p)$ and $\sqrt{n}(X_n' - L_{p'})$ are still asymptotically normal and independent. The minimum asymptotic variances σ^2/α^2 and σ'^2/α'^2, however, will not be attained and are replaced by $A^2\sigma^2(2A\alpha-1)^{-1}$ and $A'^2\sigma'^2(2A'\alpha'-1)^{-1}$.

The objective of the new procedure is to estimate the whole curve M(x). The root $x = L_{p*}$ of $M(x) = p^*$ can be expressed as a linear combination of the roots L_p and $L_{p'}$ where $M(L_p) = p$ and $M(L_{p'}) = p'$. That is, $L_{p*} = kL_p + (1-k)L_{p'}$ where k is based on the true model. If the assumed and true models are the same, by Theorem 2.3, the estimator $\hat{L}_{p*}$ in (2.15) will converge to L_{p*}. Hoever, if the assumed and the true models are not the same, then the wrong value of k will be used to estimate L_{p*}. In this case, $\hat{L}_{p*}$ will be biased and thus not converges to L_{p*}.

It is of interest to examine how robust the estimators from the new procedure are when the true model is not the same as the assumed model. The mean square error (MSE) of the estimator $\hat{L}_{p*}$ will be used as a measure of the estimation robustness.

The following notation are introduced for finding the MSE of the estimator of the true root. For any finite p, let L_p^a be the root of the assumed model M_a such that $M_a(L_p^a) = p$; L_p^t be the root of the true model M_t such that $M_t(L_p^t) = p$; $\hat{L}_p^a$ be the estimator of L_p^a. For given finite positive constants p and p', let k_a be the constant that satisfies the equation $k_a L_p^a + (1-k_a)L_{p'}^a = L_{p*}^a$; k_t be the constant satisfying the equation $k_t L_p^t + (1-k_t)L_{p'}^t = L_{p*}^t$; A^{-1} and A'^{-1} be the tangent slopes of the assumed model M_a at $x = L_p^t$ and $x = L_{p'}^t$; α and α' be the tangent slopes of the true model M_t at $x=L_p^t$ and $x=L_{p'}^t$.

The objective of the new procedure is to use X_n and X_n' to estimate L_{p*}^t. However, the estimate of L_{p*}^t is based on the assumed model. That is, L_{p*}^t is estimated by

$$\hat{L}_{p*}^a = k_a X_n + (1-k_a)X_n' \quad . \tag{3.1}$$

If the assumed and the true models are different, the value of k_a in (3.1) will not be the same as k_t. That is, the curve being estimated is not the true curve but the assumed curve. Therefore,

$$\begin{aligned} MSE(\hat{L}_{p*}^a) &= E\left(\hat{L}_{p*}^a - L_{p*}^t\right)^2 \\ &= E(\hat{L}_{p*}^a - L_{p*}^a)^2 + (L_{p*}^a - L_{p*}^t)^2 \\ &\quad + 2(L_{p*}^a - L_{p*}^t)\ E(\hat{L}_{p*}^a - L_{p*}^a). \end{aligned}$$

Now, $E(\hat{L}_{p*}^a) = k_a E(X_n) + (1-k_a)E(X_n')$, and (X_n, X_n') converges to $(L_p^t, L_{p'}^t)$, where $(L_p^t, L_{p'}^t) = (L_p^a, L_{p'}^a)$. Thus, $\hat{L}_{p*}^a$ converges to L_{p*}^a, not to L_{p*}^t. Hence, the mean square error of $\hat{L}_{p*}^a$ converges to

$$\begin{aligned} MSE(\hat{L}_{p*}^a) &= Var(\hat{L}_{p*}^a) + (L_{p*}^a - L_{p*}^t)^2 \\ &= Var(k_a X_n + (1-k_a)X_n') \end{aligned}$$

$$+ (k_a-k_t)^2(L_{p'}-L_p)^2. \quad (3.2)$$

By Lemma 2.3, the asymptotic variances of $\sqrt{n}(X_n-L_p)$ and $\sqrt{n}(X_n'-L_p)$ are $\sigma^2A^2(2A\alpha-1)^{-1}$ and $\sigma'^2A'^2(2A'\sigma'-1)^{-1}$. By Theorem 2.4, X_n and X_n' are asymptotically independent. Thus, $MSE(\hat{L}^a_{p*})$ converges to

$$\frac{1}{n}\left(k_a^2\sigma^2A^2(2A\alpha-1)^{-1} + (1-k_a)^2\sigma'^2A'^2(2A'\alpha'-1)^{-1}\right)$$
$$+ (k_a-k_t)^2(L_{p'}-L_p)^2 \quad (3.3)$$

Remark: Any root L_{p*} of the true model can be presented as the linear combination of the true roots L_p and $L_{p'}$. That is, $L_{p*} = k_tL_p+(1-k_t)L_{p'}$, where k_t depends on the true model. However, the true model is usually unknown. Thus, k_t is unknown. In estimating the true roots L_{p*}, the value of k_a will be used to replace k_t and calculated according to the assumed model. For example, if the assumed model is logit model, k_a will be calculated according to (2.28); if the assumed model is log-log model, then k_a will be calculated according to (2.50).

Remark: Since the random variables $Y(x_i)$ are generated from the true model, by the results of Robbins and Monro's paper, (X_n, X_n') converges to $(L_p, L_{p'})$ no matter what the assumed model is. In Chapter 2, it was shown that k_a and k_t are functions of p, p′, and p^*; A and A′ are functions of p, p′, L_p , and $L_{p'}$; also, L_p and $L_{p'}$ are functions of p, p′ and the parameters of the true

TABLE 3.1

MSE TABLE OF 16 POSSIBLE COMBINATIONS

P^*	True Model			
Assumed Models	Logit	Skewed Logit	Log-Log	Probit
Logit	V11	V12	V13	V14
Skewed Logit	V21	V22	V23	V24
Log-Log	V31	V32	V33	V34
Probit	V41	V42	V43	V44

model. Thus, the asymptotic MSE of $\hat{L}^{a}_{p*}$ is a function of p, p′, p^{*}, and the parameters of the true model.

The MSE of the estimator $\hat{L}^{a}_{p*}$ will be derived for the four binary data distributions which were mentioned in Chapter 2. Table 3.1 provides the sixteen possible combinations of the assumed and the true models for the four given distributions. The value of V_{ij} represents the mean square error of $\hat{L}^{a}_{p*}$ when the distribution is assumed to follow the i^{th} assumed model but the true distribution follows the j^{th} true model where i=1,...,4 and j=1,...,4. It is also assumed that μ and β are the two parameters of the true models.

<u>Case V11</u> : If the true and assumed models are logit, then

$$V11 = MSE(\hat{L}^{a}_{p*})$$
$$= \frac{1}{n}\left(k_a^2\sigma^2A^2(2A\alpha-1)^{-1}+(1-k_a)^2\sigma'^2A'^2(2A'\alpha'-1)^{-1}\right)$$
$$+ (k_a-k_t)^2(L_{p'}-L_p)^2,$$

where

$$k_a = k_t = \frac{\ln\left(\frac{P'(1-p*)}{(1-P')p*}\right)}{\ln\left(\frac{P'(1-P)}{(1-P')P}\right)}$$

$$L_p = \left(\ln\frac{P}{1-P} - \mu\right)/\beta$$

$$L_{p'} = \left(\ln\frac{P'}{1-P'} - \mu\right)/\beta.$$

$$A^{-1} = c_a\lambda = \frac{P(1-P)}{P'-P}\ \ln\left(\frac{P'(1-P)}{(1-P')P}\right)\ \frac{P'-P}{L_{p'}-L_p}$$

$$= \beta p(1-p) = \alpha$$

$$A'^{-1} = c_a'\lambda = \frac{P'(1-P')}{P'-P} \ln\left(\frac{p'(1-p)}{(1-p')P}\right) \frac{p'-p}{L_{p'}-L_p}$$

$$= \beta p'(1-p') = \alpha'$$

Case V12 : If the true model is skewed logit and the assumed model is the logit model, then

$$V12 = MSE(\hat{L}^a_{p*})$$

$$= \frac{1}{n}\left(k_a^2\sigma^2A^2(2A\alpha-1)^{-1}+(1-k_a)^2\sigma'^2A'^2(2A'\alpha'-1)^{-1}\right)$$

$$+ (k_a-k_t)^2(L_{p'}-L_p)^2,$$

where

$$k_a = \frac{\ln\left(\frac{P'(1-p*)}{(1-P')p*}\right)}{\ln\left(\frac{P'(1-P)}{(1-P')P}\right)}$$

$$L_P = \left(\ln\left(\frac{\sqrt{P}}{1-\sqrt{P}}\right) - \mu\right)/\beta$$

$$L_{P'} = \left(\ln\left(\frac{\sqrt{P'}}{1-\sqrt{P'}}\right) - \mu\right)/\beta$$

$$A^{-1} = c_a\lambda = \frac{P(1-P)}{P'-P} \ln\left(\frac{P'(1-P)}{(1-P')P}\right) \frac{P'-P}{L_{P'}-L_P}$$

$$= \frac{\beta p(1-p) \ln\left(\frac{P'(1-P)}{(1-P')P}\right)}{\ln\frac{\sqrt{P'}(1-\sqrt{P})}{\sqrt{P}(1-\sqrt{P'})}}$$

$$A'^{-1} = c_a'\lambda = \frac{P'(1-P')}{P'-P} \ln\left(\frac{P'(1-P)}{(1-P')P}\right) \frac{P'-P}{L_{P'}-L_P}$$

$$= \frac{\beta p'(1-p') \ln\left(\frac{P'(1-P)}{(1-P')P}\right)}{\ln\frac{\sqrt{P'}(1-\sqrt{P})}{(1-\sqrt{P'})\sqrt{P}}}$$

$$k_t = \frac{\ln\left(\sqrt{P'}(1-\sqrt{p*})/(1-\sqrt{P'})\sqrt{p*}\right)}{\ln\left(\sqrt{P'}(1-\sqrt{P})/(1-\sqrt{P'})\sqrt{P}\right)}$$

$$\alpha = 2\beta p(1-\sqrt{P})$$

$$\alpha' = 2\beta p'(1-\sqrt{P'})$$

Case V13: If the true model is log-log and the assumed model is the logit model, then

$$V13 = MSE(\hat{L}^a_{p*})$$

$$= \frac{1}{n}\left(k_a^2\sigma^2A^2(2A\alpha-1)^{-1} + (1-k_a)^2\sigma'^2A'^2(2A'\alpha'-1)^{-1}\right)$$

$$+ (k_a-k_t)^2(L_{p'}-L_p)^2,$$

where

$$k_a = \frac{\ln\left(\frac{P'(1-p*)}{(1-P')p*}\right)}{\ln\left(\frac{P'(1-P)}{(1-P')P}\right)}$$

$$k_t = \frac{\ln\left(\frac{\ln(1-P')}{\ln(1-p*)}\right)}{\ln\left(\frac{\ln(1-P')}{\ln(1-P)}\right)}$$

$$L_p = \left(\ln\left(\ln\frac{1}{1-P}\right) - \mu\right)/\beta$$

$$L_{p'} = \left(\ln\left(\ln\frac{1}{1-P'}\right) - \mu\right)/\beta$$

$$A^{-1} = c_a\lambda = \frac{\beta p(1-p)\ln\left(\frac{P'(1-P)}{(1-P')P}\right)}{p' - p} \frac{p' - p}{L_{p'} - L_p}$$

$$= \frac{\beta p(1-p)\ln\left(\frac{P'(1-P)}{(1-P')P}\right)}{\ln\left(\frac{\ln(1-P')}{\ln(1-P)}\right)}$$

$$A'^{-1} = c_a'\lambda = \frac{\beta p'(1-p')\ln\left(\frac{P'(1-P)}{(1-P')P}\right)}{p' - p} \frac{p' - p}{L_{p'} - L_p}$$

$$= \frac{\beta p'(1-p')\ \ln\left(\frac{P'(1-P)}{(1-P')P}\right)}{\ln\left(\frac{\ln(1-P')}{\ln(1-P)}\right)}$$

$$\alpha = \beta(1-p)\ \ln\frac{1}{1-p}$$

$$\alpha' = \beta(1-p')\ \ln\frac{1}{1-p'}$$

Case V14 : If the true model is probit and the assumed model is the logit model, then

$$V14 = MSE(\hat{L}^{a}_{p*})$$

$$= \frac{1}{n}\left(k_a^2\sigma^2A^2(2A\alpha-1)^{-1} + (1-k_a)^2\sigma'^2A'^2(2A'\alpha'-1)^{-1}\right)$$

$$+ (k_a-k_t)^2(L_{p'}-L_p)^2$$

where

$$k_a = \frac{\ln\left(\frac{P'(1-p*)}{(1-P')p*}\right)}{\ln\left(\frac{P'(1-P)}{(1-P')P}\right)}$$

$$k_t = \frac{Fz^{-1}(P') - Fz^{-1}(P^*)}{Fz^{-1}(P') - Fz^{-1}(P)}$$

$$L_p = \mu + \beta F_z^{-1}(P)$$

$$L_{p'} = \mu + \beta F_z^{-1}(P')$$

$$A^{-1} = c_a\lambda = \frac{p(1-p)\ \ln\left(\frac{P'(1-P)}{(1-P')P}\right)}{p' - p}\quad \frac{p' - p}{L_{p'} - L_p}$$

$$= \frac{p(1-p)\ \ln\left(\frac{P'(1-P)}{(1-P')P}\right)}{\beta\left(Fz^{-1}(P' - Fz^{-1}(P)\right)}$$

$$A'^{-1} = c'_a\lambda = \frac{p'(1-p')\ \ln\left(\frac{P'(1-P)}{(1-P')P}\right)}{p' - p}\quad \frac{p' - p}{L_{p'} - L_p}$$

$$= \frac{p'(1-p')\ \ln\left(\frac{p'(1-p)}{(1-p')p}\right)}{\beta\left(F_z^{-1}(p') - F_z^{-1}(p)\right)}$$

$$\alpha = \frac{1}{\sqrt{2\pi}\ \beta} \exp\left(\frac{-1}{2}\left(F_z^{-1}(p)\right)^2\right)$$

$$\alpha' = \frac{1}{\sqrt{2\pi}\ \beta} \exp\left(\frac{-1}{2}\left(F_z^{-1}(p')\right)^2\right)$$

<u>Case V21</u> : If the true model is logit and the assumed model is the skewed logit model, then

$$V21 = MSE(\hat{L}^{a}_{p*})$$

$$= \frac{1}{n}\left(k_a^2\sigma^2A^2(2A\alpha-1)^{-1} + (1-k_a)^2\sigma'^2A'^2(2A'\alpha'-1)^{-1}\right)$$

$$+ (k_a-k_t)^2(L_{p'}-L_p)^2,$$

where

$$k_a = \frac{\ln\left(\sqrt{p'}(1-\sqrt{p*})/(1-\sqrt{p'})\sqrt{p*}\ \right)}{\ln\left(\sqrt{p'}(1-\sqrt{p})/(1-\sqrt{p'})\sqrt{p}\ \right)}$$

$$k_t = \frac{\ln\left(\frac{p'(1-p*)}{(1-p')p*}\right)}{\ln\left(\frac{p'(1-p)}{(1-p')p}\right)}$$

$$L_p = \left(\ln\frac{p}{1-p} - \mu\right)/\beta$$

$$L_{p'} = \left(\ln\frac{p'}{1-p'} - \mu\right)/\beta.$$

$$A^{-1} = c_a\lambda$$

$$= \frac{2p(1-\sqrt{p})\ \ln\left(\sqrt{p'}(1-\sqrt{p})/\sqrt{p}(1-\sqrt{p'}\right)}{p' - p} \quad \frac{p' - p}{L_{p'} - L_p}$$

$$= 2\beta p(1-\sqrt{P}) \frac{\ln\left(\sqrt{P'}(1-\sqrt{P})/(1-\sqrt{P'})\sqrt{P}\right)}{\ln\left(\frac{P'(1-P)}{(1-P')P}\right)}$$

$$A'^{-1} = c_a'\lambda$$

$$= \frac{2p'(1-\sqrt{P'})\ \ln\left(\sqrt{P'}(1-\sqrt{P})/(1-\sqrt{P'})\sqrt{P}\right)}{p' - p} \frac{p' - p}{L_{p'} - L}$$

$$= 2\beta p'(1-\sqrt{P'}) \frac{\ln\left(\sqrt{P'}(1-\sqrt{P})/(1-\sqrt{P'})\sqrt{P}\right)}{\ln\left(\frac{P'(1-P)}{(1-P')P}\right)}$$

$$\alpha = \beta p(1-p)$$

$$\alpha' = \beta p'(1-p')$$

<u>Case V22</u>: If the assumed and true models are skewed logit, then

$$V22 = MSE(\hat{L}^a_{p*})$$

$$= \frac{1}{n}\left(k_a^2\sigma^2A^2(2A\alpha-1)^{-1} + (1-k_a)^2\sigma'^2A'^2(2A'\alpha'-1)^{-1}\right)$$

$$+ (k_a-k_t)^2(L_{p'}-L_p)^2,$$

where

$$k_a = k_t = \frac{\ln\left(\sqrt{P'}(1-\sqrt{p*})/(1-\sqrt{P'})\sqrt{p*}\right)}{\ln\left(\sqrt{P'}(1-\sqrt{P})/(1-\sqrt{P'})\sqrt{P}\right)}$$

$$L_p = \left(\ln\left(\frac{\sqrt{P}}{1-\sqrt{P}}\right) - \mu\right)/\beta$$

$$L_{p'} = \left(\ln\left(\frac{\sqrt{P'}}{1-\sqrt{P'}}\right) - \mu\right)/\beta$$

$$A^{-1} = \alpha = c_a\lambda = 2\beta p(1-\sqrt{P})$$

$$A'^{-1} = \alpha' = c_a'\lambda = 2\beta p'(1-\sqrt{P'})$$

<u>Case V23</u> : If the true model is log-log and the assumed

model is the skewed logit model, then

$$V23 = MSE(\hat{L}^{a}_{p*})$$

$$= \frac{1}{n}\left(k_a^2\sigma^2A^2(2A\alpha-1)^{-1} + (1-k_a)^2\sigma'^2A'^2(2A'\alpha'-1)^{-1}\right)$$

$$+ (k_a-k_t)^2(L_{p'}-L_p)^2,$$

where

$$k_a = \frac{\ln\left(\sqrt{P'}(1-\sqrt{p*})/(1-\sqrt{P'})\sqrt{p*}\right)}{\ln\left(\sqrt{P'}(1-\sqrt{P})/(1-\sqrt{P'})\sqrt{P}\right)}$$

$$k_t = \frac{\ln\left(\frac{\ln(1-P')}{\ln(1-p*)}\right)}{\ln\left(\frac{\ln(1-P')}{\ln(1-P)}\right)}$$

$$L_p = \left(\ln\left(\ln\frac{1}{1-P}\right) - \mu\right)/\beta$$

$$L_{p'} = \left(\ln\left(\ln\frac{1}{1-P'}\right) - \mu\right)/\beta$$

$$A^{-1} = c_a\lambda$$

$$= \frac{2p(1-\sqrt{P})\ \ln\left(\sqrt{P'}(1-\sqrt{P})/(1-\sqrt{P'})\sqrt{P}\right)}{p' - p}\ \frac{p' - p}{L_{p'} - L_p}$$

$$= 2\beta p(1-\sqrt{P})\ \frac{\ln\left(\sqrt{P'}(1-\sqrt{P})/(1-\sqrt{P'})\sqrt{P}\right)}{\ln\left(\frac{\ln(1-P')}{\ln(1-P)}\right)}$$

$$A'^{-1} = c'_a\lambda$$

$$= \frac{2p'(1-\sqrt{P'})\ \ln\left(\sqrt{P'}(1-\sqrt{P})/(1-\sqrt{P'})\sqrt{P}\right)}{p' - p}\ \frac{p' - p}{L_{p'} - L}$$

$$= 2\beta p'(1-\sqrt{P'})\ \frac{\ln\left(\sqrt{P'}(1-\sqrt{P})/(1-\sqrt{P'})\sqrt{P}\right)}{\ln\left(\frac{\ln(1-P')}{\ln(1-P)}\right)}$$

$$\alpha = \beta(1-p)\ \ln\frac{1}{1-P}$$

$$\alpha' = \beta(1-p')\ \ln\frac{1}{1-P'}$$

<u>Case V24</u> : If the true model is probit and the assumed model is the skewed logit model, then

$$V24 = MSE(\hat{L}^{a}_{p*})$$

$$= \frac{1}{n}\left(k_a^2\sigma^2A^2(2A\alpha-1)^{-1} + (1-k_a)^2\sigma'^2A'^2(2A'\alpha'-1)^{-1}\right)$$

$$+ (k_a-k_t)^2(L_{p'}-L_p)^2,$$

where

$$k_a = \frac{\ln\left(\sqrt{P'}(1-\sqrt{P*})/(1-\sqrt{P'})\sqrt{P*}\right)}{\ln\left(\sqrt{P'}(1-\sqrt{P})/(1-\sqrt{P'})\sqrt{P}\right)}$$

$$k_t = \frac{Fz^{-1}(P') - Fz^{-1}(P^*)}{Fz^{-1}(P') - Fz^{-1}(P)}$$

$$L_P = \mu + \beta F_z^{-1}(P)$$

$$L_{P'} = \mu + \beta F_z^{-1}(P')$$

$$A^{-1} = c_a\lambda$$

$$= \frac{2p(1-\sqrt{P})\ \ln\left(\sqrt{P'}(1-\sqrt{P})/\sqrt{P}(1-\sqrt{P'})\right)}{p' - p}\ \frac{p' - p}{L_{P'} - L_P}$$

$$= 2p(1-\sqrt{P})\ \frac{\ln\left(\sqrt{P'}(1-\sqrt{P})/(1-\sqrt{P'})\sqrt{P}\right)}{\beta\left(Fz^{-1}(P') - Fz^{-1}(P)\right)}$$

$$A'^{-1} = c_a'\lambda$$

$$= \frac{2p'(1-\sqrt{p'})\ \ln\left(\sqrt{p'}(1-\sqrt{p})/\sqrt{p}(1-\sqrt{p'})\right)}{p' - p}\ \frac{p' - p}{L_{p'} - L_p}$$

$$= 2p'(1-\sqrt{p'})\ \frac{\ln\left(\sqrt{p'}(1-\sqrt{p})/(1-\sqrt{p'})\sqrt{p}\ \right)}{\beta\left(Fz^{-1}(p') - Fz^{-1}(p)\right)}$$

$$\alpha = \frac{1}{\sqrt{2\pi}\ \beta}\ \exp\left(\frac{-1}{2}\left(Fz^{-1}(p)\right)^2\right)$$

$$\alpha' = \frac{1}{\sqrt{2\pi}\ \beta}\ \exp\left(\frac{-1}{2}\left(Fz^{-1}(p')\right)^2\right)$$

<u>Case V31</u> : If the true model is logit and the assumed model is the log-log model, then

$$V31 = MSE(\hat{L}^a_{p*})$$

$$= \frac{1}{n}\left(k_a^2\sigma^2A^2(2A\alpha-1)^{-1} + (1-k_a)^2\sigma'^2A'^2(2A'\alpha'-1)^{-1}\right)$$

$$+ (k_a-k_t)^2(L_{p'}-L_p)^2,$$

where

$$k_a = \frac{\ln\left(\frac{\ln(1-p')}{\ln(1-p*)}\right)}{\ln\left(\frac{\ln(1-p')}{\ln(1-p)}\right)}$$

$$k_t = \frac{\ln\left(\frac{p'(1-p*)}{(1-p')p*}\right)}{\ln\left(\frac{p'(1-p)}{(1-p')p}\right)}$$

$$L_p = \left(\ln\frac{p}{1-p} - \mu\right)/\beta$$

$$L_{p'} = \left(\ln\frac{p'}{1-p'} - \mu\right)/\beta.$$

$$A^{-1} = c_a\lambda$$

$$= \frac{1-p}{p'-p}\ \ln\left(\frac{1}{1-p}\right)\ \ln\left(\frac{\ln(1-p')}{\ln(1-p)}\right)\ \frac{p'-p}{L_{p'}-L_p}$$

$$= \beta(1-p)\ \ln\left(\frac{\ln(1-P')}{\ln(1-P)}\right)\ \frac{\ln\left(\frac{1}{1-P}\right)}{\ln\left(\frac{P'(1-P)}{(1-P')P}\right)}$$

$$A'^{-1} = c_a'\lambda$$

$$= \frac{1-p'}{p'-p}\ \ln\left(\frac{1}{1-p'}\right)\ \ln\left(\frac{\ln(1-p')}{\ln(1-p)}\right)\ \frac{p'-p}{L_{p'}-L_p}$$

$$= \beta(1-p')\ \ln\left(\frac{\ln(1-P')}{\ln(1-P)}\right)\ \frac{\ln\left(\frac{1}{1-P'}\right)}{\ln\left(\frac{P'(1-P)}{(1-p')P}\right)}$$

$$\alpha = \beta p(1-p)$$

$$\alpha' = \beta p'(1-p')$$

<u>Case V32</u> : If the true model is skewed logit and the assumed model is the log-log model, then

$$V32 = MSE(\hat{L}^a_{p*})$$

$$= \frac{1}{n}\left(k_a^2\sigma^2A^2(2A\alpha-1)^{-1} + (1-k_a)^2\sigma'^2A'^2(2A'\alpha'-1)^{-1}\right)$$

$$+ (k_a-k_t)^2(L_{p'}-L_p)^2,$$

where

$$k_a = \frac{\ln\left(\frac{\ln(1-p')}{\ln(1-p*)}\right)}{\ln\left(\frac{\ln(1-p')}{\ln(1-p)}\right)}$$

$$k_t = \frac{\ln\left(\sqrt{P'}(1-\sqrt{p*})/(1-\sqrt{P'})\sqrt{p*}\right)}{\ln\left(\sqrt{P'}(1-\sqrt{P})/(1-\sqrt{P'})\sqrt{P}\right)}$$

$$L_p = \left(\ln\left(\frac{\sqrt{P}}{1-\sqrt{P}}\right) - \mu\right)/\beta$$

$$L_{p'} = \left(\ln\left(\frac{\sqrt{P'}}{1-\sqrt{P'}}\right) - \mu\right)/\beta$$

$$A^{-1} = c_a\lambda$$

$$= \frac{1-P}{P'-P} \ln\left(\frac{1}{1-P}\right) \ln\left(\frac{\ln(1-P')}{\ln(1-P)}\right) \frac{P'-P}{L_{p'}-L_p}$$

$$= \beta(1-p) \ln\left(\frac{1}{1-P}\right) \frac{\ln\left(\frac{\ln(1-P')}{\ln(1-P)}\right)}{\ln\left(\sqrt{P'}(1-\sqrt{P})/(1-\sqrt{P'})\sqrt{P}\right)}$$

$$A'^{-1} = c'_a\lambda$$

$$= \frac{1-P'}{P'-P} \ln\left(\frac{1}{1-P'}\right) \ln\left(\frac{\ln(1-P')}{\ln(1-P)}\right) \frac{P'-P}{L_{p'}-L_p}$$

$$= \beta(1-p') \ln\left(\frac{1}{1-P'}\right) \frac{\ln\left(\frac{\ln(1-P')}{\ln(1-P)}\right)}{\ln\left(\sqrt{P'}(1-\sqrt{P})/(1-\sqrt{P'})\sqrt{P}\right)}$$

$$\alpha = 2\beta p(1-\sqrt{P})$$

$$\alpha' = 2\beta p'(1-\sqrt{P'})$$

<u>Case V33</u> : If the true and assumed models are log-log, then

$$V33 = MSE(\hat{L}^a_{p*})$$

$$= \frac{1}{n}\left(k_a^2\sigma^2A^2(2A\alpha-1)^{-1} + (1-k_a)^2\sigma'^2A'^2(2A'\alpha'-1)^{-1}\right)$$

$$+ (k_a-k_t)^2(L_{p'}-L_p)^2,$$

where

$$k_a = k_t = \frac{\ln\left(\frac{\ln(1-P')}{\ln(1-p*)}\right)}{\ln\left(\frac{\ln(1-P')}{\ln(1-P)}\right)}$$

$$L_p = \left(\ln\left(\ln\frac{1}{1-P}\right) - \mu\right)/\beta$$

$$L_{p'} = \left(\ln\left(\ln\frac{1}{1-P'}\right) - \mu\right)/\beta$$

$$A^{-1} = c_a\lambda = \alpha = \beta(1-p) \ln\frac{1}{1-P}$$

$$A'^{-1} = c'_a\lambda = \alpha' = \beta(1-p') \ln\frac{1}{1-P'}$$

<u>Case V34</u> : If the true model is probit and the assumed model is the log-log model, then

$$V34 = MSE(\hat{L}^{a}_{p*})$$

$$= \frac{1}{n}\left(k_a^2\sigma^2A^2(2A\alpha-1)^{-1} + (1-k_a)^2\sigma'^2A'^2(2A'\alpha'-1)^{-1}\right)$$

$$+ (k_a-k_t)^2(L_{p'}-L_p)^2,$$

where

$$k_a = \frac{\ln\left(\frac{\ln(1-P')}{\ln(1-p*)}\right)}{\ln\left(\frac{\ln(1-P')}{\ln(1-P)}\right)}$$

$$k_t = \frac{Fz^{-1}(P') - Fz^{-1}(P^*)}{Fz^{-1}(P') - Fz^{-1}(P)}$$

$$L_p = \mu + \beta F_z^{-1}(P)$$

$$L_{p'} = \mu + \beta F_z^{-1}(P')$$

$$A^{-1} = c_a\lambda$$

$$= \frac{1-P}{P'-P} \ln\left(\frac{1}{1-P}\right) \ln\left(\frac{\ln(1-P')}{\ln(1-P)}\right) \frac{P'-P}{L_{p'}-L_p}$$

$$= (1-p) \ln\left(\frac{1}{1-P}\right) \frac{\ln\left(\frac{\ln(1-P')}{\ln(1-P)}\right)}{\beta\left(Fz^{-1}(P') - Fz^{-1}(P)\right)}$$

$$A'^{-1} = c'_a\lambda$$

$$= \frac{1-P'}{P'-P} \ln\left(\frac{1}{1-P'}\right) \ln\left(\frac{\ln(1-P')}{\ln(1-P)}\right) \frac{P'-P}{L_{p'}-L_p}$$

$$= (1-p') \ln\left(\frac{1}{1-P'}\right) \frac{\ln\left(\frac{\ln(1-P')}{\ln(1-P)}\right)}{\beta\left(Fz^{-1}(P') - Fz^{-1}(P)\right)}$$

$$\alpha = \frac{1}{\sqrt{2\pi}\,\beta} \exp\left(\frac{-1}{2}\left(Fz^{-1}(P)\right)^2\right)$$

$$\alpha' = \frac{1}{\sqrt{2\pi}\ \beta} \exp\left(\frac{-1}{2}\left(Fz^{-1}(P')\right)^2\right)$$

<u>Case V41</u> : If the true model is logit and the assumed model is the probit model, then

$$V41 = MSE(\hat{L}^{a}_{p*})$$

$$= \frac{1}{n}\left(k_a^2\sigma^2A^2(2A\alpha-1)^{-1} + (1-k_a)^2\sigma'^2A'^2(2A'\alpha'-1)^{-1}\right)$$

$$+ (k_a-k_t)^2(L_{P'}-L_P)^2,$$

where

$$k_a = \frac{Fz^{-1}(P') - Fz^{-1}(P^*)}{Fz^{-1}(P') - Fz^{-1}(P)}$$

$$k_t = \frac{\ln\left(\frac{P'(1-p*)}{(1-P')p*}\right)}{\ln\left(\frac{(1-P)P'}{(1-P')P}\right)}$$

$$L_P = \left(\ln\frac{P}{1-P} - \mu\right)/\beta$$

$$L_{P'} = \left(\ln\frac{p'}{1-p'} - \mu\right)/\beta.$$

$$A^{-1} = c_a\lambda$$

$$= \frac{Fz^{-1}(P') - Fz^{-1}(P)}{\sqrt{2\pi}\ (P'-P)} \exp\left(\frac{-1}{2}\left(F_Z^{-1}(P)\right)^2\right) \frac{p'-p}{L_{P'}-L_P}$$

$$= \frac{\beta\left(Fz^{-1}(P') - Fz^{-1}(P)\right)}{\sqrt{2\pi}\ \ln\left(\frac{P'(1-P)}{(1-p')P}\right)} \exp\left(\frac{-1}{2}\left(F_Z^{-1}(P)\right)^2\right)$$

$$A'^{-1} = c'_a\lambda$$

$$= \frac{Fz^{-1}(P') - Fz^{-1}(P)}{\sqrt{2\pi}\ (P'-P)} \exp\left(\frac{-1}{2}\left(F_Z^{-1}(P')\right)^2\right) \frac{P'-P}{L_{P'}-L}$$

$$= \frac{\beta\left(Fz^{-1}(P') - Fz^{-1}(P)\right)}{\sqrt{2\pi}\ \ln\left(\frac{P'(1-P)}{(1-P')P}\right)} \exp\left(\frac{-1}{2}\left(F_Z^{-1}(P')\right)^2\right)$$

$$\alpha = \beta p(1-p)$$

$$\alpha' = \beta p'(1-p')$$

<u>Case V42</u> : If the true model is skewed logit and the assumed model is the probit model. Then

$$V42 = MSE(\hat{L}^{a}_{p*})$$

$$= \frac{1}{n}\left(k_a^2\sigma^2A^2(2A\alpha-1)^{-1} + (1-k_a)^2\sigma'^2A'^2(2A'\alpha'-1)^{-1}\right)$$

$$+ (k_a-k_t)^2(L_{p'}-L_p)^2,$$

where

$$k_a = \frac{Fz^{-1}(P') - Fz^{-1}(P^*)}{Fz^{-1}(P') - Fz^{-1}(P)}$$

$$k_t = \frac{\ln\left(\sqrt{P'}(1-\sqrt{p*})/(1-\sqrt{P'})\sqrt{p*}\right)}{\ln\left(\sqrt{P'}(1-\sqrt{P})/(1-\sqrt{P'})\sqrt{P}\right)}$$

$$L_p = \left(\ln\left(\sqrt{P}(1-\sqrt{P})^{-1}\right) - \mu\right)/\beta$$

$$L_{p'} = \left(\ln\left(\sqrt{P'}(1-\sqrt{P'})^{-1}\right) - \mu\right)/\beta$$

$$A^{-1} = c_a\lambda$$

$$= \frac{Fz^{-1}(P') - Fz^{-1}(P)}{\sqrt{2\pi}\ (p'-p)} \exp\left(\frac{-1}{2}\left(F_Z^{-1}(P)\right)^2\right) \frac{P'-P}{L_{p'}-L_p}$$

$$= \frac{\beta\left(Fz^{-1}(P') - Fz^{-1}(P)\right)}{\sqrt{2\pi}\ \ln\left(\sqrt{P'}(1-\sqrt{P})/(1-\sqrt{P'})\sqrt{P}\right)} \exp\left(\frac{-1}{2}\left(F_Z^{-1}(P)\right)^2\right)$$

$$A'^{-1} = c_a'\lambda$$

$$= \frac{Fz^{-1}(P') - Fz^{-1}(P)}{\sqrt{2\pi}\ (P'-P)} \exp\left(\frac{-1}{2}\left(F_Z^{-1}(P')\right)^2\right) \frac{P'-P}{L_{p'}-L_p}$$

$$= \frac{\beta\left(Fz^{-1}(P') - Fz^{-1}(P)\right)}{\sqrt{2\pi}\ \ln\left(\sqrt{P'}(1-\sqrt{P})/(1-\sqrt{P'})\sqrt{P}\right)} \exp\left(\frac{-1}{2}\left(F_Z^{-1}(P')\right)^2\right)$$

$$\alpha = 2\beta p(1-\sqrt{P})$$

$$\alpha' = 2\beta p'(1-\sqrt{P'})$$

Case V43 : If the true model is log-log and the assumed model is the probit model, then

$$V43 = MSE(\hat{L}^{a}_{p*})$$

$$= \frac{1}{n}\left(k_a^2\sigma^2A^2(2A\alpha-1)^{-1} + (1-k_a)^2\sigma'^2A'^2(2A'\alpha'-1)^{-1}\right)$$

$$+ (k_a-k_t)^2(L_{P'}-L_P)^2$$

where

$$k_a = \frac{Fz^{-1}(P') - Fz^{-1}(P^*)}{Fz^{-1}(P') - Fz^{-1}(P)}$$

$$k_t = \frac{\ln\left(\frac{\ln(1-P')}{\ln(1-p*)}\right)}{\ln\left(\frac{\ln(1-P')}{\ln(1-P)}\right)}$$

$$L_P = \left(\ln\left(\ln\frac{1}{1-P}\right) - \mu\right)/\beta$$

$$L_{P'} = \left(\ln\left(\ln\frac{1}{1-P'}\right) - \mu\right)/\beta$$

$$A^{-1} = c_a\lambda$$

$$= \frac{Fz^{-1}(P') - Fz^{-1}(P)}{\sqrt{2\pi}\ (P'-P)} \exp\left(\frac{-1}{2}\left(F_Z^{-1}(P)\right)^2\right) \frac{P'-P}{L_{P'}-L_P}$$

$$= \frac{\beta\left(Fz^{-1}(P') - Fz^{-1}(P)\right)}{\sqrt{2\pi}\ \ln\left(\frac{\ln(1-P')}{\ln(1-P)}\right)} \exp\left(\frac{-1}{2}\left(F_Z^{-1}(P)\right)^2\right)$$

$$A'^{-1} = c_a'\lambda$$

$$= \frac{Fz^{-1}(P') - Fz^{-1}(P)}{\sqrt{2\pi}\ (p'-p)} \exp\left(\frac{-1}{2}\left(F_Z^{-1}(P')\right)^2\right) \frac{P'-P}{L_{P'}-L_P}$$

$$= \frac{\beta\left(Fz^{-1}(P') - \bar{F}z^{-1}(P)\right)}{\sqrt{2\pi}\ \ln\left(\frac{\ln(1-P')}{\ln(1-P)}\right)} \exp\left(\frac{-1}{2}\left(F_Z^{-1}(P')\right)^2\right)$$

$$\alpha = \beta(1-p)\ \ln\frac{1}{1-P}$$

$$\alpha' = \beta(1-p')\ \ln\frac{1}{1-P'}$$

<u>Case V44</u> : If the true and the assumed models are the probit models, then

$$V44 = MSE(\hat{L}^{a}_{p*})$$

$$= \frac{1}{n}\left(k_a^2\sigma^2A^2(2A\alpha-1)^{-1} + (1-k_a)^2\sigma'^2A'^2(2A'\alpha'-1)^{-1}\right)$$

$$+ (k_a-k_t)^2(L_{p'}-L_p)^2,$$

where

$$k_a = k_t = \frac{Fz^{-1}(P') - Fz^{-1}(P^*)}{Fz^{-1}(P') - Fz^{-1}(P)}$$

$$L_p = \mu + \beta F_z^{-1}(P)$$

$$L_{p'} = \mu + \beta F_z^{-1}(P')$$

$$A^{-1} = c_a\lambda = \alpha = \frac{1}{\sqrt{2\pi}\ \beta}\ \exp\left(\frac{-1}{2}\left(Fz^{-1}(P)\right)^2\right)$$

$$A'^{-1} = c_a'\lambda = \alpha' = \frac{1}{\sqrt{2\pi}\ \beta}\ \exp\left(\frac{-1}{2}\left(Fz^{-1}(P')\right)^2\right).$$

Minimax and Bayes Rules for Selecting Optimal p and p′

Since the asymptotic variance of the estimator $\hat{L}_{p*}$ is a function of p, p′, p^* and the parameters of the true model, it is of interest to find the optimal values of p and p′ such that the asymptotic variance of $\hat{L}_{p*}$ is minimized for a given p^*, where $r \le p^* \le 1-r$ and

$0 < r < \frac{1}{2}$. That is, for a given percentile range, it is of interest to find the pair (p, p′) such that the minimum asymptotic variance of $\hat{L}_{p*}$ is attained.

Two criteria, a minimax and a Bayes criterion, are considered. Under the minimax criterion, the maximum asymptotic variance of $\hat{L}_{p*}$ is chosen in the range (r, 1-r) for each fixed pair (p, p′). The minimax rule is the pair (p, p′) that has the smallest maximum asymptotic variance. Because the minimax criterion is a conservative criteria, the asymptotic variance of $\hat{L}_{p*}$ under this criterion may be unduly large for some ranges (r, 1-r).

The Bayes criterion is another option. It is of interest to estimate all the roots of M in the range (r, 1-r). Let $\pi(p^*)$ be the prior distribution of p^*, which represents the level of interest in a specific root p^*. If all roots are of equal interest, then $\pi(p^*)$ is the continuous uniform distribution U(r, 1-r). Let $W(p,p',p^*)$ be the asymptotic variance of $\sqrt{n}(\hat{L}_{p*}-L_{p*})$. The optimal value of (p, p′) is chosen such that $\int_r^{1-r} W(p,p',p^*)\pi(p^*)\ dp^*$ is minimized.

For different values of r, the optimal values of p of logit model for minimax and Bayes criteria are listed in Table 3.2. Since the logit model is symmetric around p = 0.5, optimal (p, p′) are calculated under the restriction p′ = 1-p. From Table 3.2, the minimum asymptotic variances of $\sqrt{n}(\hat{L}_{p*}-L_{p*})$ under minimax criterion are much larger than that of Bayes criterion. From Figure 3.1, for a given r, the range (p,p') for minimax criterion is wider than that for Bayes criterion. That is, for the same range of p^*, the minimax criterion selects a wider range (p,p') to estimate the whole curve M than the Bayes criterion does.

For the Bayes criterion, if r = 0.1 and the model is logit, the optimal (p,p') is about (0.2, 0.8). For the other three binary data models, the optimal values of (p,p') can be found in a similar way. If r = 0.1, for the skewed logit model, the optimal (p,p') is about (0.07, 0.66); for the log-log model, it is about (0.14, 0.70); for the probit model, it is about (0.174, 0.826).

The MSE's of Case V_{11} to Case V_{44} in Table 3.1 are the functions of p, p', p^*, and the parameters of the true model. Thus, the idea of optimal p and p' values is now merged with the concept of robustness. For r = 0.1 and p^* = 0.3, 0.5, 0.75, the numerical values of MSE of $\hat{L}_{p*}$ in Case V_{11} to Case V_{44} are listed in Table 3.3. In Table 3.3, the optimal (p,p') of each model are used in the corresponding assumed models. For simplification, the two parameters (μ,β) of the true models are assumed to be (0,1) in both tables. The cases where the parameters (μ,β) are not equal to (0,1) or the range of p^* is not (0.1,0.9) can be calculated in a similar way.

For the diagonal elements in Table 3.3, the assumed and the true moddels are the same. Thus, the MSE does not contain the bias term and, therefore, is the variance of $\hat{L}_{p*}$. However, for the off-diagonal elements, the MSE is the sum of variance and bias of $\hat{L}_{p*}$, where the variance is a multiple of n^{-1}.

In table 3.3, it is also noted that, for a true model, the MSE from the correct assumed model (i.e. the diagonal elements) is not always less than the MSE's from the wrong assumed models (i.e. the off-diagonal elements in the same column). However, for large n, the MSE from the correct assumed model will smaller than the MSE's from the wrong assumed models due to smaller variances. For example, if $p^*=0.3$ and the true model is logit, the MSE from the logit assumed model is 4.2924/n; the MSE from the log-log assumed model is 3.8491/n + 0.0182. The former is larger than the latter for small n (e.g. n=10). However, for large n (e.g. n=100), the former is less than the latter.

TABLE 3.2

THE OPTIMAL (P,P') AND MINIMUM ASYMPTOTIC VARIANCE FOR MINIMAX AND BAYES CRITERIA IN THE RANGE (r,1-r) OF p* FOR THE LOGIT MODEL

r	Minimax Criterion		Bayesian Criterion	
	P	Minimum Variance	P	Minimum Variance
0.02	0.115	22.82	0.176	6.62
0.05	0.131	15.05	0.188	5.41
0.08	0.142	11.67	0.200	4.55
0.10	0.149	10.23	0.206	4.07
0.15	0.165	7.77	0.222	3.10
0.20	0.182	6.21	0.239	2.35
0.25	0.206	5.09	0.258	1.74
0.30	0.235	4.20	0.282	1.25
0.35	0.253	3.53	0.304	0.85
0.40	0.292	2.95	0.340	0.5
0.45	0.349	2.41	0.383	0.25
0.48	0.403	2.19	0.421	0.08

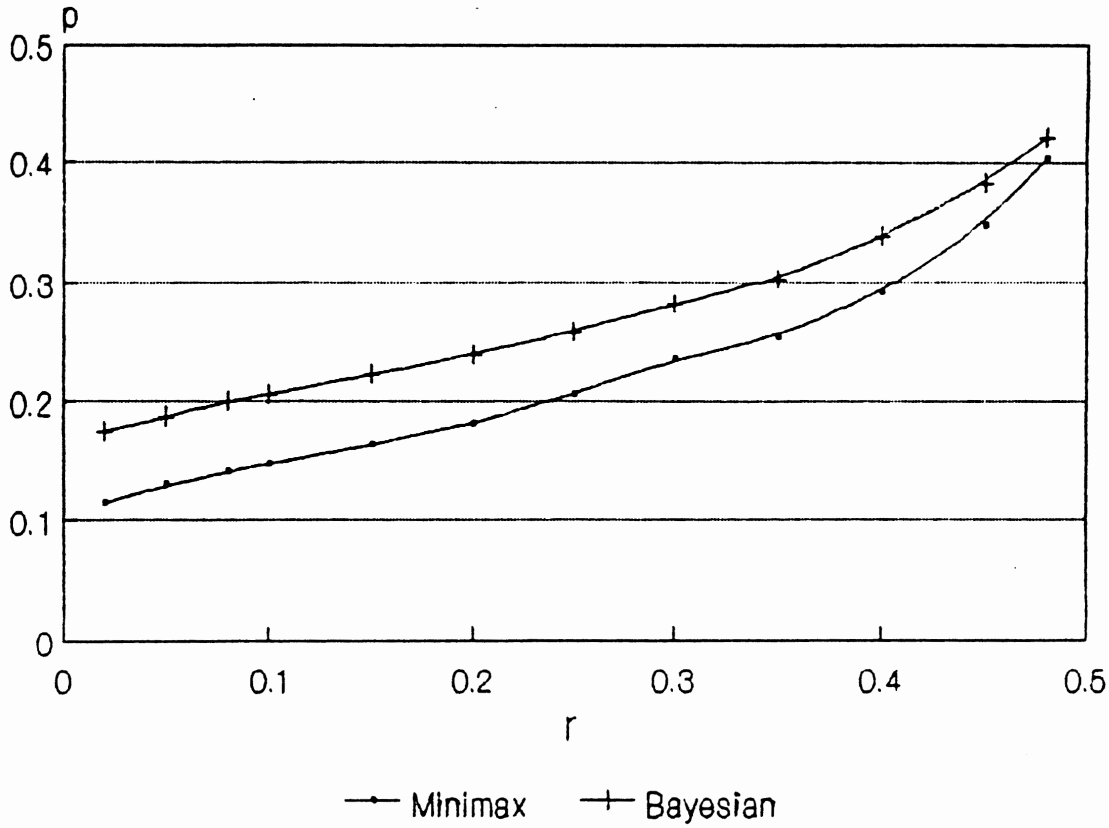

Figure 3.1 Optimal Values of P for Minimax and Bayesian Criteria Based on Logit Model

TABLE 3.3

THE ASYMPTOTIC MSE OF $\hat{L}_{p*}$ WITH OPTIMAL (p, p') FOR EACH MODEL

p* = 0.3	True Model $(\mu, \beta) = (0, 1)$			
Assumed Model	Logit	Skewed Logit	Log-Log	Probit
Logit	$\frac{4.2924}{n}$	$\frac{2.2108}{n} + 0.0028$	$\frac{3.5553}{n} + 0.0072$	$\frac{1.4071}{n} + 0.0001$
Skewed Logit	$\frac{5.3026}{n} + 0.0236$	$\frac{2.4816}{n}$	$\frac{5.5602}{n} + 0.0395$	$\frac{1.3494}{n} + 0.0009$
Log-Log	$\frac{3.8491}{n} + 0.0182$	$\frac{1.3310}{n} + 0.0521$	$\frac{2.7312}{n}$	$\frac{1.2251}{n} + 0.0333$
Probit	$\frac{4.5986}{n} + 0.0006$	$\frac{2.4702}{n} + 0.0030$	$\frac{4.1823}{n} + 0.0187$	$\frac{1.4297}{n}$

TABLE 3.3 (Continued)

p* = 0.5	True Model (μ, β) = (0, 1)			
Assumed Model	Logit	Skewed Logit	Log-Log	Probit
Logit	$\frac{3.1250}{n}$	$\frac{2.3852}{n} + 0.0066$	$\frac{1.7697}{n} + 0.0221$	$\frac{1.0244}{n}$
Skewed Logit	$\frac{3.5600}{n} + 0.0100$	$\frac{2.4810}{n}$	$\frac{2.0829}{n} + 0.0367$	$\frac{1.1897}{n} + 0.0574$
Log-Log	$\frac{3.5087}{n} + 0.0195$	$\frac{3.0756}{n} + 0.0505$	$\frac{1.3733}{n}$	$\frac{1.2721}{n} + 0.0266$
Probit	$\frac{3.5050}{n}$	$\frac{2.5489}{n} + 0.0103$	$\frac{1.1102}{n} + 0.0331$	$\frac{1.0908}{n}$

TABLE 3.3 (Continued)

p* = 0.75	True Model $(\mu, \beta) = (0, 1)$			
Assumed Model	Logit	Skewed Logit	Log-Log	Probit
Logit	$\frac{5.0872}{n}$	$\frac{4.6047}{n} + 0.0007$	$\frac{1.4231}{n} + 0.0031$	$\frac{1.6677}{n} + 0.0001$
Skewed Logit	$\frac{5.5717}{n} + 0.0078$	$\frac{5.0919}{n}$	$\frac{2.6700}{n} + 0.1513$	$\frac{2.2942}{n} + 0.0018$
Log-Log	$\frac{6.1532}{n} + 0.0050$	$\frac{6.3478}{n} + 0.0116$	$\frac{1.8687}{n}$	$\frac{2.2615}{n} + 0.0046$
Probit	$\frac{5.1627}{n} + 0.0005$	$\frac{4.9862}{n} + 0.0041$	$\frac{1.3810}{n} + 0.0063$	$\frac{1.6532}{n}$

CHAPTER IV

A SIMULATION STUDY

In this chapter, the Monte Carlo mean square errors from Robbins-Monro's procedure, Anbar's procedure, Wu's procedure, and this new procedure are compared.

Simulation Outline

Under comparison are Robbins-Monro's two root independent estimation procedure with n observations each (called RM procedure), Anbar's one root estimation procedure with 2n observations (called Anbar's one root procedure), Anbar's two root independent estimation procedure with n observations each (called Anbar's 2-root procedure), Wu's one root estimation procedure with 2n observations (called Wu's one root procedure), Wu's two root independent estimation procedure with n observations each (called Wu's 2-root procedure), and this new procedure with n observations each.

For the RM procedure, (x_{n+1}, x'_{n+1}) are calculated by

$$\begin{pmatrix} x_{n+1} \\ x'_{n+1} \end{pmatrix} = \begin{pmatrix} x_n \\ x'_n \end{pmatrix} - \frac{A}{n} \begin{pmatrix} y_n - p \\ y'_n - p' \end{pmatrix} \qquad (4.1)$$

where n = 1, 2,....

Wetherill (1963) showed that the RM procedure in (1.9) with large A is less susceptible to a poor choice of x_1, especially for small samples. Thus, three levels of A — 1, 6, and 36, were used in the simulations.

For Anbar's one root procedure, x_{n+1} is calculated by

$$x_{n+1} = x_n - n^{-1}A_{mn}(y_n-p) \tag{4.2}$$

where A_{mn} is defined by (1.5) to (1.7).

For Anbar's 2-root procedure, x_{n+1} and x'_{n+1} are calculated independently by

$$\begin{pmatrix} x_{n+1} \\ x'_{n+1} \end{pmatrix} = \begin{pmatrix} x_n \\ x'_n \end{pmatrix} - \frac{1}{n} \begin{pmatrix} A_{mn}(y_n-p) \\ A'_{mn}(y'_n-p') \end{pmatrix} \tag{4.3}$$

where both A_{mn} and A'_{mn} are defined by (1.5) to (1.7).

For Wu's one root procedure, x_{n+1} is calculated by

$$x_{n+1} = x_n - n^{-1}d^*_n(y_n-p) \tag{4.4}$$

where d^*_n is defined by (1.10) and (1.11).

For Wu's 2-root procedure, x_{n+1} and x'_{n+1} are calculated independently by

$$\begin{pmatrix} x_{n+1} \\ x'_{n+1} \end{pmatrix} = \begin{pmatrix} x_n \\ x'_n \end{pmatrix} - \frac{1}{n} \begin{pmatrix} d^*_n(y_n-p) \\ d'^*_n(y'_n-p') \end{pmatrix} \tag{4.5}$$

where both d^*_n and d'^*_n are defined by (1.10) and (1.11).

For the new procedure, (x_{n+1}, x'_{n+1}) is calculated by

$$\begin{pmatrix} x_{n+1} \\ x'_{n+1} \end{pmatrix} = \begin{pmatrix} x_n \\ x'_n \end{pmatrix} - \begin{pmatrix} a_n \ (y_n - p) \\ a'_n \ (y'_n - p') \end{pmatrix} \tag{4.6}$$

where a_n and a'_n are defined by (1.17) to (1.20).

In Wu's and Anbar's papers, the estimators of the tangent slopes of M are constrained by finite positive constants. Thus, four pairs of bounded values (δ_1, δ_2), (0.005, 36), (0.005, 50), (0.005, 100), and (0.005, 200), for the estimators of the inverse tangent slopes of M were used in Anbar's, Wu's, and the new procedures.

The convergence speed is an important criterion to evaluate a stochastic approximation procedure. Thus, four sample sizes, n = 15, 30, 50, and 100, were used in the simulations.

Four different 2-parameter models, the logit model, the skewed logit model, the log-log model, and the probit model, are used to generate the binary observations. In each case, the model used to generate the observations represents the true model. The two parameters (μ, β) of the true model are derived such that $M(0) = 0.5$ and $\frac{\partial}{\partial x} M(0) = 0.25$. Thus, for logit model, (μ, β) is (0, 1); for skewed logit model, (μ, β) is about (0.8814, 0.8536); for log-log model, (μ, β) is about (-0.3665, 0.7213); for probit model, (μ, β) is about (0, 1.5958).

For any true model, the logit model is used as the assumed model. Therefore, the MLE's for Wu's procedure are calculated from a logit model. Also, for the new procedure, all estimators of roots and parameters are calculated using the logit model equations (2.10) to (2.21).

Since the assumed model is the logit model, if the range of p^* is (0.1, 0.9), the optimal $(p, p') = (0.2, 0.8)$ under the Bayes criterion will be used to obatin the minimum asymptotic variance. Thus, $(L_{0.2}, L_{0.8})$ are estimated in the 2-root finding procedures (i.e. the RM procedure, Anbar's 2-root procedure, Wu's 2-root procedure, and the new procedure). The two roots $L_{0.5}$ and $L_{0.75}$ are estimated by

$$\hat{L}_{p*} = k\hat{L}_p + (1-k)\hat{L}_{p'}, \tag{4.7}$$

where $p = 0.2$, $p' = 0.8$, $p^* = 0.5$ or 0.75, and k is defined by (2.21).

For Anbar's and Wu's one root procedures, $L_{0.5}$ was estimated by (4.2) and (4.4), respectively. For the logit model, the p^{th} percentile is $L_p=(\log\frac{p}{1-p}-\mu)/\beta$. Thus, for Wu's one root procedure, $L_{0.75}$ is estimated by

$$\hat{L}_{0.75} = (\log\frac{0.75}{1-0.75} - \hat{\mu}_{2n})/\hat{\beta}_{2n} \tag{4.8}$$

where $(\hat{\mu}_{2n}, \hat{\beta}_{2n})$ are the MLE's of (μ, β) with 2n observations.

For Anbar's one root procedure, b_{mn} in (1.5) is used to estimate the tangent slope $\frac{\partial}{\partial x} M(L_p) = \beta p(1-p)$. Thus, for $p = 0.5$, β is estimated by

$$\hat{\beta}_{2n} = b_{m(2n)}/[0.5(1-0.5)]. \tag{4.9}$$

Also, μ is estimated by

$$\hat{\mu}_{2n} = \log\frac{0.5}{1-0.5} - \hat{\beta}_{2n}\hat{L}_{0.5}. \tag{4.10}$$

Thus, $L_{0.75}$ is estimated by

$$\hat{L}_{0.75} = (\log\frac{0.75}{1-0.75} - \hat{\mu}_{2n})/\hat{\beta}_{2n} \tag{4.11}$$

where $\hat{\mu}_{2n}$ and $\hat{\beta}_{2n}$ are defined by (4.10) and (4.9).

The MLE's of the parameters (μ,β) of a logit model are used in Wu's procedure. However, the MLE's do not always exist. Silvapulle (1981) showed that the MLE's of the parameters of any distribution function exist if and only if

$$(x^{+}_{min}, x^{+}_{max}) \cap (x^{-}_{min}, x^{-}_{max}) \text{ is nonempty,} \tag{4.12}$$

where

$$x^{+}_{min(max)} = \min(\max)\{x_i : y_i = 1\}$$

and

$$x^{-}_{min(max)} = \min(\max)\{x_i : y_i = 0\}.$$

Once (4.12) is satisfied, it is always satisfied with the addition of more observations.

Wu's procedure can not be carried out until the MLE's of the parameters exist. Thus, it is necessary to initiate Wu's procedure by some predetermined initial design procedures. Once enough observations are generated so that the MLE's exist, then the future observations can be generated from the Wu's procedure. Although Anbar's, RM's, and the new procedures do not require the existence of the MLE's, the same initial designs were used to initiate all procedures. In this way, all the procedures begin in an equivalent and comparable manner. Two different initial designs were used in the simulation study. They are discussed in the following two sections.

Initial Design I

For the first initial design, the first ten x's are chosen at two different sets of starting points, and the corresponding y's are generated according to the true model. Starting points I are chosen at ($L_{.1}$, $L_{.3}$, $L_{.5}$, $L_{.7}$, $L_{.9}$) with (1, 2, 4, 2, 1) observations each. Starting points II are chosen at ($L_{.3}$, $L_{.46}$, $L_{.56}$, $L_{.66}$, $L_{.8}$) with (1, 2, 4, 2, 1) observations each. If the MLE's of (μ, β) based on the ten pairs of (x,y) exist, then the four 2-root finding procedures are initiated at the common starting points (x_{11},x'_{11}) where both x_{11} and x'_{11} are calculated by (4.4); and the two 1-root finding procedures (Anbar's 1-root procedure and Wu's 1-root procedure) are initiated at the common starting point x''_1 which is also calculated by (4.4). If the MLE's of (μ, β) based on the initial data set do not exist, then the sample is discarded. This is repeated 500 times for each procedure including those samples discarded due to the nonexistence of MLE's.

For sample size n, the Monte Carlo mean squares error (MSE) of a sequential design is calculated as the average of $(\hat{L}_p - L_p)^2$ over all the non-discarded simulation samples.

The $\sqrt{MSE}$'s from the six different procedures are listed in Tables 4.1 – 4.8. Four sample sizes, n = 15, 30, 50, and 100, are used in each table. In these tables, Robbins-Monro's procedure is referred as "RM"; Anbar's 1-root procedure is referred as "Anb2n"; Anbar's 2-root procedure is referred as "Anb"; Wu's 1-root procedure is referred as "Wu2n"; Wu's 2-root procedure is referred as "Wu"; the new procedure is refered as "Fei". The first column of these tables represents the six procedures with different bounded values on the inverse tangent slopes of M(x). The subsequent columns are the $\sqrt{MSE}$'s of the estimators of percentiles $L_{0.2}$, $L_{0.8}$, $L_{0.5}$, and $L_{0.75}$ under the different sample sizes.

From Table 4.1 to 4.8, the $\sqrt{MSE}$'s of $\hat{L}_{0.5}$ from Anbar's and Wu's one root procedures are always less than that from Anbar's and Wu's 2-root procedures. However, the $\sqrt{MSE}$'s of $\hat{L}_{0.75}$ from Anbar's and Wu's one root procedures are greater than that from Anbar's and Wu's 2-root procedures for large n. This implies that a single root is more accurately estimated by a 1-root procedure than by a 2-root procedure. However, for estimating other roots, one root procedures perform worse than 2-root procedures.

Among the 2-root finding procedures (i.e. RM procedure, Anbar's 2-root procedure, Wu's 2-root procedure, and this new procedure), Wu's and the new procedures perform substantially better than RM and Anbar's procedures. Although RM and Anbar's procedures do not assume that the parametric form of M is known, Wu's and the new procedures do. Tables 4.1 to 4.8 show that the $\sqrt{MSE}$'s from Wu's and the new procedures are always smaller than that from RM and Anbar's procedures no matter what the true model is. Thus, for initial design I, Wu's 2-root procedure and the new procedure outperform the others.

From Tables 4.1 (the true model is logit) and Tables 4.5 (the true model is probit), for starting points I, Wu's 2-root procedure has smaller $\sqrt{MSE}$'s when n = 30 and 50. However, from Tables 4.2 and 4.6, for starting points II, the new procedure has smaller $\sqrt{MSE}$'s when n = 15, 30, and 50. Note that both procedures have similar $\sqrt{MSE}$'s as n = 100.

From Tables 4.3 and 4.4 (the true model is log-log), it can be found that the performances of Wu's 2-root procedure and the new procedure depend on the bounded values of the inverse tangent slopes of M and the percentiles to be estimated. For example, in Table 4.3, the $\sqrt{MSE}$ of $\hat{L}_{0.8}$ from the new procedure are smaller than that from Wu's 2-root procedure for all bounded values as n = 15, 30, 100. However, the $\sqrt{MSE}$ of

$\hat{L}_{0.2}$ from Wu's 2-root procedure are smaller than that from the new procedure for all bounded values as n = 30, 50, and 100. Also, when n = 15 and bounded value is 36, the $\sqrt{MSE}$ of $\hat{L}_{0.2}$ from the Wu's 2-root procedure is 1.71, which is larger than 1.66 - the $\sqrt{MSE}$ from the new procedure. However, for n = 15 and a bounded value of 100, the $\sqrt{MSE}$ of $\hat{L}_{0.2}$ from Wu's 2-root procedure is 1.43, which is smaller than 1.66 - the $\sqrt{MSE}$ from the new procedure.

From Tables 4.7 and 4.8 (the true model is skewed logit model with different starting points), Wu's procedure has smaller $\sqrt{MSE}$'s than the new procedure for n = 30, 50 and 100.

It is worthy to note that, for a given sample size, the $\sqrt{MSE}$ of an estimator from the new procedure varies for different bounded values only when the true models are log-log and skewed logit models and the bounded values are 36 and 50. This indicates that bounding the estimators of inverse tangent slopes of M does not affect the performance of the new procedure. However, for Wu's 2-root procedure, the optimal bounded values such that the $\sqrt{MSE}$ is minimized varies for the different true models. For example, for Wu's 2-root procedure, the optimal bounded value is 36 for the logit model. However, it is 100 for the log-log model, and 200 for the skewed logit model.

TABLE 4.1

MONTE CARLO $\sqrt{MSE}$ OF SEQUENTIAL DESIGN FOR INITIAL DESIGN I WITH STARTING POINTS I (BASED ON LOGIT MODEL)

Design	n = 15				n = 30			
	L20	L80	L50	L75	L20	L80	L50	L75
RM1	1.58	1.45	.97	1.28	1.39	1.51	.87	1.32
RM6	1.48	1.36	.89	1.20	1.09	1.19	.70	1.04
RM36	1.62	1.43	1.02	1.28	.92	.92	.58	.81
Anb36	1.46	1.33	.85	1.16	.96	1.04	.58	.90
Anb50	1.48	1.34	.85	1.17	.99	1.04	.58	.90
Anb100	1.52	1.41	.86	1.22	1.01	1.07	.58	.93
Anb200	1.62	1.63	.94	1.41	1.09	1.08	.60	.93
Wu36	1.29	1.20	.80	1.06	.57	.68	.43	.61
Wu50	1.21	1.20	.78	1.06	.56	.68	.43	.61
Wu100	.99	1.20	.72	1.06	.57	.68	.43	.61
Wu200	.99	1.20	.73	1.07	.57	.68	.43	.61
Fei36	1.28	1.19	.78	1.05	.70	.75	.47	.67
Fei50	1.28	1.19	.78	1.05	.70	.75	.47	.67
Fei100	1.28	1.19	.78	1.05	.70	.75	.47	.67
Fei200	1.28	1.19	.78	1.05	.70	.75	.47	.67
Wu2n36	—	—	.43	.77	—	—	.28	.72
Wu2n50	—	—	.43	.77	—	—	.28	.72
Wu2n100	—	—	.42	.77	—	—	.28	.72
Wu2n200	—	—	.42	.77	—	—	.28	.72
Anb2n36	—	—	.72	1.53	—	—	.38	.70
Anb2n50	—	—	.75	1.84	—	—	.38	.71
Anb2n100	—	—	.73	2.11	—	—	.38	.72
Anb2n200	—	—	.84	2.04	—	—	.38	.76

TABLE 4.1 (Continued)

Design	n = 50				n = 100			
	L20	L80	L50	L75	L20	L80	L50	L75
RM1	1.22	1.21	.77	1.08	1.25	.99	.68	.86
RM6	.85	.84	.55	.74	.95	1.29	.71	1.14
RM36	.66	.65	.45	.58	.46	.47	.33	.42
Anb36	.71	.67	.44	.59	.48	.38	.28	.33
Anb50	.73	.68	.44	.59	.47	.38	.28	.33
Anb100	.77	.70	.43	.61	.48	.38	.28	.33
Anb200	.82	.78	.44	.67	.48	.38	.28	.33
Wu36	.43	.39	.28	.35	.29	.29	.20	.26
Wu50	.43	.39	.28	.35	.29	.29	.21	.27
Wu100	.43	.39	.28	.35	.29	.29	.21	.26
Wu200	.43	.39	.28	.35	.29	.29	.21	.27
Fei36	.47	.46	.32	.41	.31	.29	.21	.26
Fei50	.47	.46	.32	.41	.31	.29	.21	.26
Fei100	.47	.46	.32	.41	.31	.29	.21	.26
Fei200	.47	.46	.32	.41	.31	.29	.21	.26
Wu2n36	—	—	.22	.66	—	—	.16	.64
Wu2n50	—	—	.22	.66	—	—	.16	.64
Wu2n100	—	—	.22	.66	—	—	.16	.64
Wu2n200	—	—	.22	.66	—	—	.16	.64
Anb2n36	—	—	.27	.52	—	—	.18	.41
Anb2n50	—	—	.27	.52	—	—	.18	.41
Anb2n100	—	—	.27	.52	—	—	.18	.42
Anb2n200	—	—	.27	.53	—	—	.18	.45

TABLE 4.2

MONTE CARLO $\sqrt{MSE}$ OF SEQUENTIAL DESIGN FOR INITIAL DESIGN I WITH STARTING POINTS II (BASED ON LOGIT MODEL)

Design	n = 15				n = 30			
	L20	L80	L50	L75	L20	L80	L50	L75
RM1	1.08	1.37	.75	1.20	.91	1.34	.76	1.19
RM6	.95	1.29	.71	1.14	.58	1.07	.61	.96
RM36	1.31	1.51	.98	1.36	.92	.92	.61	.82
Anb36	1.16	1.34	.69	1.16	.78	1.05	.53	.91
Anb50	1.28	1.40	.72	1.20	.86	1.09	.55	.95
Anb100	1.83	1.68	.96	1.44	1.29	1.33	.69	1.14
Anb200	2.97	2.45	1.62	2.12	2.14	1.99	1.12	1.70
Wu36	1.02	1.16	.73	1.03	.66	.72	.49	.65
Wu50	1.10	1.17	.74	1.04	.68	.72	.50	.66
Wu100	1.31	1.17	.81	1.03	.70	.73	.50	.65
Wu200	1.81	1.17	1.01	1.04	.77	.73	.53	.66
Fei36	.92	1.13	.63	.99	.56	.73	.42	.65
Fei50	.92	1.13	.63	.99	.56	.73	.42	.65
Fei100	.92	1.13	.63	.99	.56	.73	.42	.65
Fei200	.92	1.13	.63	.99	.56	.73	.42	.65
Wu2n36	—	—	.40	.73	—	—	.28	1.32
Wu2n50	—	—	.40	.69	—	—	.26	.67
Wu2n100	—	—	.41	.69	—	—	.26	.67
Wu2n200	—	—	.43	.69	—	—	.31	.80
Anb2n36	—	—	.94	1.70	—	—	.51	.85
Anb2n50	—	—	1.14	2.06	—	—	.56	.89
Anb2n100	—	—	1.68	2.34	—	—	.75	1.02
Anb2n200	—	—	3.07	3.91	—	—	1.18	1.41

TABLE 4.2 (Continued)

Design	n = 50				n = 100			
	L20	L80	L50	L75	L20	L80	L50	L75
RM1	1.00	1.08	.60	.94	.94	1.42	.73	1.25
RM6	.54	.70	.39	.62	.36	.89	.45	.80
RM36	.71	.67	.47	.60	.44	.49	.34	.45
Anb36	.61	.66	.37	.57	.43	.59	.32	.52
Anb50	.65	.68	.39	.59	.45	.60	.33	.53
Anb100	.88	.83	.52	.73	.45	.66	.37	.58
Anb200	1.50	1.28	.90	1.13	.82	.82	.52	.72
Wu36	.50	.42	.30	.37	.30	.31	.21	.28
Wu50	.51	.42	.30	.37	.30	.31	.21	.28
Wu100	.51	.42	.30	.37	.30	.31	.21	.28
Wu200	.51	.42	.30	.37	.30	.31	.21	.28
Fei36	.42	.42	.26	.37	.29	.34	.20	.30
Fei50	.42	.42	.26	.37	.29	.34	.20	.30
Fei100	.42	.42	.26	.37	.29	.34	.20	.30
Fei200	.42	.42	.26	.37	.29	.34	.20	.30
Wu2n36	—	—	.22	.69	—	—	.18	.68
Wu2n50	—	—	.22	.70	—	—	.18	.68
Wu2n100	—	—	.22	.70	—	—	.18	.67
Wu2n200	—	—	.22	.69	—	—	.18	.67
Anb2n36	—	—	.36	.75	—	—	.23	.63
Anb2n50	—	—	.38	.74	—	—	.23	.73
Anb2n100	—	—	.44	.78	—	—	.24	.66
Anb2n200	—	—	.50	1.09	—	—	.23	1.16

TABLE 4.3

MONTE CARLO $\sqrt{MSE}$ OF SEQUENTIAL DESIGN FOR INITIAL DESIGN I WITH STARTING POINTS I (BASED ON LOG-LOG MODEL)

Design	n = 15				n = 30			
	L20	L80	L50	L75	L20	L80	L50	L75
RM1	2.01	1.05	1.07	.92	1.79	1.05	.99	.92
RM6	1.90	.94	.99	.81	1.52	.70	.82	.60
RM36	1.96	1.09	1.09	.95	1.09	.71	.71	.64
Anb36	1.87	.92	.96	.78	1.31	.65	.70	.55
Anb50	1.88	.92	.96	.78	1.32	.65	.70	.55
Anb100	1.93	.92	.98	.77	1.38	.65	.72	.54
Anb200	2.09	.92	1.04	.77	1.62	.65	.81	.53
Wu36	1.71	.88	.86	.75	.81	.45	.46	.41
Wu50	1.62	.88	.83	.75	.70	.45	.40	.40
Wu100	1.43	.88	.75	.76	.70	.45	.40	.40
Wu200	1.61	.88	.84	.77	.74	.45	.42	.40
Fei36	1.66	.78	.87	.68	.98	.43	.56	.39
Fei50	1.66	.78	.87	.68	.98	.43	.56	.40
Fei100	1.66	.78	.87	.68	.98	.43	.56	.40
Fei200	1.66	.78	.87	.68	.98	.43	.56	.40
Wu2n36	—	—	.44	.66	—	—	.29	.59
Wu2n50	—	—	.45	.66	—	—	.29	.59
Wu2n100	—	—	.44	.66	—	—	.29	.59
Wu2n200	—	—	.44	.66	—	—	.29	.59
Anb2n36	—	—	.86	1.27	—	—	.53	.79
Anb2n50	—	—	.86	1.27	—	—	.53	.80
Anb2n100	—	—	.86	1.27	—	—	.53	.81
Anb2n200	—	—	.86	1.27	—	—	.54	.84

TABLE 4.3 (Continued)

Design	n = 50				n = 100			
	L20	L80	L50	L75	L20	L80	L50	L75
RM1	1.93	2.17	1.50	1.96	1.77	.89	.90	.75
RM6	1.50	1.85	1.22	1.66	1.17	.29	.65	.27
RM36	.78	1.12	.71	1.00	.52	.37	.40	.34
Anb36	1.14	1.37	.91	1.22	.70	.27	.44	.24
Anb50	1.15	1.37	.92	1.22	.70	.27	.44	.24
Anb100	1.22	1.38	.93	1.22	.72	.27	.44	.24
Anb200	1.41	1.39	.99	1.24	.77	.27	.46	.24
Wu36	.55	1.12	.64	1.00	.37	.22	.28	.21
Wu50	.53	.83	.50	.74	.36	.22	.28	.21
Wu100	.53	.77	.48	.69	.36	.22	.28	.21
Wu200	.54	.77	.48	.69	.37	.22	.28	.21
Fei36	.79	1.03	.67	.92	.45	.20	.33	.20
Fei50	.79	.90	.61	.80	.45	.20	.33	.20
Fei100	.79	.89	.61	.79	.45	.20	.33	.20
Fei200	.79	.89	.61	.79	.45	.20	.33	.20
Wu2n36	—	—	.23	.57	—	—	.16	.53
Wu2n50	—	—	.23	.57	—	—	.17	.53
Wu2n100	—	—	.23	.57	—	—	.17	.53
Wu2n200	—	—	.23	.57	—	—	.17	.53
Anb2n36	—	—	.40	.69	—	—	.23	.42
Anb2n50	—	—	.40	.68	—	—	.23	.42
Anb2n100	—	—	.40	.69	—	—	.23	.42
Anb2n200	—	—	.40	.70	—	—	.23	.42

TABLE 4.4

MONTE CARLO $\sqrt{MSE}$ OF SEQUENTIAL DESIGN FOR INITIAL DESIGN I WITH STARTING POINTS II (BASED ON LOG-LOG MODEL)

Design	n = 15				n = 30			
	L20	L80	L50	L75	L20	L80	L50	L75
RM1	1.74	1.33	.79	1.10	1.61	1.42	.89	1.22
RM6	1.62	1.27	.76	1.06	1.28	1.20	.79	1.05
RM36	1.67	1.35	.95	1.16	1.16	1.02	.82	.92
Anb36	1.72	1.24	.76	1.02	1.26	1.03	.74	.89
Anb50	1.79	1.27	.80	1.04	1.33	1.06	.76	.91
Anb100	2.34	1.39	1.07	1.14	1.78	1.21	.97	1.04
Anb200	3.46	1.62	1.64	1.35	2.82	1.63	1.52	1.42
Wu36	1.43	1.11	.73	.95	.95	.82	.63	.74
Wu50	1.37	1.08	.73	.94	.87	.73	.57	.66
Wu100	1.50	1.08	.83	.96	.86	.73	.58	.67
Wu200	1.88	1.09	1.01	.96	.89	.73	.59	.67
Fei36	1.49	1.06	.69	.89	.93	.78	.60	.70
Fei50	1.46	1.02	.69	.86	.90	.73	.58	.66
Fei100	1.45	1.01	.69	.85	.89	.73	.58	.66
Fei200	1.45	1.01	.69	.85	.89	.73	.58	.66
Wu2n36	—	—	.40	.83	—	—	.27	.85
Wu2n50	—	—	.41	.71	—	—	.28	.86
Wu2n100	—	—	.46	.71	—	—	.29	.70
Wu2n200	—	—	.54	.71	—	—	.38	.63
Anb2n36	—	—	1.04	1.41	—	—	.55	.83
Anb2n50	—	—	1.17	1.57	—	—	.61	.85
Anb2n100	—	—	1.77	2.08	—	—	.78	.95
Anb2n200	—	—	3.04	3.49	—	—	1.19	1.32

TABLE 4.4 (Continued)

Design	n = 50				n = 100			
	L20	L80	L50	L75	L20	L80	L50	L75
RM1	1.83	1.68	.99	1.44	1.25	1.19	.79	1.05
RM6	1.39	1.37	.82	1.18	.62	.95	.60	.85
RM36	.86	.74	.61	.66	.52	.41	.41	.37
Anb36	1.06	.95	.67	.82	.56	.56	.42	.49
Anb50	1.12	.96	.68	.82	.57	.57	.43	.50
Anb100	1.34	1.01	.79	.87	.65	.61	.47	.54
Anb200	1.95	1.14	1.07	.98	.82	.76	.58	.67
Wu36	.75	.65	.51	.59	.36	.31	.29	.29
Wu50	.60	.47	.40	.43	.37	.20	.28	.21
Wu100	.57	.44	.38	.41	.37	.20	.27	.21
Wu200	.56	.44	.38	.41	.36	.20	.27	.21
Fei36	.74	.65	.50	.59	.39	.38	.30	.35
Fei50	.70	.59	.47	.53	.39	.35	.29	.33
Fei100	.70	.59	.47	.53	.39	.35	.29	.33
Fei200	.70	.59	.47	.53	.39	.35	.29	.33
Wu2n36	—	—	.21	.56	—	—	.17	.57
Wu2n50	—	—	.21	.56	—	—	.17	.57
Wu2n100	—	—	.21	.56	—	—	.17	.57
Wu2n200	—	—	.21	.56	—	—	.17	.57
Anb2n36	—	—	.43	.70	—	—	.25	.55
Anb2n50	—	—	.45	.70	—	—	.26	.57
Anb2n100	—	—	.53	.74	—	—	.26	.67
Anb2n200	—	—	.62	1.12	—	—	.25	1.27

TABLE 4.5

MONTE CARLO $\sqrt{MSE}$ OF SEQUENTIAL DESIGN FOR INITIAL DESIGN I WITH STARTING POINTS I (BASED ON PROBIT MODEL)

Design	n = 15				n = 30			
	L20	L80	L50	L75	L20	L80	L50	L75
RM1	1.44	1.32	.88	1.16	1.25	1.35	.79	1.18
RM6	1.33	1.23	.81	1.08	.96	1.04	.62	.91
RM36	1.45	1.30	.92	1.15	.85	.82	.53	.72
Anb36	1.29	1.21	.77	1.06	.87	.91	.53	.79
Anb50	1.29	1.22	.77	1.07	.88	.91	.53	.79
Anb100	1.29	1.29	.77	1.13	.91	.91	.53	.79
Anb200	1.29	1.52	.82	1.33	1.00	.91	.55	.79
Wu36	1.16	1.09	.73	.96	.53	.60	.38	.54
Wu50	1.09	1.09	.70	.96	.53	.60	.39	.54
Wu100	.91	1.09	.65	.96	.53	.60	.39	.54
Wu200	1.00	1.09	.64	.95	.53	.61	.39	.54
Fei36	1.15	1.07	.71	.94	.62	.67	.42	.59
Fei50	1.15	1.07	.71	.94	.62	.67	.42	.59
Fei100	1.15	1.07	.71	.94	.62	.67	.42	.59
Fei200	1.15	1.07	.71	.94	.62	.67	.42	.59
Wu2n36	—	—	.41	.75	—	—	.28	.70
Wu2n50	—	—	.41	.75	—	—	.28	.70
Wu2n100	—	—	.41	.75	—	—	.28	.70
Wu2n200	—	—	.41	.75	—	—	.28	.70
Anb2n36	—	—	.66	1.45	—	—	.37	.70
Anb2n50	—	—	.70	1.82	—	—	.37	.70
Anb2n100	—	—	.68	1.46	—	—	.37	.71
Anb2n200	—	—	.73	1.53	—	—	.37	.72

TABLE 4.5 (Continued)

Design	n = 50				n = 100			
	L20	L80	L50	L75	L20	L80	L50	L75
RM1	1.08	1.08	.69	.95	1.16	.87	.62	.76
RM6	.73	.72	.47	.64	.63	.42	.35	.37
RM36	.63	.62	.43	.55	.45	.44	.31	.40
Anb36	.62	.59	.39	.52	.41	.32	.24	.28
Anb50	.63	.59	.38	.52	.42	.32	.24	.28
Anb100	.64	.59	.38	.51	.42	.32	.24	.28
Anb200	.70	.59	.40	.51	.42	.32	.24	.28
Wu36	.40	.36	.25	.32	.26	.27	.18	.24
Wu50	.40	.36	.25	.32	.26	.27	.18	.24
Wu100	.40	.36	.25	.32	.26	.27	.18	.24
Wu200	.40	.36	.25	.32	.27	.27	.19	.24
Fei36	.42	.41	.29	.37	.28	.26	.19	.24
Fei50	.42	.41	.29	.37	.28	.26	.19	.24
Fei100	.42	.41	.29	.37	.28	.26	.19	.24
Fei200	.42	.41	.29	.37	.28	.26	.19	.24
Wu2n36	—	—	.21	.65	—	—	.16	.63
Wu2n50	—	—	.22	.65	—	—	.16	.63
Wu2n100	—	—	.21	.65	—	—	.16	.63
Wu2n200	—	—	.21	.65	—	—	.16	.63
Anb2n36	—	—	.26	.52	—	—	.18	.42
Anb2n50	—	—	.26	.52	—	—	.18	.42
Anb2n100	—	—	.26	.52	—	—	.18	.43
Anb2n200	—	—	.26	.54	—	—	.18	.47

TABLE 4.6

MONTE CARLO $\sqrt{MSE}$ OF SEQUENTIAL DESIGN FOR INITIAL DESIGN I WITH STARTING POINTS II (BASED ON PROBIT MODEL)

Design	n = 15				n = 30			
	L20	L80	L50	L75	L20	L80	L50	L75
RM1	1.04	1.28	.71	1.12	.88	1.18	.69	1.05
RM6	.90	1.20	.67	1.06	.54	.90	.52	.81
RM36	1.23	1.41	.92	1.27	.86	.85	.58	.76
Anb36	1.03	1.26	.64	1.09	.70	.92	.47	.80
Anb50	1.13	1.32	.66	1.14	.78	.97	.49	.84
Anb100	1.57	1.59	.87	1.37	1.13	1.20	.65	1.04
Anb200	2.57	2.38	1.50	2.07	1.97	1.89	1.17	1.65
Wu36	1.01	1.09	.70	.97	.63	.64	.44	.57
Wu50	1.05	1.08	.70	.96	.63	.64	.44	.57
Wu100	1.22	1.08	.76	.96	.69	.64	.46	.58
Wu200	1.65	1.08	.94	.97	.70	.64	.47	.58
Fei36	.86	1.05	.61	.93	.52	.64	.38	.57
Fei50	.86	1.05	.61	.93	.52	.64	.38	.57
Fei100	.86	1.05	.61	.93	.52	.64	.38	.57
Fei200	.86	1.05	.61	.93	.52	.64	.38	.57
Wu2n36	—	—	.41	.67	—	—	.28	1.33
Wu2n50	—	—	.42	.67	—	—	.26	.65
Wu2n100	—	—	.44	.67	—	—	.30	.78
Wu2n200	—	—	.41	.67	—	—	.31	.79
Anb2n36	—	—	.91	1.63	—	—	.51	.84
Anb2n50	—	—	1.10	1.85	—	—	.56	.87
Anb2n100	—	—	1.65	2.24	—	—	.73	.99
Anb2n200	—	—	3.03	3.92	—	—	1.18	1.37

TABLE 4.6 (Continued)

Design	n = 50				n = 100			
	L20	L80	L50	L75	L20	L80	L50	L75
RM1	1.00	1.00	.56	.86	.95	1.39	.71	1.22
RM6	.52	.61	.34	.53	.35	.90	.45	.79
RM36	.67	.64	.44	.57	.43	.46	.32	.41
Anb36	.57	.58	.33	.50	.39	.55	.30	.48
Anb50	.60	.60	.35	.52	.40	.55	.31	.49
Anb100	.81	.73	.47	.63	.48	.59	.34	.51
Anb200	1.38	1.09	.80	.96	.81	.73	.50	.64
Wu36	.45	.38	.27	.34	.27	.29	.19	.26
Wu50	.45	.38	.26	.34	.28	.29	.19	.26
Wu100	.45	.38	.27	.34	.27	.29	.19	.26
Wu200	.45	.38	.26	.34	.27	.29	.19	.26
Fei36	.39	.37	.24	.33	.27	.33	.20	.29
Fei50	.39	.37	.24	.33	.27	.33	.20	.29
Fei100	.39	.37	.24	.33	.27	.33	.20	.29
Fei200	.39	.37	.24	.33	.27	.33	.20	.29
Wu2n36	—	—	.22	.67	—	—	.17	.66
Wu2n50	—	—	.22	.68	—	—	.18	.66
Wu2n100	—	—	.22	.68	—	—	.17	.66
Wu2n200	—	—	.22	.68	—	—	.17	.65
Anb2n36	—	—	.36	.73	—	—	.23	.62
Anb2n50	—	—	.38	.73	—	—	.24	.62
Anb2n100	—	—	.44	.75	—	—	.24	.66
Anb2n200	—	—	.50	1.08	—	—	.23	1.16

TABLE 4.7

MONTE CARLO $\sqrt{MSE}$ OF SEQUENTIAL DESIGN FOR INITIAL DESIGN I WITH STARTING POINTS I (BASED ON SKEWED LOGIT MODEL)

Design	n = 15				n = 30			
	L20	L80	L50	L75	L20	L80	L50	L75
RM1	2.13	3.51	1.78	3.09	1.61	2.82	1.52	2.51
RM6	2.02	3.39	1.66	2.98	1.28	2.51	1.24	2.22
RM36	1.88	3.01	1.45	2.64	.92	1.65	.93	1.49
Anb36	1.90	3.15	1.61	2.78	1.04	1.98	1.07	1.77
Anb50	1.90	3.15	1.61	2.78	1.04	1.98	1.07	1.77
Anb100	1.90	3.15	1.61	2.78	1.04	1.99	1.07	1.78
Anb200	1.90	3.15	1.61	2.78	1.04	2.01	1.08	1.80
Wu36	1.68	2.88	1.42	2.54	.58	1.66	.90	1.49
Wu50	1.54	2.72	1.39	2.40	.51	1.38	.75	1.22
Wu100	1.16	2.39	1.30	2.14	.47	1.26	.71	1.14
Wu200	.91	2.37	1.32	2.14	.48	1.25	.70	1.13
Fei36	1.62	2.96	1.37	2.59	.65	1.61	.86	1.45
Fei50	1.53	2.85	1.36	2.51	.63	1.50	.82	1.35
Fei100	1.51	2.84	1.36	2.50	.63	1.50	.82	1.35
Fei200	1.51	2.84	1.36	2.50	.63	1.50	.82	1.35
Wu2n36	—	—	.53	.87	—	—	.27	.75
Wu2n50	—	—	.47	.81	—	—	.27	.75
Wu2n100	—	—	.46	.80	—	—	.27	.75
Wu2n200	—	—	.45	.79	—	—	.27	.74
Anb2n36	—	—	.83	2.08	—	—	.33	.75
Anb2n50	—	—	.86	2.25	—	—	.33	.75
Anb2n100	—	—	.91	2.59	—	—	.33	.76
Anb2n200	—	—	1.06	3.82	—	—	.33	.77

TABLE 4.7 (Continued)

Design	n = 50				n = 100			
	L20	L80	L50	L75	L20	L80	L50	L75
RM1	1.70	2.81	1.55	2.51	2.04	3.01	1.45	2.63
RM6	1.27	2.39	1.19	2.12	1.44	2.33	1.03	2.02
RM36	.64	1.21	.70	1.09	.41	.62	.37	.56
Anb36	.86	1.63	.90	1.46	.55	1.09	.62	.99
Anb50	.87	1.63	.90	1.46	.55	1.09	.62	.99
Anb100	.87	1.64	.90	1.47	.55	1.09	.62	.99
Anb200	.87	1.67	.90	1.49	.55	1.10	.62	.99
Wu36	.42	1.07	.61	.97	.25	.51	.29	.45
Wu50	.39	.90	.42	.63	.25	.36	.22	.32
Wu100	.38	.65	.40	.59	.25	.36	.22	.32
Wu200	.38	.65	.40	.58	.25	.36	.23	.32
Fei36	.52	1.21	.67	1.09	.35	.74	.42	.67
Fei50	.50	1.13	.62	1.02	.33	.68	.39	.62
Fei100	.49	1.13	.62	1.02	.33	.68	.39	.62
Fei200	.49	1.13	.62	1.02	.33	.68	.39	.62
Wu2n36	—	—	.22	.76	—	—	.16	.65
Wu2n50	—	—	.22	.76	—	—	.16	.65
Wu2n100	—	—	.22	.76	—	—	.16	.65
Wu2n200	—	—	.22	.76	—	—	.16	.65
Anb2n36	—	—	.26	.68	—	—	.17	.52
Anb2n50	—	—	.26	.68	—	—	.17	.53
Anb2n100	—	—	.26	.70	—	—	.17	.53
Anb2n200	—	—	.27	.76	—	—	.17	.58

TABLE 4.8

MONTE CARLO $\sqrt{MSE}$ OF SEQUENTIAL DESIGN FOR INITIAL DESIGN I WITH STARTING POINTS II (BASED ON SKEWED LOGIT MODEL)

Design	n = 15				n = 30			
	L20	L80	L50	L75	L20	L80	L50	L75
RM1	3.25	1.70	1.76	1.53	2.20	1.56	1.36	1.42
RM6	3.14	1.60	1.67	1.44	1.95	1.26	1.16	1.15
RM36	2.88	1.69	1.54	1.50	1.38	.99	.85	.91
Anb36	2.99	1.60	1.54	1.40	1.63	1.17	.95	1.05
Anb50	3.01	1.67	1.52	1.45	1.66	1.23	.97	1.10
Anb100	3.15	2.01	1.49	1.71	1.80	1.44	1.01	1.27
Anb200	3.62	2.89	1.87	2.44	2.16	2.07	1.28	1.81
Wu36	2.80	1.31	1.45	1.16	1.24	.71	.72	.64
Wu50	2.66	1.27	1.36	1.13	1.00	.70	.60	.62
Wu100	2.20	1.20	1.17	1.06	.57	.69	.43	.61
Wu200	1.63	1.20	.95	1.06	.56	.69	.42	.61
Fei36	2.78	1.39	1.40	1.21	1.32	.84	.77	.75
Fei50	2.69	1.38	1.37	1.21	1.26	.83	.74	.75
Fei100	2.68	1.38	1.37	1.21	1.26	.83	.74	.75
Fei200	2.68	1.38	1.37	1.21	1.26	.83	.74	.75
Wu2n36	—	—	.42	.76	—	—	.25	.72
Wu2n50	—	—	.41	.75	—	—	.25	.72
Wu2n100	—	—	.40	.75	—	—	.25	.72
Wu2n200	—	—	.40	.75	—	—	.25	.72
Anb2n36	—	—	1.50	2.24	—	—	.65	.98
Anb2n50	—	—	1.61	2.65	—	—	.68	1.01
Anb2n100	—	—	2.00	3.43	—	—	.85	1.12
Anb2n200	—	—	3.05	5.29	—	—	1.29	1.50

TABLE 4.8 (Continued)

Design	n = 50				n = 100			
	L20	L80	L50	L75	L20	L80	L50	L75
RM1	2.48	1.50	1.35	1.33	2.47	1.67	1.36	1.47
RM6	2.07	1.05	1.08	.93	1.99	1.01	1.07	.92
RM36	.98	.69	.60	.63	.54	.49	.36	.45
Anb36	1.51	.82	.80	.73	1.08	.59	.60	.54
Anb50	1.53	.84	.80	.75	1.09	.60	.60	.55
Anb100	1.62	.93	.83	.82	1.13	.66	.63	.60
Anb200	1.97	1.26	1.00	1.10	1.33	.80	.72	.72
Wu36	.87	.50	.48	.44	.43	.37	.25	.31
Wu50	.54	.50	.35	.44	.26	.37	.19	.32
Wu100	.41	.50	.30	.43	.26	.37	.20	.32
Wu200	.41	.50	.30	.43	.26	.37	.19	.31
Fei36	1.02	.56	.57	.51	.69	.39	.40	.35
Fei50	.96	.56	.55	.51	.62	.39	.37	.35
Fei100	.96	.56	.55	.51	.62	.39	.37	.35
Fei200	.96	.56	.55	.51	.62	.39	.37	.35
Wu2n36	—	—	.23	.73	—	—	.17	2.25
Wu2n50	—	—	.23	.73	—	—	.17	3.09
Wu2n100	—	—	.23	.73	—	—	.17	.73
Wu2n200	—	—	.23	.73	—	—	.17	.73
Anb2n36	—	—	.42	.77	—	—	.21	.71
Anb2n50	—	—	.43	.76	—	—	.22	.69
Anb2n100	—	—	.48	.78	—	—	.22	.74
Anb2n200	—	—	.53	1.03	—	—	.21	1.09

Initial Design II

In the second initial design, the 2-parameter logit and log-log models are used as the true models. The assumed model is again the 2-parameter logit model. As in the initial design I, the roots $L_{0.2}$ and $L_{0.8}$ are estimated first in the 2-root finding procedures. The root $L_{0.5}$ is estimated first in the one root finding procedures. However, the x's in the initial data set are no longer fixed.

For the 2-root finding procedures, two independent RM procedures (4.1), one with p = 0.2 and the other with p = 0.8, generate five initial observations each. Three pairs of starting points, $(L_{.3}, L_{.9})$, $(L_{.3}, L_{.4})$, and $(L_{.45}, L_{.55})$, and three different values of A - 1, 6, and 36, are used to generate the initial data sets. Then, Silvapulle's condition (4.12) is checked. If MLE's of (μ, β) based on the logit model do not exist for both of the two initial data sets, an additional pair of observations is independently generated by the RM procedure. This process is continued until the MLE's exist or the number of observations is greater than or equal to the sample size. If the MLE's exist, then the subsequent (x_i, x_i') are generated by the corresponding procedures, (4.1), (4.3), (4.5), and (4.6). The roots $L_{0.5}$ and $L_{0.75}$ are estimated by (4.7). If the MLE's do not exist, the sample is discarded.

For the Anbar's and Wu's one root procedures, the first 10 observations are generated by the RM procedure (4.1) with p = 0.5. Three starting points, $L_{.5}$, $L_{.7}$, and $L_{.9}$, and three levels of A - 1, 6, and 36, are used to generate the initial data sets. If the MLE's of (μ, β) based on the logit model do not exist, then an additional observation is generated by RM procedure. This process is continued until the MLE's exist or the number of observations is greater or equal to the predetermined sample size. If the MLE's exist, then the subsequent x_i are generated by Anbar's (4.2) and Wu's (4.4) one root procedures. If the MLE's do not exist, then the sample is discarded.

These processes are repeated 500 times for each procedure including those samples discarded due to the nonexistence of MLE's. For Anbar's, Wu's, and the new procedures, the bounded value for the estimators of inverse tangent slopes of M is (0.005, 200). For all the six procedures, the MSE of $\hat{L}_p$ is calculated as the average of $(\hat{L}_p-L_p)^2$ over all non-discarded samples.

Tables 4.9 to 4.11 (the true model is logit) shows that the $\sqrt{MSE}$'s from Anbar's and Wu's 2-root procedures depend on the value of A. For A=1, Anbar's 2-root procedure has the largest $\sqrt{MSE}$'s among all 2-root finding procedures. However, for A=36, Wu's 2-root procedure has the largest $\sqrt{MSE}$'s among the 2-root finding procedures. Also, the new procedure has smallest $\sqrt{MSE}$'s among all 2-root finding procedures except when A = 36. For A = 36, the new procedure still has the second smallest $\sqrt{MSE}$ for n = 15, 30, and 50. Similar results are also found in Table 4.13 to Table 4.15 (the true model is log-log model).

Table 4.12 and Table 4.16 (Anbar's and Wu's one root procedures) shown that, in estimating $L_{0.5}$, Wu's one root procedure has smaller $\sqrt{MSE}$'s than Wu's 2-root procedure. However, in estimating $L_{0.75}$, Wu's one root procedure has larger $\sqrt{MSE}$'s than Wu's 2-root procedure except when A = 36 and n = 15. Also, in estimating $L_{0.5}$, Anbar's one root procedure has smaller $\sqrt{MSE}$'s than Anbar's 2-root procedure except when n = 15. However, in estimating $L_{0.75}$, Anbar's 2-root procedure has smaller $\sqrt{MSE}$'s than Anbar's one root procedure except when A = 1.

Time Consumption

In practical applications, simplicity and fast response are important criteria for a good stochastic approximation procedure. On an IBM 10 MHz AT compatible computer with math co-processor, the time consumption of these six procedures for initial design I with 500 samples are listed in Table 4.17. Since Wu's procedure requires using the Newton-Raphson method repeatedly for each additional observation and the Newton-Raphson method is a time consuming procedure, the time consumption for Wu's procedure is significantly greater than that for the other procedures. The differences of time consumption between Wu's procedure and other procedures increases quickly as n is increased.

General Conclusions

In the simulation comparisons, it is difficult to compare Wu's procedure with the other procedures. The existence of MLE's is required for Wu's procedure. However, this is not required for the others procedures. In both initial designs, all procedures will start their sequential designs independently after the MLE's of the parameters exist. This means that all procedures will start under conditions which favor Wu's procedure.

By Wu's paper (1985) and this research, it is shown that Wu's procedure performs well when some prior information about the function M is known or the sample size is large. For example, in initial design I, Wu's procedure performs well when the locations of the first ten x's is such that the probability of the sample to be discarded is small; or in design II, the $\sqrt{MSE}$'s from Wu's procedure with bounded value 36 are small only when n = 100.

If the objective is to find a non-extreme root only, the Wu's procedure performs well. However, if the objective is to estimate the whole function M, the new procedure has the benefits of accuracy, simplicity, and ease of calculation.

TABLE 4.9

MONTE CARLO $\sqrt{MSE}$ OF SEQUENTIAL DESIGN FOR INITIAL DESIGN II WITH RM STARTING POINT (L_{30}, L_{90}) (BASED ON LOGIT MODEL)

Bounded values	Design	n = 15				n = 30			
		L20	L80	L50	L75	L20	L80	L50	L75
1	RM	.47	.60	.46	.56	.46	.51	.41	.48
	Anbar	.90	.93	.69	.85	2.00	.90	1.06	.82
	Wu	.50	.65	.49	.61	.54	.57	.43	.52
	Fei	.49	.59	.45	.55	.43	.43	.33	.39
6	RM	.90	.85	.59	.76	.70	.75	.48	.67
	Anbar	1.11	.87	.66	.77	1.10	.81	.63	.71
	Wu	1.31	1.10	.78	.97	.72	.88	.54	.79
	Fei	.91	.85	.59	.75	.63	.73	.45	.65
36	RM	1.20	1.12	.84	1.01	1.11	.91	.68	.81
	Anbar	1.18	1.07	.82	.98	1.03	.80	.62	.71
	Wu	6.72	4.70	3.57	4.09	4.30	3.06	2.32	2.67
	Fei	1.19	1.06	.82	.97	1.05	.85	.66	.76

TABLE 4.9 (Continued)

A	Design	n = 50				n = 100			
		L20	L80	L50	L75	L20	L80	L50	L75
1	RM	.41	.45	.37	.43	.35	.40	.32	.38
	Anbar	1.12	.95	.66	.83	.89	.53	.53	.49
	Wu	.43	.48	.36	.44	.37	.36	.28	.33
	Fei	.34	.31	.24	.28	.25	.23	.18	.21
6	RM	.53	.64	.39	.57	.34	.42	.24	.37
	Anbar	.73	.66	.48	.59	.38	.40	.27	.36
	Wu	.53	.68	.42	.61	.35	.38	.25	.34
	Fei	.46	.61	.36	.54	.29	.33	.20	.29
36	RM	.90	.80	.57	.71	.48	.46	.33	.41
	Anbar	.86	.75	.54	.66	.42	.41	.29	.37
	Wu	1.89	1.68	1.12	1.48	.47	.45	.33	.41
	Fei	.87	.77	.55	.68	.55	.51	.38	.46

TABLE 4.10

MONTE CARLO $\sqrt{MSE}$ OF SEQUENTIAL DESIGN FOR INITIAL DESIGN II WITH RM STARTING POINT (L_{30}, L_{40}) (BASED ON LOGIT MODEL)

Bounded values	Design	n = 15				n = 30			
		L20	L80	L50	L75	L20	L80	L50	L75
1	RM	.47	.95	.40	.82	.44	.82	.35	.71
	Anbar	1.74	3.44	1.96	3.10	2.37	3.35	1.94	2.98
	Wu	.53	1.31	.65	1.17	.59	.69	.41	.61
	Fei	.47	.89	.39	.78	.41	.64	.29	.56
6	RM	.89	.78	.50	.68	.69	.60	.40	.53
	Anbar	.99	.96	.61	.84	1.00	.86	.62	.76
	Wu	1.23	.96	.67	.84	.75	.61	.43	.54
	Fei	.90	.77	.50	.67	.65	.57	.38	.50
36	RM	1.02	1.08	.65	.95	.95	1.04	.67	.93
	Anbar	.96	1.03	.63	.90	.90	1.00	.64	.90
	Wu	7.05	8.80	4.03	7.54	3.95	4.67	2.24	4.00
	Fei	.95	1.03	.63	.91	.91	1.02	.65	.91

TABLE 4.10 (Continued)

A	Design	n = 50				n = 100			
		L20	L80	L50	L75	L20	L80	L50	L75
1	RM	.41	.75	.31	.65	.36	.64	.26	.55
	Anbar	1.61	2.51	1.37	2.22	1.05	1.50	.85	1.33
	Wu	.47	.47	.29	.42	.40	.34	.23	.30
	Fei	.35	.50	.23	.43	.26	.34	.17	.30
6	RM	.54	.45	.30	.40	.32	.33	.21	.29
	Anbar	.87	.52	.46	.46	.43	.38	.27	.34
	Wu	.54	.47	.34	.42	.33	.34	.23	.31
	Fei	.47	.39	.28	.35	.27	.27	.19	.24
36	RM	.88	.92	.56	.81	.47	.44	.31	.39
	Anbar	.85	.89	.53	.78	.54	.55	.38	.50
	Wu	1.93	1.68	.92	1.42	.43	.43	.30	.38
	Fei	.86	.90	.54	.79	.58	.57	.36	.51

TABLE 4.11

MONTE CARLO $\sqrt{MSE}$ OF SEQUENTIAL DESIGN FOR INITIAL DESIGN II WITH RM STARTING POINT (L_{45}, L_{55}) (BASED ON LOGIT MODEL)

Bounded values	Design	n = 15				n = 30			
		L20	L80	L50	L75	L20	L80	L50	L75
1	RM	.71	.71	.29	.59	.64	.62	.24	.51
	Anbar	3.21	2.80	2.20	2.55	2.99	3.09	2.01	2.75
	Wu	.99	1.11	.71	.99	.63	.62	.36	.53
	Fei	.68	.69	.29	.57	.52	.53	.23	.45
6	RM	.86	.86	.48	.74	.62	.63	.36	.55
	Anbar	1.21	1.10	.75	.97	.65	1.02	.54	.90
	Wu	1.10	1.08	.63	.93	.61	.64	.39	.56
	Fei	.86	.86	.48	.74	.57	.58	.35	.51
36	RM	1.18	1.08	071	.95	.99	.95	.59	.83
	Anbar	1.09	1.01	.67	.89	.94	.89	.55	.78
	Wu	8.30	7.78	3.93	6.56	4.50	4.22	2.21	3.58
	Fei	1.10	1.02	.68	.90	.95	.93	.57	.81

TABLE 4.11 (Continued)

A	Design	n = 50				n = 100			
		L20	L80	L50	L75	L20	L80	L50	L75
1	RM	.55	.61	.23	.51	.50	.51	.21	.43
	Anbar	2.31	2.49	1.52	2.20	1.28	1.12	.79	1.00
	Wu	.49	.52	.33	.46	.32	.33	.22	.29
	Fei	.42	.45	.20	.38	.29	.30	.16	.26
6	RM	.48	.46	.28	.40	.34	.33	.20	.29
	Anbar	.80	.81	.49	.71	.42	.42	.28	.37
	Wu	.54	.50	.34	.44	.35	.35	.23	.31
	Fei	.42	.42	.27	.37	.29	.27	.19	.24
36	RM	1.00	.95	.60	.86	.49	.46	.32	.41
	Anbar	.96	.92	.58	.81	.52	.56	.38	.51
	Wu	1.97	1.88	.99	1.60	.43	.50	.31	.45
	Fei	.97	.93	.58	.81	.59	.60	.35	.52

TABLE 4.12

MONTE CARLO $\sqrt{MSE}$ OF SEQUENTIAL DESIGN FOR INITIAL DESIGN II WITH 2N OBSERVATIONS TO ESTIMATE L_{50} AND L_{75} (BASED ON LOGIT MODEL)

Starting points	Bounded values	Design	n = 15		n = 30		n = 50		n = 100	
			L50	L75	L50	L75	L50	L75	L50	L75
0.5	1	Wu	.42	1.05	.35	1.02	.33	1.01	.24	.91
		Anbar	.85	1.22	.49	.99	.41	.95	.29	2.74
	6	Wu	.57	1.00	.38	.84	.28	.75	.19	.71
		Anbar	.68	.86	.37	.54	.25	.46	.16	.38
	36	Wu	.62	.87	.37	.66	.25	.59	.18	.55
		Anbar	.79	4.44	.46	4.01	.35	3.86	.22	3.64
0.7	1	Wu	.69	.74	.43	.70	.32	.73	.21	.86
		Anbar	1.64	3.55	.82	2.50	.55	2.60	.29	.74
	6	Wu	.56	.87	.36	.74	.27	.69	.19	.63
		Anbar	.64	.84	.36	.51	.23	.53	.16	.42
	36	Wu	.68	1.01	.37	.73	.27	.67	.18	.59
		Anbar	.81	4.35	.48	4.04	.35	3.80	.22	3.55
0.9	1	Wu	1.00	.58	.49	.43	.44	3.93	.19	.42
		Anbar	2.66	6.90	1.19	.92	.56	.51	.23	.59
	6	Wu	.60	.94	.39	.82	.29	.76	.19	.67
		Anbar	.57	.65	.35	.50	.24	.48	.16	.41
	36	Wu	.59	1.01	.37	.84	.29	.74	.20	.67
		Anbar	.73	3.76	.44	3.49	.32	3.36	.22	3.05

TABLE 4.13

MONTE CARLO $\sqrt{MSE}$ OF SEQUENTIAL DESIGN FOR INITIAL DESIGN II WITH RM STARTING POINT (L_{30},L_{90}) (BASED ON LOG-LOG MODEL)

Bounded values	Design	n = 15				n = 30			
		L20	L80	L50	L75	L20	L80	L50	L75
1	RM	.55	.37	.26	.32	.50	.39	.26	.33
	Anbar	1.67	.79	.89	.70	1.68	.79	.92	.72
	Wu	.57	.39	.28	.33	.52	.44	.28	.38
	Fei	.55	.41	.28	.35	.49	.43	.29	.37
6	RM	.95	.56	.60	.49	.68	.53	.47	.45
	Anbar	1.32	1.05	.87	.93	.88	.90	.64	.79
	Wu	1.21	.92	.74	.80	.73	.75	.51	.65
	Fei	.95	.57	.60	.50	.65	.53	.43	.45
36	RM	1.35	.94	.96	.88	1.11	.80	.72	.70
	Anbar	1.30	.85	.92	.81	1.02	.58	.66	.52
	Wu	6.72	4.53	3.60	3.92	3.90	3.15	2.20	2.72
	Fei	1.30	.84	.92	.80	1.03	.63	.70	.57

TABLE 4.13 (Continued)

A	Design	n = 50				n = 100			
		L20	L80	L50	L75	L20	L80	L50	L75
1	RM	.49	.32	.23	.27	.44	.30	.19	.24
	Anbar	1.40	.57	.80	.53	.80	.41	.45	.35
	Wu	.50	.33	.26	.29	.43	.28	.21	.23
	Fei	.42	.34	.23	.29	.38	.29	.17	.24
6	RM	.53	.50	.35	.40	.36	.39	.22	.30
	Anbar	.66	.74	.46	.63	.40	.51	.30	.43
	Wu	.56	.56	.37	.48	.38	.37	.24	.30
	Fei	.48	.49	.32	.41	.33	.36	.20	.28
36	RM	1.04	.68	.65	.59	.47	.50	.32	.42
	Anbar	.99	.52	.61	.45	.47	.49	.30	.41
	Wu	1.90	1.56	1.11	1.34	.47	.32	.32	.29
	Fei	1.02	.53	.64	.47	.56	.34	.37	.29

TABLE 4.14

MONTE CARLO $\sqrt{MSE}$ OF SEQUENTIAL DESIGN FOR INITIAL
DESIGN II WITH RM STARTING POINT (L_{30},L_{40})
(BASED ON LOG-LOG MODEL)

Bounded values	Design	n = 15				n = 30			
		L20	L80	L50	L75	L20	L80	L50	L75
1	RM	.57	.76	.43	.71	.54	.63	.37	.60
	Anbar	1.93	3.37	1.96	3.03	2.33	3.02	1.93	2.71
	Wu	.62	1.45	.75	1.31	.60	.59	.38	.55
	Fei	.58	.73	.41	.68	.49	.48	.29	.45
6	RM	.90	.50	.52	.41	.74	.44	.46	.36
	Anbar	1.29	.81	.76	.69	1.18	.57	.68	.49
	Wu	1.11	.68	.57	.54	.75	.46	.48	.40
	Fei	.91	.51	.52	.41	.70	.44	.44	.37
36	RM	1.11	.76	.67	.62	1.07	.62	.63	.52
	Anbar	1.04	.69	.64	.59	1.02	.57	.60	.48
	Wu	6.70	9.11	4.09	7.80	4.01	4.83	2.20	4.08
	Fei	1.04	.69	.64	.59	1.03	.59	.60	.50

TABLE 4.14 (Continued)

A	Design	n = 50				n = 100			
		L20	L80	L50	L75	L20	L80	L50	L75
1	RM	.50	.57	.35	.54	.42	.48	.32	.47
	Anbar	1.71	2.59	1.47	2.29	.79	1.56	.77	1.36
	Wu	.58	.48	.34	.43	.41	.32	.23	.28
	Fei	.44	.39	.24	.36	.33	.27	.17	.24
6	RM	.56	.40	.36	.32	.34	.33	.23	.25
	Anbar	.63	.67	.46	.58	.41	.42	.27	.34
	Wu	.55	.39	.36	.33	.37	.34	.24	.28
	Fei	.48	.41	.33	.34	.30	.32	.20	.25
36	RM	.99	.59	.59	.49	.44	.48	.32	.40
	Anbar	.95	.54	.57	.45	.53	.61	.38	.52
	Wu	1.83	1.72	.93	1.42	.48	.35	.32	.30
	Fei	.97	.56	.58	.47	.62	.41	.40	.33

TABLE 4.15

MONTE CARLO $\sqrt{MSE}$ OF SEQUENTIAL DESIGN FOR INITIAL DESIGN II WITH RM STARTING POINT (L_{45}, L_{55}) (BASED ON LOG-LOG MODEL)

Bounded values	Design	n = 15				n = 30			
		L20	L80	L50	L75	L20	L80	L50	L75
1	RM	.86	.55	.29	.49	.81	.44	.26	.40
	Anbar	3.09	2.72	2.02	2.44	2.98	3.32	2.07	2.92
	Wu	1.28	1.01	.79	.93	.76	.57	.41	.52
	Fei	.83	.54	.30	.48	.67	.39	.25	.35
6	RM	.84	.61	.50	.50	.67	.49	.40	.39
	Anbar	.88	.95	.62	.81	.87	.67	.55	.56
	Wu	1.07	1.11	.65	.93	.72	.51	.42	.41
	Fei	.84	.62	.49	.51	.63	.48	.38	.38
36	RM	1.12	.77	.72	.67	.96	.65	.61	.55
	Anbar	1.05	.67	.68	.59	.94	.58	.61	.50
	Wu	7.68	8.70	3.79	7.31	4.20	4.63	2.29	3.91
	Fei	1.04	.68	.67	.59	.95	.58	.61	.50

TABLE 4.15 (Continued)

A	Design	n = 50				n = 100			
		L20	L80	L50	L75	L20	L80	L50	L75
1	RM	.72	.42	.23	.39	.67	.36	.21	.34
	Anbar	2.31	2.32	1.43	2.00	1.16	1.45	.84	1.25
	Wu	.61	.45	.32	.40	.43	.29	.22	.26
	Fei	.56	.36	.22	.32	.43	.26	.16	.22
6	RM	.49	.43	.31	.34	.33	.35	.21	.26
	Anbar	.70	.66	.44	.55	.39	.42	.28	.35
	Wu	.55	.44	.36	.37	.39	.33	.24	.27
	Fei	.45	.42	.29	.34	.30	.32	.20	.25
36	RM	.96	.56	.59	.47	.44	.51	.32	.43
	Anbar	.93	.51	.57	.42	.55	.58	.37	.50
	Wu	1.81	2.05	.95	1.70	.44	.34	.31	.30
	Fei	.95	.51	.57	.42	.60	.40	.40	.33

TABLE 4.16

MONTE CARLO $\sqrt{MSE}$ OF SEQUENTIAL DESGN FOR INITIAL DESIGN II
WITH 2N OBSERVATIONS TO ESTIMATE L50 AND L75
(BASED ON LOG-LOG MODEL)

Starting points	A	Design	n = 15 L50	n = 15 L75	n = 30 L50	n = 30 L75	n = 50 L50	n = 50 L75	n = 100 L50	n = 100 L75
0.5	1	Wu	.40	.94	.35	.91	.33	.88	.24	.83
		Anbar	.58	2.51	.50	.87	.37	.97	.30	.72
	6	Wu	.52	.91	.37	.76	.27	.67	.19	.63
		Anbar	.74	.97	.36	.63	.24	.42	.16	.40
	36	Wu	.63	.83	.35	.59	.25	.52	.17	.46
		Anbar	.83	4.57	.50	4.22	.34	4.02	.23	3.78
0.7	1	Wu	.66	.69	.41	.67	.30	.65	.22	.72
		Anbar	1.53	2.23	.79	2.63	.52	2.62	.29	.65
	6	Wu	.57	.84	.36	.68	.29	.64	.19	.53
		Anbar	.56	.68	.36	.61	.28	.50	.16	.46
	36	Wu	.61	.95	.38	.72	.29	.62	.19	.53
		Anbar	.80	4.41	.47	4.16	.34	3.91	.24	3.65
0.9	1	Wu	.89	.91	.45	.44	.34	.45	.24	.42
		Anbar	2.26	5.27	1.12	1.98	.55	.55	.30	2.61
	6	Wu	.59	.87	.38	.73	.28	.64	.19	.55
		Anbar	.65	.78	.33	.52	.25	.51	.16	.43
	36	Wu	.61	.96	.39	.83	.29	.74	.19	.65
		Anbar	.77	4.04	.44	3.78	.32	3.55	.24	3.36

TABLE 4.17

TIME CONSUMING OF SEQUENTIAL DESIGNS FOR INITIAL DESIGN I
(unit: second)

Model	Starting point	Design	n=15	n=30	n=50	n=100
Logit	I	RM	8	26	55	118
		Anbar	17	72	148	320
		Wu	170	1157	3507	13570
		Fei	9	40	82	176
		Anbar2n	17	72	147	318
		Wu2n	184	1603	5470	23402
	II	RM	5	24	48	112
		Anbar	15	64	134	306
		Wu	151	1027	3151	12941
		Fei	9	35	74	167
		Anbar2n	15	63	133	302
		Wu2n	167	1429	4936	22309
Log-log	I	RM	10	43	90	202
		Anbar	22	97	202	450
		Wu	184	1255	3821	15116
		Fei	15	62	130	291
		Anbar2n	22	96	200	444
		Wu2n	201	1725	5909	25918
	II	RM	9	36	78	167
		Anbar	19	79	174	372
		Wu	155	1038	3267	12313
		Fei	12	51	112	240
		Anbar2n	18	80	172	368
		Wu2n	172	1434	5086	21473

TABLE 4.17 (Continued)

Model	Starting point	Design	n=15	n=30	n=50	n=100
Probit	I	RM	11	51	107	232
		Anbar	23	106	218	473
		Wu	177	1193	3578	13761
		Fei	18	74	152	328
		Anbar2n	24	105	219	473
		Wu2n	193	1628	5541	23616
	II	RM	11	47	97	221
		Anbar	22	94	197	452
		Wu	157	1059	3233	13124
		Fei	15	65	137	314
		Anbar2n	22	94	199	452
		Wu2n	175	1459	5014	22516
Skewed Logit	I	RM	7	32	63	144
		Anbar	18	79	160	367
		Wu	167	1185	3538	14557
		Fei	11	46	93	213
		Anbar2n	17	78	159	364
		Wu2n	183	1648	5509	24972
	II	RM	6	26	55	117
		Anbar	15	67	138	296
		Wu	144	1029	3083	11798
		Fei	9	39	81	171
		Anbar2n	15	67	138	292
		Wu2n	160	1405	4781	22080

CHAPTER V

GENERAL FORM AND SUMMARY

In this chapter, a general form of this new procedure for an increasing function with r parameters is given. Conclusions about the new procedure are also made.

General Form

All theorems in chapter II have been proved under the 2-parameter case. In this section, the three parameter case will be given first. Then, the general form r parameter case will be proposed.

Let $M(x) = F(x;\theta_1,\theta_2,\theta_3)$ be an increasing function where θ_1, θ_2, θ_3 are the unknown parameters of M. In order to estimate the whole curve, the roots L_{p_1}, L_{p_2}, L_{p_3} are chosen to satisfy $M(L_{p_i})=p_i$ and $\frac{\partial}{\partial x} M(L_{p_i})=\alpha_i$. In the sequential procedure, a random vector $(x_{1(n)}, x_{2(n)}, x_{3(n)})$ at stage n is used as the estimator of $(L_{p_1}, L_{p_2}, L_{p_3})$. Similar to the 2-parameter case, α_i can be presented as

$$\alpha_i = c_{ij} \frac{p_j - p_i}{L_{p_j} - L_{p_i}}, \quad i=1,2,3 \text{ and } j\neq i \tag{5.1}$$

where c_{ij} depends on the true model and is a function of p_1, p_2, and p_3. A natural estimator of α_i^{-1} is $(x_{j(n)}-x_{i(n)})/[c_{ij}(p_j-p_i)]$, where i=1,2,3 and j≠i. By Figure 5.1, two estimators of α_1^{-1} can be found. Let

$$\hat{\alpha}_{1(n)}^{-1} = \frac{1}{2}\sum_{j=2}^{3}(x_{j(n)}-x_{1(n)})/[c_{1j}(p_j-p_1)] \tag{5.2}$$

$$= \sum_{j=1}^{3} d_{1j}x_j \ .$$

That is, use the average of all possible estimators as the estimator of $\alpha_{1(n)}^{-1}$. Let δ_1, δ_2 be two constants such that $0 < \delta_1 < \delta_2 < \infty$. Define

$$a_{i(n)} = \begin{cases} \delta_1^{-1} & \text{if } \hat{\alpha}_{i(n)} \le \delta_1 \\ \hat{\alpha}_{i(n)}^{-1} & \text{if } \delta_1 < \hat{\alpha}_{i(n)} < \delta_2 \\ \delta_2^{-1} & \text{if } \hat{\alpha}_{i(n)} \ge \delta_2 \end{cases} \tag{5.3}$$

and the sequential procedure is defined by

$$\begin{pmatrix} X_{1(n+1)} \\ X_{2(n+1)} \\ X_{3(n+1)} \end{pmatrix} = \begin{pmatrix} X_{1(n)} \\ X_{2(n)} \\ X_{3(n)} \end{pmatrix} - \frac{1}{n}\begin{pmatrix} a_{1(n)}(Y_{1(n)}-p_1) \\ a_{2(n)}(Y_{2(n)}-p_2) \\ a_{3(n)}(Y_{3(n)}-p_3) \end{pmatrix}. \tag{5.4}$$

By Theorem 2.1 and 2.2, $(x_{1(n)},x_{2(n)},x_{3(n)})'$ converges to $(L_{p_1},L_{p_2},L_{p_3})'$ almost surely and $a_{i(n)}$ converges to α_i^{-1} almost surely. By Lemma 2.3,

$$\sqrt{n}\begin{pmatrix} X_{1(n+1)}-L_{p_1} \\ X_{2(n+1)}-L_{p_2} \\ X_{3(n+1)}-L_{p_3} \end{pmatrix} \sim AN_3\left(\begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \sigma_1^2/\alpha_1^2 & 0 & 0 \\ 0 & \sigma_1^2/\alpha_1^2 & 0 \\ 0 & 0 & \sigma_1^2/\alpha_1^2 \end{pmatrix} \right).$$

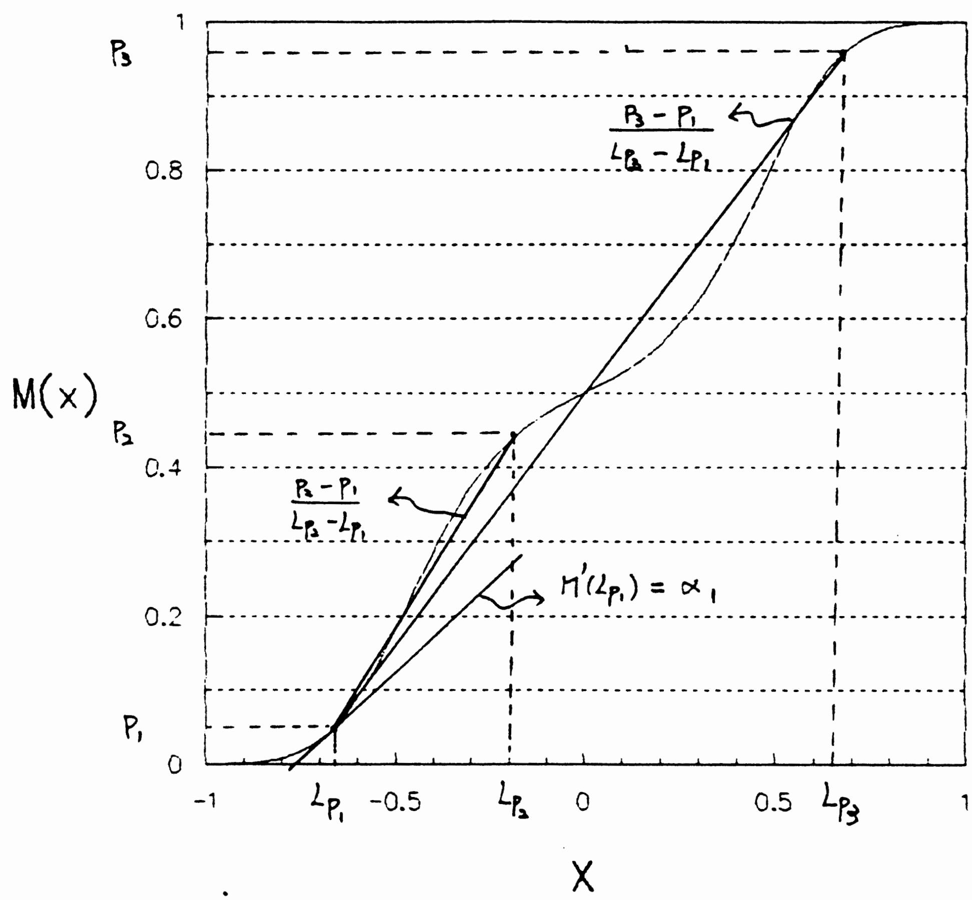

Figure 5.1 Relationship Between α_j and c_{ij} for Three Parameters Case

It is straightforward to generalize the three parameter case to r-parameter case where $r \geq 3$. Let $M(x) = F(x;\theta_1,\ldots,\theta_r)$ be an increasing function with r parameters. In order to estimate the whole curve, $(L_{p_1},\ldots,L_{p_r})$ is chosen to satisfy $M(L_{p_i}) = p_i$ and $\frac{\partial}{\partial x} M(L_{p_i}) = \alpha_i$ where $i=1,\ldots,r$. Similar to the three parameter case, α_i^{-1} can be estimated by

$$(x_{j(n)}-x_{i(n)})/[c_{ij}(p_j-p_i)], \quad i=1,\ldots,r, \quad j\neq i. \tag{5.5}$$

There are r-1 possible estimators for α_i^{-1}. Let

$$\hat{\alpha}^{-1}_{i(n)} = \frac{1}{r-1}\sum_{j\neq i}^{r} (x_{j(n)}-x_{i(n)})/[c_{ij}(p_j-p_i)] \tag{5.6}$$

$$= \sum_{j\neq i}^{r} d_{ij}x_{j(n)}.$$

The sequential procedure is given by

$$\begin{pmatrix} X_{1(n+1)} \\ \vdots \\ X_{r(n+1)} \end{pmatrix} = \begin{pmatrix} X_{1(n)} \\ \vdots \\ X_{r(n)} \end{pmatrix} - \frac{1}{n}\begin{pmatrix} a_{1(n)}(Y_{1(n)}-p_1) \\ \vdots \\ a_{r(n)}(Y_{r(n)}-p_r) \end{pmatrix} \tag{5.7}$$

where $a_{i(n)}$ is defined by equations (5.3) and (5.6). As in the three parameter case, it can be proved that $(x_{1(n)},\ldots, x_{r(n)})'$ converges to $(L_{p_1},\ldots,L_{p_r})'$ a.s., and $a_{i(n)}$ converges to α_i^{-1} a.s. where $i=1,\ldots,r$. Let $\mathbf{X}_{(n)}$ be the r-dimension random vector at stage n which is defined in (5.7) and $\mathbb{L}_p$ be the r-dimension root vector of M such that $M(L_{p_i})=p_i$ for each element L_{p_i}.

By Theorem 2.4, the following result holds

$$\sqrt{n}(\mathbf{X}_{(n)}-\mathbf{L}_p) \sim AN_r(\mathbf{0}, \mathbf{V}) \qquad (5.8)$$

where $\mathbf{0}$ be r×1 null vector, $\mathbf{V}$ be a r-dimension diagonal matrix with nonzero diagonal elements σ_i^2/α_i^2 for i = 1, 2,..., r.

Although all the theorems of the new procedure in chapter II are based on the 2-parameter case, they can be generalized through (5.3), (5.6), (5.7), and (5.8) for the r-parameter case.

Summary

The objective of this thesis is to estimate all roots of an increasing function M(x), that is, to estimate the whole curve M(x). Wetherill (1963) showed that, for a non-adaptive RM procedure, a good estimate of the root of M depends on a good initial guess and the constant A. By the simulation results, if the objective is to estimate a single root of M(x), Wu's 1-root procedure performs best. However, it performs poor when estimating other roots. If the objective is to estimate two or more roots, the new procedure and Wu's 2-root procedure perform substantially better than RM procedure and Anbar's 2-root procedure in initial design I. However, Wu's 2-root procedure performs poor in the initial design II. By the simulation outputs of initial design II, it shows that Anbar's procedure and

Wu's procedure do not performs very well for small sample sizes especially when prior information about the locations of percentiles of M(x) is not available. However, for the four 2-root finding procedure, only the new procedure perform well in both initial design I and initial design II. It is also noted that the estimate of the inverse of the tangent slope for Anbar's and Wu's procedures must be re-calculated when additional observations are obtained. However, in estimating (x_{n+1}, x'_{n+1}), (x_n, x'_n) is the unique observation which is needed for the new procedure. The previous observations (x_i, x'_i), $i=1,\ldots,n-1$, are not needed for the future iterations. This means that the new procedure has the benefit of being easy to calculate. It is helpful for the applications which require fast response. If the objective of an experiment is to estimate one root only, this new procedure is not recommended.

SELECTED BIBLIOGRAPHY

Albert, A. E. and Gardner, L. A. (1967). Stochastic Approximation and Nonlinear Regression. Research Monograph No. 42 MIT Press

Anbar, D. (1978) A Stochastic Newton-Raphson Method. J. Stat. Planning and Inference 2 : 153-163.

Blum, J. R. (1954) Approximation Method which Converges with Probability 1. Ann. Prob., 4 :890-901.

Chung, K. L. (1954). On a Stochastic Approximation Method. Ann. Math. Stat., 25 : 463-483.

Cox, D.R. (1970). Analysis of Binary Data. London : Chapman and Hall.

Dixon, W. J. and Mood, A. M. (1948). A Method for Obtaining and Analyzing Sensitivity Data. J. Amer. Stat. Asso., 43 : 109-126.

Dvoretzky, A. (1956) On Stochastic Approximation. proc. Third Berkley symptoms, Math. Stat. Prob., 1 : 39-56.

Finney, D. J. (1971) Probit Analysis. 3rd Ed. cambridge University Press.

Lai, T. L. and Robbins, H. (1978). Adaptive Design in Regression and Control. Proc. Natl. Acad. Sci, 75: 586-587.

Lai, T. L. and Robbins, H. (1979). Local Convergence Theorem for Adaptive Stochastic Approximation Schemes. Proc. Natl. Acad. Sci., 76: 3065-3067.

Lai, T. L. and Robbins, H. (1979). Adaptive Design and Stochastic Approximation. Ann. Stat., 7: 1196-1221

Lai, T. L. and Robbins, H. (1981) Consistency and Asymptotic Efficiency of Slope Estimates in Stochastic Approximation Schemes. Z. Wahrscheinlichkeitstheorie Verw. Gebiete, 56 : 329-360

Lord, N. C. (1975). Tailored Testing: An Application of Stochastic Approximation. J. Amer. Stat. Assoc., 66: 707-711.

Robbins, H. and Monro, S. (1951). A stochastic Approximation Method. Ann. Math. Stat., 22 : 400-407

Robbins, H. and Siegmund, D. (1971). A Convergent Theorem for Non-negative Almost Supermartingales and Some Applications. Opti. Methods in Stat. 233-257.

Sacks, J. (1958). Asymptotic Distribution of Stochastic Approximation Procedures. Ann. Math. Stat., 29 : 373-405.

Sakrison, D. J. (1965). Efficient Recursive Estimation: Application to Estimate the Parameters of a Covariance Function. Int. J. Eng. Sci., 3 : 461-483

Serfling, R. J. (1980). Approximation Theorems of Mathematical Statistics.

Silvapulle, M. J. (1981). On the Existence of Maximum Likelihood Estimators for the Binary Response Model. J. Royal Stat. Soc., Ser. B, 43 : 310-313.

Venter, J. H. (1967). An Extension of Robbin-Monro procedure. Ann. Math. Stat., 23 : 457-461.

Wetherill, G. B. (1963). Sequential Estimation of Quantal Response Curves. J. Royal Stat. Soc., Ser. B, 25 : 1-48.

Wu, C. F. (1985). Efficient Sequential Designs With Binary Data. J. Amer. Stat. Assoc., 80 : 794-984.

APPENDIXES

SIMULATION PROGRAM OF INITIAL DESIGN I WITH STARTING POINTS I FOR THE LOGIT TRUE MODEL

```
REAL msw20, msw80, msw50, msw75, mswc50, mswc75
REAL msa20, msa80, msa50, msa75, msac50, msac75
REAL msf20, msf80, msf50, msf75
REAL msr20, msr80, msr50, msr75
REAL l20, l80, l50, l75, mumle, mumlel, mumler, mumlec, mu, lb
REAL t,tl,tr,tc,tm,ts,thd
INTEGER*2 ih,im,is,ihd,lh,lm,ls,lhd
DIMENSION ub(4), brm(3), ul(110), ur(110), uc(210), n(4)
DIMENSION x(10), y(10), xlwul(110, 4), xrwul(110, 4), xlanb(4)
DIMENSION xlfei(4), xrfei(4), xlrm(3), xrrm(3), xranb(4)
DIMENSION xcwul(200, 4), sywc(4), sxywc(4)
DIMENSION xcanb(4), sxac(4), syac(4), sxxac(4), sxyac(4)
DIMENSION sywl(4), sywr(4), sxywl(4), sxywr(4)
DIMENSION syal(4), syar(4), sxyal(4), sxyar(4)
DIMENSION sxal(4), sxar(4), sxxal(4), sxxar(4)
DIMENSION ssw20(4), ssw80(4), ssw50(4), ssw75(4), sswc50(4)
DIMENSION sswc75(4), ssac75(4)
DIMENSION ssa20(4), ssa80(4), ssa50(4), ssa75(4), ssac50(4)
DIMENSION ssf20(4), ssf80(4), ssf50(4), ssf75(4)
DIMENSION ssr20(3), ssr80(3), ssr50(3), ssr75(3), mswc75(4)
DIMENSION msw20(4), msw80(4), msw50(4), msw75(4), mswc50(4)
DIMENSION msa20(4), msa80(4), msa50(4), msa75(4), msac50(4)
DIMENSION msf20(4), msf80(4), msf50(4), msf75(4), msac75(4)
DIMENSION msr20(3), msr80(3), msr50(3), msr75(3)
DATA lb,ub(1),ub(2),ub(3),ub(4)/.005,36.,50.,100.,200./
DATA brm(1),brm(2),brm(3)/1.,6.,36./
DATA p1,p2,p3,p4,p5/.1,.3,.5,.7,.9/
DATA n(1),n(2),n(3),n(4)/15,30,50,100/
a=1./3.
b=3./5.
c=5./7.
d=7./13.
OPEN(UNIT=5,FILE='h:\prog\for\flogit5.out')
DO 99999 ndata = 1 , 4
  nouse = 0
  nsimu = 0
  timew = 0.
  timew2 = 0.
  timea = 0.
  timea2 = 0.
  timef = 0.
  timerm = 0.
DO 100 j = 1 , 4
  ssw20(j) = 0.
```

```
          ssw80(j) = 0.
          ssw50(j) = 0.
          ssw75(j) = 0.
          ssa20(j) = 0.
          ssa80(j) = 0.
          ssa50(j) = 0.
          ssa75(j) = 0.
          ssf20(j) = 0.
          ssf80(j) = 0.
          ssf50(j) = 0.
          ssf75(j) = 0.
          sswc50(j) = 0.
          sswc75(j) = 0.
          ssac50(j) = 0.
          ssac75(j) = 0.
100     CONTINUE
        DO 200 j = 1 , 3
          ssr20(j) = 0.
          ssr80(j) = 0.
          ssr50(j) = 0.
          ssr75(j) = 0.
200     CONTINUE
        l20 = -LOG(4.)
        l80 = LOG(4.)
        l50 = 0.
        l75 = LOG(3.)
        x1 = LOG(p1 / (1. - p1))
        x2 = LOG(p2 / (1. - p2))
        x3 = LOG(p3 / (1. - p3))
        x4 = LOG(p4 / (1. - p4))
        x5 = LOG(p5 / (1. - p5))
c       =========================
c         Simulations 500 times
c       =========================
98        sx = 0.
          sy = 0.
          sxy = 0.
          syy = 0.
          sp = 0.
          spp = 0.
          sxp = 0.
          sxpp = 0.
          sxxpp = 0.
          bmle = 0.
c         ========================
c          Generate y(1) to y(10)
c         ========================
99        CALL RND(a,b,c,d,unirnd)
          IF (unirnd .LT. p1) THEN
             y(1) = 1.
          ELSE
             y(1) = 0.
          ENDIF
          x(1) = x1
```

```
      DO 300 i = 2 , 3
        CALL RND(a,b,c,d,unirnd)
        IF (unirnd .LT. p2) THEN
          y(i) = 1.
        ELSE
          y(i) = 0.
        ENDIF
        x(i) = x2
300   CONTINUE
      DO 400 i = 4 , 7
        CALL RND(a,b,c,d,unirnd)
        IF (unirnd .LT. p3) THEN
          y(i) = 1.
        ELSE
          y(i) = 0.
        ENDIF
        x(i) = x3
400   CONTINUE
      DO 500 i = 8 , 9
        CALL RND(a,b,c,d,unirnd)
        IF (unirnd .LT. p4) THEN
          y(i) = 1.
        ELSE
          y(i) = 0.
        ENDIF
        x(i) = x4
500   CONTINUE
      CALL RND(a,b,c,d,unirnd)
      IF (unirnd .LT. p5) THEN
        y(10) = 1.
      ELSE
        y(10) = 0.
      ENDIF
      x(10) = x5
c     ===============================
c      Test silvapulle's conditions
c     ===============================
      min0 = 10
      max0 = 1
      min1 = 10
      max1 = 1
      DO 600 k = 1 , 10
        IF ((y(k) .EQ. 0.) .AND. (k .GT. max0)) THEN
          max0 = k
        END IF
        IF ((y(k) .EQ. 0.) .AND. (k .LT. min0)) THEN
          min0 = k
        ENDIF
        IF ((y(k) .EQ. 1.) .AND. (k .GT. max1)) THEN
          max1 = k
        ENDIF
        IF ((y(k) .EQ. 1.) .AND. (k .LT. min1)) THEN
          min1 = k
        ENDIF
```

```
600     CONTINUE
        IF((x(min1) .GT. x(max0)) .OR. (x(min0) .GT. x(max1))) THEN
          index = 0
        ELSE
          index = 1
        ENDIF
          nsimu=nsimu+1
        IF (index .EQ. 0) THEN
          nouse = nouse + 1
          GO TO 99
        ENDIF
c       ===========================================
c        Estimate mu and beta by the first 10 obs.
c       ===========================================
        mu = 0.
        beta = 1.
        nt = 1
        grad = 100.
        IF ((grad .GT. .0001) .AND. (nt .LE. 10)) THEN
          sx = 0.
          sy = 0.
          sxy = 0.
          sxx = 0.
          sp = 0.
          spp = 0.
          sxp = 0.
          sxpp = 0.
          sxxpp = 0.
          DO 700 i = 1 , 10
            t = mu + beta * x(i)
            if(t .GE. 20.) then
              pt = 1.
            else
              pt = exp(t) / (1. + exp(t))
            endif
            sx = sx + x(i)
            sxx = sxx + x(i) * x(i)
            sy = sy + y(i)
            sxy = sxy + x(i) * y(i)
            sp = sp + pt
            spp = spp + pt * (1. - pt)
            sxp = sxp + x(i) * pt
            sxpp = sxpp + x(i) * pt * (1. - pt)
            sxxpp = sxxpp + x(i) * x(i) * pt * (1. - pt)
700       CONTINUE
          det = sxxpp * spp - sxpp * sxpp
          if(det .LT. .001) then
            nouse = nouse + 1
            go to 99
          endif
          debeta = (sxpp * (sy - sp) - spp * (sxy - sxp)) / det
          demu = (sxpp * (sxy - sxp) - sxxpp * (sy - sp)) / det
          mu = mu - demu
          beta = beta - debeta
```

```
            grad = (sy - sp) ** 2 + (sxy - sxp) ** 2
            nt = nt + 1
          ENDIF
          bmle = beta
          mumle = mu
          IF (bmle .LE. 0.1) THEN
            nouse = nouse + 1
            GO TO 99
          ENDIF
          xl = (-mumle - LOG(4.)) / bmle
          xr = (-mumle + LOG(4.)) / bmle
          xc = -mumle / bmle
c         ================================================================
c          Generate n - 11 uniform random numbers each for estimating l20,
c          L80. Also using these r.n.'s to estimate L50 with 2(n-11) obs.
c         ================================================================
        DO 800 i = 11 , n(ndata) - 1
          CALL RND(a,b,c,d,ul(i))
          CALL RND(a,b,c,d,ur(i))
800     CONTINUE
        DO 900 i = 11 , n(ndata) - 1
          uc(2 * i - 11) = ul(i)
          uc(2 * i - 10) = ur(i)
900     CONTINUE
c
c       =========================
c        Simulate Wu's procedure
c       =========================
c
        CALL GETTIM(ih,im,is,ihd)
        DO 1500 j = 1 , 4
          sywl(j) = sy
          sywr(j) = sy
          sxywl(j) = sxy
          sxywr(j) = sxy
          DO 1000 i = 1 , 10
            xlwul(i, j) = x(i)
            xrwul(i, j) = x(i)
1000      CONTINUE
          xlwul(11, j) = xl
          xrwul(11, j) = xr
          bmlel = bmle
          bmler = bmle
          mumlel = mumle
          mumler = mumle
c         ====================================
c          Bounded 1/slope = 36, 50, 100, 200
c         ====================================
        DO 1400 i = 11 , n(ndata) - 1
          tl = EXP(xlwul(i, j))
          tr = EXP(xrwul(i, j))
          pl = tl / (1. + tl)
          pr = tr / (1. + tr)
          IF (ul(i) .LT. pl) THEN
```

```
        yl = 1.
      ELSE
        yl = 0.
      ENDIF
      IF (ur(i) .LT. pr) THEN
        yr = 1.
      ELSE
        yr = 0.
      ENDIF
      sywl(j) = sywl(j) + yl
      sywr(j) = sywr(j) + yr
      sxywl(j) = sxywl(j) + xlwul(i, j) * yl
      sxywr(j) = sxywr(j) + xrwul(i, j) * yr
c     ====================
c      Estimate Mu & Beta
c     ====================
      ntl = 1
      gradl = 100.
    IF ((gradl .GT. .0001) .AND. (ntl .LE. 10)) THEN
      sxpl = 0.
      spl = 0.
      sppl = 0.
      sxppl = 0.
      sxxppl = 0.
      DO 1100 ki = 1 , i
        tl = mumlel + bmlel * xlwul(ki, j)
        if(tl .GE. 20.) then
          pl = 1.
        else
          pl = exp(tl) / (1. + exp(tl))
        endif
        sxpl = sxpl + xlwul(ki, j) * pl
        spl = spl + pl
        sppl = sppl + pl * (1. - pl)
        sxppl = sxppl + xlwul(ki, j) * pl * (1. - pl)
        sxxppl = sxxppl + xlwul(ki, j) ** 2 * pl * (1. - pl)
1100  CONTINUE
      detl = sxxppl * sppl - sxppl * sxppl
      if(detl .LT. .001) then
        nouse = nouse + 1
        go to 99
      endif
      be1 = sxppl * (sywl(j) - spl)
      be2 = sppl * (sxywl(j) - sxpl)
      rm1 = sxppl * (sxywl(j) - sxpl)
      rm2 = sxxppl * (sywl(j) - spl)
      bmlel = bmlel - (be1 - be2) / detl
      mumlel = mumlel - (rm1 - rm2) / detl
      gradl = (sywl(j) - spl) ** 2 + (sxywl(j) - sxpl) ** 2
      ntl = ntl + 1
    ENDIF
      ntr = 1
      gradr = 100.
    IF ((gradr .GT. .0001) .AND. (ntr .LE. 10)) THEN
```

```
        sxpr = 0.
        spr = 0.
        sppr = 0.
        sxppr = 0.
        sxxppr = 0.
        DO 1200 ki = 1 , i
          tr = mumler + bmler * xrwul(ki, j)
          if(tr .GE. 20.) then
            pr = 1.
          else
            pr = exp(tr) / (1. + exp(tr))
          endif
          sxpr = sxpr + xrwul(ki, j) * pr
          spr = spr + pr
          sppr = sppr + pr * (1. - pr)
          sxppr = sxppr + xrwul(ki, j) * pr * (1. - pr)
          sxxppr = sxxppr + xrwul(ki, j) ** 2 * pr * (1. - pr)
1200    CONTINUE
        detr = sxxppr * sppr - sxppr * sxppr
        if(detr .LT. .001) then
          nouse = nouse + 1
          go to 99
        endif
        be1 = sxppr * (sywr(j) - spr)
        be2 = sppr * (sxywr(j) - sxpr)
        rm1 = sxppr * (sxywr(j) - sxpr)
        rm2 = sxxppr * (sywr(j) - spr)
        bmler = bmler - (be1 - be2) / detr
        mumler = mumler - (rm1 - rm2) / detr
        gradr = (sywr(j) - spr) ** 2 + (sxywr(j) - sxpr) ** 2
        ntr = ntr + 1
      ENDIF
        xlwu2 = (-mumlel - LOG(4)) / bmlel
        xrwu2 = (-mumler + LOG(4)) / bmler
c     ========================================
c      Bound the inverse of tangent slopes
c     ========================================
        cnmlel = (xlwul(i, j) - xlwu2) * i / (yl - .2)
        cnmler = (xrwul(i, j) - xrwu2) * i / (yr - .8)
        IF (cnmlel .LE. lb) THEN
          cnmlel = lb
        ELSEIF (cnmlel .GE. ub(j)) THEN
          cnmlel = ub(j)
        ENDIF
        IF (cnmler .LE. lb) THEN
          cnmler = lb
        ELSEIF (cnmler .GE. ub(j)) THEN
          cnmler = ub(j)
        ENDIF
        xlwu2 = xlwul(i, j) - (yl - .2) * cnmlel / i
        xrwu2 = xrwul(i, j) - (yr - .8) * cnmler / i
        xlwul(i + 1, j) = xlwu2
        xrwul(i + 1, j) = xrwu2
1400  CONTINUE
```

```
          c50 = .5
          c75 = LOG(4. / 3.) / LOG(16.)
          nndata=n(ndata)
          wul50 = c50 * (xlwul(nndata, j) + xrwul(nndata, j))
          wul75 = c75 * xlwul(nndata, j) + (1. - c75) * xrwul(nndata, j)
          ssw20(j) = ssw20(j) + (xlwul(nndata, j) - 120) ** 2
          ssw80(j) = ssw80(j) + (xrwul(nndata, j) - 180) ** 2
          ssw50(j) = ssw50(j) + (wul50 - 150) ** 2
          ssw75(j) = ssw75(j) + (wul75 - 175) ** 2
1500    CONTINUE
        CALL GETTIM(lh,lm,ls,lhd)
        IF(lm .LT. im) THEN
          tm = 59. - im + lm
          ts = 59. - is + ls
          thd = 100. - ihd + lhd
          timew = timew + 60. * tm + ts + thd / 100.
        ELSE
          tm = lm - im - 1.
          ts = 59. - is + ls
          thd = 100. - ihd + lhd
          timew = timew + 60. * tm + ts + thd / 100.
        ENDIF
c       =====================================
c        Estimate L(0.5) for 2n observations
c       =====================================
        CALL GETTIM(ih,im,is,ihd)
        DO 2000 j = 1 , 4
          sywc(j) = sy
          sxywc(j) = sxy
          DO 1550 i = 1 , 10
            xcwul(i, j) = x(i)
1550      CONTINUE
          xcwul(11, j) = xc
          bmlec = bmle
          mumlec = mumle
          iter2n = 2 * n(ndata) - 12
        DO 1900 i = 11 , iter2n
          tc = EXP(xcwul(i, j))
          pc = tc / (1. + tc)
          IF (uc(i) .LT. pc) THEN
            yc = 1.
          ELSE
            yc = 0.
          ENDIF
          sywc(j) = sywc(j) + yc
          sxywc(j) = sxywc(j) + xcwul(i, j) * yc
c       ====================
c        Estimate Mu & Beta
c       ====================
          ntc = 1
          gradc = 100.
        IF ((gradc .GT. .0001) .AND. (ntc .LE. 10)) THEN
          sxpc = 0.
          spc = 0.
```

```
           sppc = 0.
           sxppc = 0.
           sxxppc = 0.
           DO 1600 ki = 1 , i
             tc = mumlec + bmlec * xcwul(ki, j)
             if(tc .GE. 20.) then
               pc = 1.
             else
               pc = exp(tc) / (1. + exp(tc))
             endif
             sxpc = sxpc + xcwul(ki, j) * pc
             spc = spc + pc
             sppc = sppc + pc * (1. - pc)
             sxppc = sxppc + xcwul(ki, j) * pc * (1. - pc)
             sxxppc = sxxppc + xcwul(ki, j) ** 2 * pc * (1. - pc)
1600       CONTINUE
           detc = sxxppc * sppc - sxppc * sxppc
           if(detc .LT. .001) then
             nouse = nouse + 1
             go to 99
           endif
           be1 = sxppc * (sywc(j) - spc)
           be2 = sppc * (sxywc(j) - sxpc)
           rm1 = sxppc * (sxywc(j) - sxpc)
           rm2 = sxxppc * (sywc(j) - spc)
           bmlec = bmlec - (be1 - be2) / detc
           mumlec = mumlec - (rm1 - rm2) / detc
           gradc = (sywc(j) - spc) ** 2 + (sxywc(j) + sxpc) ** 2
           ntc = ntc + 1
         ENDIF
           xcwu2 = -mumlec / bmlec
c        ========================================
c         Bound the inverse of tangent slopes
c        ========================================
           cnmlec = (xcwul(i, j) - xcwu2) * i / (yc - .5)
           IF (cnmlec .LE. lb) THEN
             cnmlec = lb
           ELSEIF (cnmlec .GE. ub(j)) THEN
             cnmlec = ub(j)
           ENDIF
           xcwu2 = xcwul(i, j) - (yc - .5) * cnmlec / i
           xcwul(i + 1, j) = xcwu2
1900     CONTINUE
           sswc50(j) = sswc50(j) + (xcwul(iter2n + 1, j) - 150) ** 2
           sswc75(j) = sswc75(j) + ((LOG(3) - mumlec) / bmlec - 175) ** 2
2000     CONTINUE
         CALL GETTIM(lh,lm,ls,lhd)
         IF(lm .LT. im) THEN
           tm = 59. - im + lm
           ts = 59. - is + ls
           thd = 100. - ihd + lhd
           timew2 = timew2 + 60. * tm + ts + thd / 100.
         ELSE
           tm = lm - im - 1.
```

```
          ts = 59. - is + ls
          thd = 100. - ihd + lhd
          timew2 = timew2 + 60. * tm + ts + thd / 100.
        ENDIF
c
c       =============================
c        Simulate Anbar's procedure
c       =============================
c
        CALL GETTIM(ih,im,is,ihd)
        DO 2500 j = 1 , 4
          sxal(j) = sx
          sxar(j) = sx
          sxxal(j) = sxx
          sxxar(j) = sxx
          syal(j) = sy - 2.
          syar(j) = sy - 8.
          sxyal(j) = sxy - .2 * sx
          sxyar(j) = sxy - .8 * sx
          xlanb(j) = xl
          xranb(j) = xr
c       ====================================
c        Bounded 1/slpoe = 36, 50, 100, 200
c       ====================================
        DO 2400 i = 11 , n(ndata) - 1
          tl = EXP(xlanb(j))
          tr = EXP(xranb(j))
          pl = tl / (1. + tl)
          pr = tr / (1. + tr)
          IF (ul(i) .LT. pl) THEN
            ylanb = 1.
          ELSE
            ylanb = 0.
          ENDIF
          IF (ur(i) .LT. pr) THEN
            yranb = 1.
          ELSE
            yranb = 0.
          ENDIF
          sxal(j) = sxal(j) + xlanb(j)
          sxar(j) = sxar(j) + xranb(j)
          sxxal(j) = sxxal(j) + xlanb(j) * xlanb(j)
          sxxar(j) = sxxar(j) + xranb(j) * xranb(j)
          syal(j) = syal(j) + ylanb - .2
          syar(j) = syar(j) + yranb - .8
          sxyal(j) = sxyal(j) + xlanb(j) * (ylanb - .2)
          sxyar(j) = sxyar(j) + xranb(j) * (yranb - .8)
          rnumbl = i * sxyal(j) - sxal(j) * syal(j)
          banbl = rnumbl / (i * sxxal(j) - sxal(j) * sxal(j))
          rnumbr = i * sxyar(j) - sxar(j) * syar(j)
          banbr = rnumbr / (i * sxxar(j) - sxar(j) * sxar(j))
          IF (banbl .LE. (1. / ub(j))) THEN
            cnanbl = ub(j)
          ELSEIF (banbl .GE. (1. / lb)) THEN
```

```
        cnanbl = lb
      ELSE
        cnanbl = 1. / banbl
      ENDIF
      IF (banbr .LE. (1. / ub(j))) THEN
        cnanbr = ub(j)
      ELSEIF (banbr .GE. (1. / lb)) THEN
        cnanbr = lb
      ELSE
        cnanbr = 1. / banbr
      ENDIF
      xlanb(j) = xlanb(j) - (ylanb - .2) * cnanbl / i
      xranb(j) = xranb(j) - (yranb - .8) * cnanbr / i
2400  CONTINUE
      anbl50 = c50 * (xlanb(j) + xranb(j))
      anbl75 = c75 * xlanb(j) + (1. - c75) * xranb(j)
      ssa20(j) = ssa20(j) + (xlanb(j) - 120) ** 2
      ssa80(j) = ssa80(j) + (xranb(j) - 180) ** 2
      ssa50(j) = ssa50(j) + (anbl50 - 150) ** 2
      ssa75(j) = ssa75(j) + (anbl75 - 175) ** 2
2500  CONTINUE
      CALL GETTIM(lh,lm,ls,lhd)
      IF(lm .LT. im) THEN
        tm = 59. - im + lm
        ts = 59. - is + ls
        thd = 100. - ihd + lhd
        timea = timea + 60. * tm + ts + thd / 100.
      ELSE
        tm = lm - im - 1.
        ts = 59. - is + ls
        thd = 100. - ihd + lhd
        timea = timea + 60. * tm + ts + thd / 100.
      ENDIF
c     =====================================
c      Estimate L(0.5) for 2n observations
c     =====================================
      CALL GETTIM(ih,im,is,ihd)
      DO 3000 j = 1 , 4
        xcanb(j) = xc
        sxac(j) = sx
        syac(j) = sy - .5 * sx
        sxxac(j) = sxx
        sxyac(j) = sxy - .5 * sx
      DO 2900 i = 11 , iter2n
        tc = EXP(xcanb(j))
        pc = tc / (1. + tc)
        IF (uc(i) .LT. pc) THEN
          ycanb = 1.
        ELSE
          ycanb = 0.
        ENDIF
        sxac(j) = sxac(j) + xcanb(j)
        sxxac(j) = sxxac(j) + xcanb(j) * xcanb(j)
        syac(j) = syac(j) + ycanb - .5
```

```
        sxyac(j) = sxyac(j) + xcanb(j) * (ycanb - .5)
        rnumbc = i * sxyac(j) - sxac(j) * syac(j)
        banbc = rnumbc / (i * sxxac(j) - sxac(j) * sxac(j))
        IF (banbc .LE. (1. / ub(j))) THEN
          cnanbc = ub(j)
        ELSEIF (banbc .GE. (1. / lb)) THEN
          cnanbc = lb
        ELSE
          cnanbc = 1. / banbc
        ENDIF
        xcanb(j) = xcanb(j) - (ycanb - .5) * cnanbc / i
2900  CONTINUE
        ssac50(j) = ssac50(j) + (xcanb(j) - 150) ** 2
        tterm = xcanb(j) + LOG(3) * cnanbc / 4.
        ssac75(j) = ssac75(j) + (tterm - 175) ** 2
3000  CONTINUE
      CALL GETTIM(lh,lm,ls,lhd)
      IF(lm .LT. im) THEN
        tm = 59. - im + lm
        ts = 59. - is + ls
        thd = 100. - ihd + lhd
        timea2 = timea2 + 60. * tm + ts + thd / 100.
      ELSE
        tm = lm - im - 1.
        ts = 59. - is + ls
        thd = 100. - ihd + lhd
        timea2 = timea2 + 60. * tm + ts + thd / 100.
      ENDIF
c
c     ===========================
c      Simulate Fei's procedure
c     ===========================
c
      CALL GETTIM(ih,im,is,ihd)
      DO 3500 j = 1 , 4
        rk = 1. / (2 * .16 * LOG(4))
        xlfei(j) = xl
        xrfei(j) = xr
c       ====================================
c        Bounded 1/slope = 36, 50, 100, 200
c       ====================================
      DO 3400 i = 11 , n(ndata) - 1
        cnfei = rk * (xrfei(j) - xlfei(j))
        IF (cnfei .LE. lb) THEN
          cnfei = lb
        ELSEIF (cnfei .GE. ub(j)) THEN
          cnfei = ub(j)
        ENDIF
        tl = EXP(xlfei(j))
        tr = EXP(xrfei(j))
        pl = tl / (1. + tl)
        pr = tr / (1. + tr)
        IF (ul(i) .LT. pl) THEN
          ylfei = 1.
```

```
        ELSE
          ylfei = 0.
        ENDIF
        IF (ur(i) .LT. pr) THEN
          yrfei = 1.
        ELSE
          yrfei = 0.
        ENDIF
        xlfei2 = xlfei(j) - cnfei * (ylfei - .2) / i
        xrfei2 = xrfei(j) - cnfei * (yrfei - .8) / i
        IF (xrfei2 .LE. xlfei2) THEN
          xlfei2 = xlfei(j)
          xrfei2 = xrfei(j)
        ENDIF
        xlfei(j) = xlfei2
        xrfei(j) = xrfei2
3400  CONTINUE
        feil50 = c50 * (xlfei(j) + xrfei(j))
        feil75 = c75 * xlfei(j) + (1. - c75) * xrfei(j)
        ssf20(j) = ssf20(j) + (xlfei(j) - 120) ** 2
        ssf80(j) = ssf80(j) + (xrfei(j) - 180) ** 2
        ssf50(j) = ssf50(j) + (feil50 - 150) ** 2
        ssf75(j) = ssf75(j) + (feil75 - 175) ** 2
3500  CONTINUE
      CALL GETTIM(lh,lm,ls,lhd)
      IF(lm .LT. im) THEN
        tm = 59. - im + lm
        ts = 59. - is + ls
        thd = 100. - ihd + lhd
        timef = timef + 60. * tm + ts + thd / 100.
      ELSE
        tm = lm - im - 1.
        ts = 59. - is + ls
        thd = 100. - ihd + lhd
        timef = timef + 60. * tm + ts + thd / 100.
      ENDIF
c
c     ========================
c      Simulate RM procedure
c     ========================
c
      CALL GETTIM(ih,im,is,ihd)
      DO 4000 j = 1 , 3
        xlrm(j) = xl
        xrrm(j) = xr
c       ============================
c        Bounded C value = 1, 6, 36
c       ============================
      DO 3900 i = 11 , n(ndata) - 1
        tl = EXP(xlrm(j))
        tr = EXP(xrrm(j))
        pl = tl / (1. + tl)
        pr = tr / (1. + tr)
        IF (ul(i) .LT. pl) THEN
```

```
          ylrm = 1.
        ELSE
          ylrm = 0.
        ENDIF
        IF(ur(i) .LT. pr) THEN
          yrrm = 1.
        ELSE
          yrrm = 0.
        ENDIF
        xlrm(j) = xlrm(j) - brm(j) * (ylrm - .2) / i
        xrrm(j) = xrrm(j) - brm(j) * (yrrm - .8) / i
3900  CONTINUE
        rml50 = c50 * (xlrm(j) + xrrm(j))
        rml75 = c75 * xlrm(j) + (1 - c75) * xrrm(j)
        ssr20(j) = ssr20(j) + (xlrm(j) - 120) ** 2
        ssr80(j) = ssr80(j) + (xrrm(j) - 180) ** 2
        ssr50(j) = ssr50(j) + (rml50 - 150) ** 2
        ssr75(j) = ssr75(j) + (rml75 - 175) ** 2
4000  CONTINUE
      CALL GETTIM(lh,lm,ls,lhd)
      IF(lm .LT. im) THEN
        tm = 59. - im + lm
        ts = 59. - is + ls
        thd = 100. - ihd + lhd
        timerm = timerm + 60. * tm + ts + thd / 100.
      ELSE
        tm = lm - im - 1.
        ts = 59. - is + ls
        thd = 100. - ihd + lhd
        timerm = timerm + 60. * tm + ts + thd / 100.
      ENDIF
        write(0,59) nsimu,ndata
59      FORMAT(1x,i4,'th of simulations for ',i1,'th data set')
      IF(nsimu .LE. 500) GO TO 98
c     ===========================
c      Calculate the SQRT(MSE)'s
c     ===========================
        ntrue = nsimu - nouse
      DO 4500 j = 1 , 4
        mswc50(j) = SQRT(sswc50(j) / ntrue)
        mswc75(j) = SQRT(sswc75(j) / ntrue)
        msw20(j) = SQRT(ssw20(j) / ntrue)
        msw80(j) = SQRT(ssw80(j) / ntrue)
        msw50(j) = SQRT(ssw50(j) / ntrue)
        msw75(j) = SQRT(ssw75(j) / ntrue)
        msac50(j) = SQRT(ssac50(j) / ntrue)
        msac75(j) = SQRT(ssac75(j) / ntrue)
        msa20(j) = SQRT(ssa20(j) / ntrue)
        msa80(j) = SQRT(ssa80(j) / ntrue)
        msa50(j) = SQRT(ssa50(j) / ntrue)
        msa75(j) = SQRT(ssa75(j) / ntrue)
        msf20(j) = SQRT(ssf20(j) / ntrue)
        msf80(j) = SQRT(ssf80(j) / ntrue)
        msf50(j) = SQRT(ssf50(j) / ntrue)
```

```
        msf75(j) = SQRT(ssf75(j) / ntrue)
4500  continue
      DO 5000 j = 1 , 3
        msr20(j) = SQRT(ssr20(j) / ntrue)
        msr80(j) = SQRT(ssr80(j) / ntrue)
        msr50(j) = SQRT(ssr50(j) / ntrue)
        msr75(j) = SQRT(ssr75(j) / ntrue)
5000  CONTINUE
      WRITE(5,1) n(ndata)
1     FORMAT(1x,'   # of iterations is        ',I5)
      WRITE(5,2) nouse
2     FORMAT(1x,'   # of discard samples      ',I5)
      WRITE(5,3) nsimu
3     FORMAT(1x,'   # of simulations          ',I5)
      WRITE(5,4) timew,timew2
4     FORMAT(1x,'   Time needed for Wu proc.   ',2f10.3)
      WRITE(5,5) timea,timea2
5     FORMAT(1x,'   Time needed for Anbar proc.',2f10.3)
      WRITE(5,6) timef
6     FORMAT(1x,'   Time needed for Fei proc.  ',f10.3)
      WRITE(5,7) timerm
7     FORMAT(1x,'   Time needed for RM proc.   ',f10.3)
      WRITE(5,8)
8     FORMAT(1x,70('='))
      WRITE(5,9)
9     FORMAT(1x,' The horizontal output sequence of SQRT(MSE) is L20,')
      WRITE(5,10)
10    FORMAT(1x,' L80,L50,L75,and L50(2(n-11)), L75(2(n-11)).')
      WRITE(5,8)
      WRITE(5,11) msw20(1),msw80(1),msw50(1),msw75(1),mswc50(1),
     *            mswc75(1)
11    FORMAT(1x,' Wu36:     ',6F10.5)
      WRITE(5,12) msw20(2),msw80(2),msw50(2),msw75(2),mswc50(2),
     *            mswc75(2)
12    FORMAT(1x,' Wu50:     ',6F10.5)
      WRITE(5,13) msw20(3),msw80(3),msw50(3),msw75(3),mswc50(3),
     *            mswc75(3)
13    FORMAT(1x,' Wu100:    ',6F10.5)
      WRITE(5,14) msw20(4),msw80(4),msw50(4),msw75(4),mswc50(4),
     *            mswc75(4)
14    FORMAT(1x,' Wu200:    ',6F10.5)
      write(5,15)
15    FORMAT(1x,70('-'))
      WRITE(5,16) msa20(1),msa80(1),msa50(1),msa75(1),msac50(1),
     *            msac75(1)
16    FORMAT(1x,' Anb36:    ',6F10.5)
      WRITE(5,17) msa20(2),msa80(2),msa50(2),msa75(2),msac50(2),
     *            msac75(2)
17    FORMAT(1x,' Anb50:    ',6F10.5)
      WRITE(5,18) msa20(3),msa80(3),msa50(3),msa75(3),msac50(3),
     *            msac75(3)
18    FORMAT(1x,' Anb100:   ',6F10.5)
      WRITE(5,19) msa20(4),msa80(4),msa50(4),msa75(4),msac50(4),
     *            msac75(4)
```

```
19    FORMAT(1x,' Anb200:   ',6F10.5)
      WRITE(5,15)
      WRITE(5,20) msf20(1),msf80(1),msf50(1),msf75(1)
20    FORMAT(1x,' Fei36:    ',4F10.5)
      WRITE(5,21) msf20(2),msf80(2),msf50(2),msf75(2)
21    FORMAT(1x,' Fei50:    ',4F10.5)
      WRITE(5,22) msf20(3),msf80(3),msf50(3),msf75(3)
22    FORMAT(1x,' Fei100:   ',4F10.5)
      WRITE(5,23) msf20(4),msf80(4),msf50(4),msf75(4)
23    FORMAT(1x,' Fei200:   ',4F10.5)
      WRITE(5,15)
      WRITE(5,24) msr20(1),msr80(1),msr50(1),msr75(1)
24    FORMAT(1x,' RM1:      ',4F10.5)
      WRITE(5,25) msr20(2),msr80(2),msr50(2),msr75(2)
25    FORMAT(1x,' RM6:      ',4F10.5)
      WRITE(5,26) msr20(3),msr80(3),msr50(3),msr75(3)
26    FORMAT(1x,' RM36:     ',4F10.5)
      WRITE(5,27)
27    FORMAT(1x,////)
99999 CONTINUE
      CLOSE(5)
      STOP
      END
c     ============================================
c      Uniform random number generator subroutine
c     ============================================
      SUBROUTINE RND(a,b,c,d,r)
      r=a+b+c+d
      r=r-INT(r)
      a=b
      b=c
      c=d
      d=(1.-r)*11.11111111
      RETURN
      END
```

VITA

WuChen Fei

Candidate for the Degree of

Doctor of Philosophy

Thesis: A NEW METHOD FOR THE ROBBINS-MONRO STOCHASTIC APPROXIMATION PROCEDURE

Major Field: Statistics

Biographical:

Personal Data: Born in Taipei, Taiwan, Republic of China, December 18, 1955, the second son of Mr. and Mrs. Tzu-Juin Fei.

Education: Graduated from Pan-Chao High School, Pan-Chao, Taipei, Taiwan, in June 1973; received Bachelor of Business Administration from National Defense Management College, Taipei, Taiwan, in August 1978; received Master of Management Sciences from Tamkang University, Taipei, Taiwan, in May 1982; completed requirements for the Doctor of Philosophy degree at Oklahoma State University, in July, 1989.

Professional Experience: Teaching Assistant, Department of Statistics, National Defense Management College, 1978-1980; Instructor, Department of Statistics, National Defense Management College, 1982-1986.

Professional Organizations: Chinese System Analysis Association.