2002

Estimation in generalized linear models and time series models with nonparametric correlation coefficients

HuaiQing Sheng
The University of Montana

Estimation in Generalized Linear Models and Time Series Models with Nonparametric Correlation Coefficients

by

HuaiQing Sheng

M.A. The University of Montana, 1994
M.S. Changsha Institute of Technology, 1988
B.S. National University of Defense Technology, 1985

presented in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

The University of Montana

January 2002

Approved by:

Rudy A. Gideon
Chairperson

Dean, Graduate School

3-6-02
Date

ProQuest Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

Sheng, HuaiQing, Ph.D., January 2002 Mathematical Sciences

Estimation in Generalized Linear Models and Time Series Models with Nonparametric Correlation Coefficients

Director: Dr. Rudy A. Gideon

ABSTRACT

In this dissertation we propose estimation procedures for generalized linear models and time series models with nonparametric correlation coefficients, addressing the issues of prediction, estimation and quality control. Nonparametric correlation coefficients are introduced in the present study as a comprehensive robust statistical tool. In particular, the method of estimation is valid for any correlation coefficient, but it will be illustrated using the Greatest Deviation correlation coefficient, r_{gd}. Parameter estimation in generalized linear models and time series models can be performed using nonparametric correlation coefficients. The methodology is demonstrated using health care management data. Subsequently we discuss the estimation method for generalized linear models, nonlinear models, and time series with nonparametric correlation coefficients. One reason for using a nonparametric correlation coefficient is to have the conclusions valid under a wider class of bivariate distributions. Another reason is that the estimation process of r_{gd} regression adapts the robustness of the nonparametric correlation coefficient. Parameter estimates obtained for several data sets and through simulation show that the new methodology compares favorably with other general least squares and likelihood estimation methods, when the data are good, but performs robustly when the data have numerous suspect values.

KEY WORDS: Generalized Linear Model, Nonlinear Model, Time Series, Nonparametric Correlation Coefficients, Greatest Deviation Correlation Coefficient, Maximum Likelihood

Acknowledgments

I would like to express my deepest gratitude to my advisor, Professor Rudy Gideon, whose advice, constructive criticism, and encouragement made it possible to develop and complete this dissertation. His knowledge, insight and enthusiasm in statistics has been inspiring, and his guidance, invaluable.

I am very grateful to the committee members, Dr. Jon Graham, Dr. William Derrick, Dr. Tomas Tonev and Dr. Alden Wright. They all advised me and encouraged me during my graduate studies. Their constructive remarks and suggestions improved this dissertation.

I am also indebted to my wife, Mei Ke, my parents and my grandmother. They all offered me unlimited support and constant encouragement when I needed it.

Finally, I would like to thank all professors that I took classes from in the Department of Mathematical Sciences at the University of Montana.

Table of Contents

List of Figures

List of Tables

Chapter 1

Linear Regression and Nonparametric Correlation Coefficients

1.1 Introduction

Gideon and Hollister (1987) introduced a nonparametric correlation coefficient that was based on the concept of greatest deviations (r_{gd}). This new nonparametric correlation coefficient is defined on ranks and is easy to compute by hand for small to medium sample sizes. In comparing it with existing correlation coefficients, it was found to be superior in a sampling situation that we called "biased outliers" and hence appears to be more resistant to outliers than the Pearson, Spearman, and Kendall correlation coefficients. In a correlational study, the Greatest Deviation Correlation Coefficient r_{gd} was compared with the three other correlation coefficients. The Greatest Deviation Correlation Coefficient was far more "robust" to outliers than other correlation coefficients (Gideon and Hollister, 1987, [20]).

The standard least squares approach in estimating the regression slope b is to minimize a squared error distance function with the centered data. Point estimation of the regression coefficient in linear regression is equivalent to finding the value that makes the residual vector orthogonal to the vector of observations of the explanatory variables. This orthogonality condition is identical to Pearson's correlation coefficient between these vectors equaling zero. When the correlation coefficient is a robust measure of correlation such as the Greatest Deviation Correlation Coefficient r_{gd}, it results in robust point estimates of the regression coefficient (Gideon and Hollister [20]).

1.2 Definition of Greatest Deviation Correlation Coefficient r_{gd}

Let $p = (p_1, p_2, \ldots p_N)$ be a permutation of the first N positive integers. For a bivariate set of data $(x_i, y_i)_{i=1}^N$, let $r(x_i)$ be the rank of x_i among the x data and similarly define $r(y_i)$. We shall assume a continuous distribution for x and y so that with probability 1 the ranks are unique. Now order the x data and let p_i be the rank of the y datum that corresponds to the ith smallest x value.

Let S_N be the symmetric group of degree N. There are $N!$ possible p in S_N. Let the group operation "$\circ$" be the composition of mappings. Thus if both $p = (p_1, p_2, \ldots, p_N)$ and $q = (q_1, q_2, \ldots, q_N)$ are in S_N, then $p \circ q$ has as its ith component $p \circ q(i) = p_{(q_i)}$ $(i = 1,2,\ldots, N)$. For each (X, Y) data set of size N, the permutation p is denoted by $p = p(X, Y)$ and formally defined by $p_{r(x_i)} = p(r(x_i)) = r(y_i)$, where (x_i, y_i) is the ith pair in the data set $(i = 1,2,\ldots, N)$.

There are two permutations in S_N that are of special interest. These are the identity permutation, $e = (1,2,\ldots,N)$, and the reverse permutation $\varepsilon = (N, N-1,\ldots,1)$. Since $\varepsilon(i) = N + 1 - i$, then $\varepsilon \circ p = (N + 1 - p_1, \ldots, N + 1 - p_N)$ and $p \circ \varepsilon = (p_N, \ldots, p_1)$. The composition $\varepsilon \circ p$ results from the reversal of the order of the y values. So, $p(X, -Y) = \varepsilon \circ p(X, Y)$. Similarly, the composition $p \circ \varepsilon$ results from the reversal of the order of the x values, and so $p(-X,Y) = p(X,Y) \circ \varepsilon$. Now we shall explain our definition of the nonparametric correlation coefficient r_{gd}.

In comparing the permutation determined by the sample $p(X,Y)$ with e, we measure the deviation at i (for $i = 1,2,\dots,N$) by the number of $p_1, p_2, \dots, p_i$ that exceed $e_i = i$.

Definition 1.

Let $I(E) = 1$ if E is true and 0 if E is false, and let

$$d_i(\underline{p}) = \sum_{j=1}^{i} I(i < p_j) = \sum_{j=1}^{N} I(r(x_j) \le i < r(y_j))$$

$$d_i(\underline{\varepsilon} \circ \underline{p}) = \sum_{j=1}^{i} I(p_j < N+1-i) = \sum_{j=1}^{i} I(i < N+1-p_j)$$

Definition 2.

$$d(\underline{p}) = \max_i d_i(\underline{p}).$$

$$d(\underline{\varepsilon} \circ \underline{p}) = \max_i d_i(\underline{\varepsilon} \circ \underline{p})$$

Definition 3.

$$r_{gd}(X,Y) = (d(\underline{\varepsilon} \circ \underline{p}) - d(\underline{p}))/[N/2]$$

where $p = p(X,Y)$, the permutation determined by the sample, and [ ] is the greatest integer notation.

Example: For the bivariate set of data $(x_i, y_i)_{i=1}^{16}$, let the rank of x_i and y_i be $(r(x_i), r(y_i))_{i=1}^{16}$ = {(1,14),(2,11),(3,16),(4,2),(5,12),(6,13),(7,7),(8,9),(9,10),(10,3),(11,8), (12,1),(13,5),(14,6),(15,4),(16,5)}. The permutation of the first 16 positive integers is p = (14,11,16,2,12,13,7,9,10,3,8,1,15,6,4,5). Then, by Definition 1, $(d_1(p), d_2(p), \dots, d_{16}(p))$ = (1,2,3,3,4,5,5,6,6,5,4,3,3,2,1,0), $\varepsilon \circ p =$ (3,6,1,15,5,4,10,8,7,14,9,16,2,11,13,12), and $(d_1(\varepsilon \circ p), d_2(\varepsilon \circ p), \dots, d_{16}(\varepsilon \circ p))$ = (1,2,1,2,2,1,2,2,2,2,2,2,3,3,2,1,0).

By Definition 2 and Definition 3, $d(p) = 6$ and $d(\varepsilon \circ p) = 3$, so that $r_{gd} = \frac{(3-6)}{[16/2]} = -\frac{3}{8}$.

1.3 Properties of r_{gd}

The nonparametric correlation coefficients have the following properties (Schweizer and Wolfe, 1981).

Property 1: $r_{gd}(X,Y)$ is well defined.

Property 2: $-1 \le r_{gd}(X,Y) \le +1$

Property 3: $r_{gd}(Y,X) = r_{gd}(X,Y)$

Property 4:

$r_{gd}(-X,Y) = r_{gd}(X,-Y) = -r_{gd}(X,Y)$

Property 5:

$r_{gd}(X,Y) = +1$ with probability 1 if and only if Y is a strictly monotone increasing function of X.

$r_{gd}(X,Y) = -1$ with probability 1 if and only if Y is a strictly monotone decreasing function of X.

Property 6:

If X and Y are independent, then $E[r_{gd}(X,Y)] = 0$

Property 7:

$r_{gd}(f(X), g(Y)) = r_{gd}(X,Y)$ if f and g are strictly monotone increasing functions on the ranges of X and Y, respectively.

The above properties were proved by Hollister (Hollister's Ph.D Dissertation, 1987).

1.4 Simple Linear Regression with Nonparametric Correlation Coefficients

Parameter estimation and hypothesis testing in simple linear regression can be performed using any nonparametric correlation coefficient (Gideon, Li and Rummel [19] and Rummel [40]). In particular, the method given is illustrated using the Greatest Deviation correlation coefficient, r_{gd}, which was developed by Gideon and Hollister, but other correlations such as the modified footrule correlation, r_{mf}, developed by Gideon (1992), or Spearman's and Kendall's correlations could be used in the same manner. There are two reasons for doing nonparametric regression. One reason is to have the conclusions valid under a wider class of bivariate distributions. Another reason is that the estimation process of r_{gd} regression adapts the robustness and resistance of the nonparametric correlation coefficient (Gideon and Rummel [19], [40]).

Let the vector notation x, y denote the random bivariate data (x_i, y_i), $i = 1,2,\ldots,n$. For any correlation R, let $R(x,y)$ be the value of the correlation coefficient on the data. Suppose y, the response variable, and x, the regressor variable, have a continuous bivariate distribution function. Assume the simple linear regression relationship:

$$E(y \mid x) = \alpha + \beta x. \qquad (1.4.1)$$

The standard least squares approach in estimating β is to minimize a squared error distance with the centered data. This is done by differentiating the sum of squared error with respect to β and equating the result to zero. Let b represent the estimate of β; then this is equivalent to choosing b to make the residuals $\underline{y} - \underline{x}b$ orthogonal to the vector $\underline{x}$, when the data are centered, and it is easy to show that this is the same as setting Pearson's correlation of the uncentered vectors $\underline{y} - \underline{x}b$ and $\underline{x}$ to zero (since the vectors $(\underline{y} - \underline{x}b)$

and $\underline{x}$ are orthogonal, $(\underline{y}-\underline{x}b)\perp\underline{x}$, the inner product $(\underline{y}-\underline{x}b,\underline{x})=0$, $r=\dfrac{(\underline{y}-\underline{x}b,\underline{x})}{\|\underline{y}-\underline{x}b\|\cdot\|\underline{x}\|}=0$, i.e., Pearson's correlation coefficient is zero).

Thus, in order to find the estimated slope b, let b be the solution to the equation

$$R(\underline{x},\underline{y}-\underline{x}b)=0 \tag{1.4.2}$$

This estimation method for β is valid for nonparametric correlation coefficients. In the case of nonparametric correlations, when there is an interval of solutions, a standard approach is to take the midpoint.

In solving equation (1.4.2) with either r_{gd} or Pearson's r as b proceeds from minus infinity to plus infinity, $R(\underline{x},\underline{y}-\underline{x}b)$ proceeds monotonically from +1 to -1. For Pearson's r, the decrease is strictly monotonic while for nonparametric R's there are intervals of constant value. The monotonicity allows b to be found after a few iterations using an iterative computer language such as S-Plus or C.

The monotonicity is reviewed in the following section. New estimators of the intercept and the residual scale are also developed from correlation coefficients, and used here, but the development appears in Gideon et al (1992). Because of the monotonicity of $R(\underline{x},\underline{y}-\underline{x}b)$ as a function of b for a given data set, hypothesis testing and confidence intervals are possible. These forms of inference are presented in a general fashion and illustrated for r_{gd}.

The advantage that the Greatest Deviation correlation coefficient method has over other nonparametric regression methods is that the null distribution of the correlation coefficient can be used for testing and for confidence intervals for β.

1.4.1 Background and Monotonicity of $R(\underline{x}, \underline{y} - \underline{x}b)$ as a Function of b

The monotonicity of $R(\underline{x}, \underline{y} - \underline{x}b)$ as a function of b was first discussed by Gideon et. al. in 1993. In order to motivate the nonparametric results, the case $R = r$, Pearson's correlation coefficient is considered first because the techniques are analogous. It can be shown that:

$$r(\underline{x}, \underline{y} - \underline{x}b) = \frac{r(\underline{x}, \underline{y})s_y - bs_x}{(s_y^2 - 2br(x,y)s_x s_y + b^2 s_x^2)^{1/2}}. \qquad (1.4.3)$$

The derivative with respect to b is always nonpositive; hence, $r(\underline{x}, \underline{y} - \underline{x}b)$ being continuous with respect to b is monotonically decreasing. A graph of $r(\underline{x}, \underline{y} - \underline{x}b)$ versus b is shown in figure 1.1 and details of this computation appear in Rummel (1991).

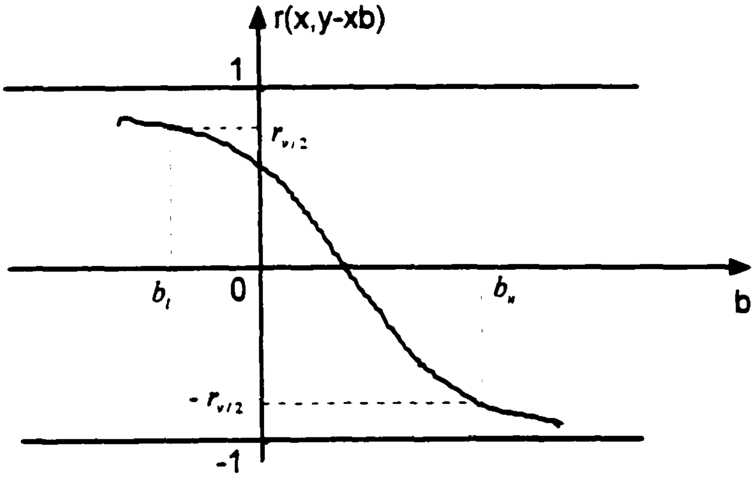


Figure 1.1 Correlation Coefficient as a Decreasing Function

In order to illustrate the calculation of a confidence interval, assume the bivariate normal distribution with parameter set, $(\mu_x, \mu_y, \sigma_x, \sigma_y, \rho)$. Then $E(y \mid x) = \mu_y - \rho(\sigma_y / \sigma_x)(x - \mu_x)$ which defines α and β. $\underline{x}$ and $(\underline{y} - \beta \underline{x})$ are independent random variables, and it follows that $r(\underline{x}, \underline{y} - \beta \underline{x})$ will have the null distribution for these independent random variables. Let $-r_{v/2}$ be the upper $v/2$ critical value for sample size n, and let b_u and b_l be such that

$$r(\underline{x}, \underline{y} - b_l \underline{x}) = r_{v/2} \text{ and}$$

$$r(\underline{x}, \underline{y} - b_u \underline{x}) = -r_{v/2}. \text{ (see Figure 1.1)} \tag{1.4.4}$$

Then by the monotonicity property,

$$b \le b_l \Leftrightarrow r(\underline{x}, \underline{y} - \underline{x}b) \ge r_{v/2} \text{and}$$

$$b \ge b_u \Leftrightarrow r(\underline{x}, \underline{y} - \underline{x}b) \le -r_{v/2}.$$

Thus it follows that (b_l, b_u) is a $1 - v$ confidence interval because $P(b_l \le b \le b_u) = 1 - v$.

It was shown in Rummel (1991) that with

$$h = r_{v/2} \left(\frac{1 - r^2(\underline{x}, \underline{y})}{1 - r_{v/2}^2} \right)^{1/2} \tag{1.4.5}$$

$$b_l = (r(\underline{x}, \underline{y}) - h) \cdot s_y / s_x \text{ and } b_u = (r(\underline{x}, \underline{y}) + h) \cdot s_y / s_x. \tag{1.4.6}$$

In addition, this r-based confidence interval for b is exactly the same as the least squares confidence interval using the appropriate t-distribution. The above concepts and procedures can be adapted to encompass any correlation coefficients. Nonparametric correlation coefficients are decreasing step functions of b, but the geometrical ideas remain the same. We will illustrate it in Section 1.4.3.

1.4.2 Definitions, Tied Values, and Review

Let I be a 0,1 indicator function obtaining a 1 if the event is true. For a bivariate data set $(\underline{x},\underline{y})$, with no tied values, order the x data and let $t'=(t_1,\dots,t_n)$ be the associated rank vector for $\underline{y}$. Then for a nonparametric correlation coefficient such as r_{gd}, $r_{gd}(\underline{x},\underline{y})=r_{gd}(\underline{e},\underline{t})$ where $e'=(1,2,\dots,n)$ (Gideon, 1992). For convenience and without loss of generality consider from here on the $(\underline{x},\underline{y})$ data ordered by the $\underline{x}$ data. If tied values exist in the x and/or y data, create two non-tied $\underline{t}$ vectors, one that favors positive correlation $\underline{t}^{+}$, and a second that favors negative correlation $\underline{t}^{-}$. The vector $\underline{t}^{+}$ is formed by choosing ranks for the y data within the restriction of ties to have the higher ranks as close as possible to the nth position; $\underline{t}^{-}$ would position the higher ranks as close as possible to the first position. An example appears in Gideon and Hollister (1987). Then for tied data $r_{gd}(\underline{x},\underline{y})=[r_{gd}(\underline{e},\underline{t}^{+})+r_{gd}(\underline{e},\underline{t}^{-})]/2$. This is called the max-min procedure.

Example: Let the rank of $(\underline{x},\underline{y})$ be

x	1	2	3.5	3.5	5	6.5	6.5	8	9	10	11.5	11.5
y	1	2.5	8	7	4.5	6	2.5	10	4.5	9	12	11

We list these two permutations:

x	1	2	3	4	5	6	7	8	9	10	11	12
$y(+)$	1	2	7	8	4	3	6	10	5	9	11	12
$y(-)$	1	3	8	7	5	6	2	10	4	9	12	11

In both cases $r_{gd}^{+} = 0.5$ and $r_{gd}^{-} = 0.5$, so $r_{gd} = (r_{gd}^{+} + r_{gd}^{-})/2 = 0.5$.

The definitions of r_{gd} and r_{mf} are now given for non-tied value data. These definitions appear in Gideon (1992) along with some comparisons to Kendall's and Spearman's correlations.

Let
$$d_i^{-}(\underline{t}) = \sum_{j=1}^{i} I(t_j < n+1-i),$$

$$d_i^{+}(\underline{t}) = \sum_{j=1}^{i} I(i < t_j) \text{ for } i=1,2,\ldots,\text{n}. \qquad (1.4.7)$$

$$r_{gd}(\underline{e},\underline{t}) = (\max_i d_i^{-}(\underline{t}) - \max_i d_i^{+}(\underline{t})) / \left[\frac{n}{2}\right]$$

and
$$r_{mf}(\underline{e},\underline{t}) = (\sum_{i=1}^{n} d_i^{-}(\underline{t}) - \sum_{i=1}^{n} d_i^{+}(\underline{t})) / \left[\frac{n^2}{4}\right] \qquad (1.4.8)$$

For regression, $\underline{t}$ will be the function of b, and the vector $\underline{t}(b)$ has for its ith component the rank of $y_i - bx_i$ among the set of n residuals. If ties exist among the x's or the residuals, then $\underline{t}^{+}(b)$ and $\underline{t}^{-}(b)$ need to be formed. The problem is to determine b so that for r_{gd}, say,

$$r_{gd} = (\underline{x}, \underline{y} - \underline{x}b) = (r_{gd}^{+} + r_{gd}^{-})/2 = 0 \qquad (1.4.9)$$

For $i < j, x_i < x_j$ and letting $b^* = (y_j - y_i)/(x_j - x_i)$ then $y_j - x_j b^* = y_i - x_i b^*$. For small $\varepsilon > 0$, as b changes from $b^* - \varepsilon$ to $b^* + \varepsilon$, the ranks of these two residuals are interchanged with the higher rank moving towards the beginning of the residual vector (beginning rank is $i = 1$). Nonparametric correlation coefficients will never increase and possibly decrease at such a transition. With this max-min method, at b^*,

$$r_{gd} = (\underline{x}, \underline{y} - \underline{x}b^*) = (r_{gd}^{-} + r_{gd}^{+})/2 = (\inf_{b<b^*} r_{gd}(\underline{x}, \underline{y} - \underline{x}b) + \sup_{b>b^*} r_{gd}(\underline{x}, \underline{y} - \underline{x}b))/2 . \qquad (1.4.10)$$

1.4.3 Properties of the Estimators for β

Let

$$b^* = \sup\{b : r_{gd}(\underline{x}, \underline{y} - \underline{x}b) > 0\}$$

$$b^{**} = \inf\{b : r_{gd}(\underline{x}, \underline{y} - \underline{x}b) < 0\}$$

$$b = (b^* + b^{**})/2 \qquad (1.4.11)$$

and define b to be the estimate of β for data $(\underline{x}, \underline{y})$. Let $b(\underline{x}, \underline{y})$ denote this estimate.

It follows from the properties of correlations that for constants d_1, d_2, c_1, c_2 with $d_2 \neq 0$,

$$b(d_1\underline{1} + d_2\underline{x},\ c_1\underline{1} + c_2\underline{y}) = (c_2 / d_2) b(\underline{x}, \underline{y}), \qquad (1.4.12)$$

$$b(\underline{x},\ \underline{y} + a\underline{x}) = a + b(\underline{x}, \underline{y}). \qquad (1.4.13)$$

<u>**Theorem**</u>: The distribution of b defined by (1.4.11) is symmetric about the parameter β in a simple linear regression model.

Proof. $r_{gd}(\underline{x}, \underline{y} - \underline{x}\beta)$ is symmetric about zero and without loss of generality by the linearity property (1.4.13) above, let $\beta = 0$. Then $r_{gd}(\underline{x}, -\underline{y}) = -r_{gd}(\underline{x}, \underline{y})$, a standard property of correlation coefficients, and the null distribution is symmetric about zero (if

$\beta = 0$, x and y are uncorrelated). Thus, $r_{gd}(\underline{x},\underline{y}) \overset{d}{=} r_{gd}(\underline{x},-\underline{y})$ (equal in distribution). This fact and the earlier statement that $b(\underline{x},\underline{y}) = -b(\underline{x},-\underline{y})$ are enough to show that $b(\underline{x},\underline{y}) \overset{d}{=} b(\underline{x},-\underline{y})$, i.e., b is symmetric about zero. This shows that b is unbiased for β assuming $E(b)$ exists.

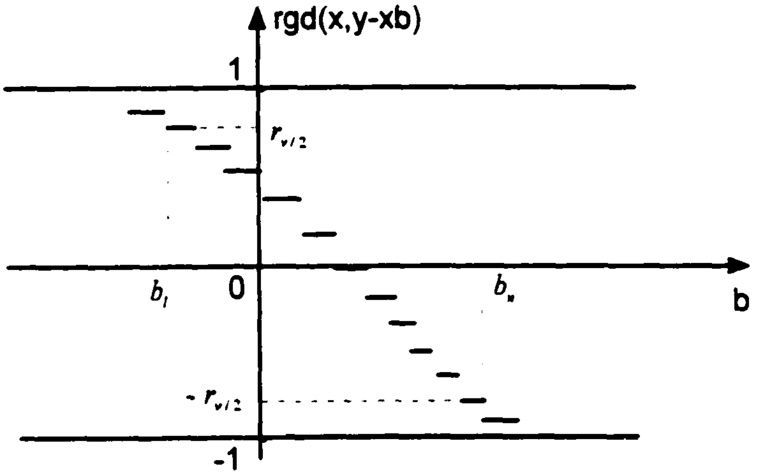


Figure 1.2 The Greatest Deviation Correlation Coefficient as a Decreasing Step Function of b

The Greatest Deviation correlation coefficient $r_{gd}(x, y - \underline{x}b)$ is a decreasing step function of b (Figure 1.2). Confidence intervals for β are illustrated in Figure 1.2 with r_{gd} and the symmetric null distribution of $r_{gd}(\underline{x},\underline{y} - \underline{x}\beta)$. Let $r_{v/2}$ be such that

$$P\{-r_{v/2} \le r_{gd}(\underline{x},\underline{y} - \underline{x}b) \le r_{v/2}\} = 1 - v \text{ (see Figure 1.2)}$$

Then define b_u and b_l in the equations

$$b_u = \sup\{b^* : r_{gd}(\underline{x}, \underline{y} - \underline{x}b^*) \geq -r_{v/2}\}$$

$$b_l = \inf\{b^* : r_{gd}(\underline{x}, \underline{y} - \underline{x}b^*) \leq r_{v/2}\} \tag{1.4.14}$$

Then $P\{b_l \leq b \leq b_u\} = 1 - v$ and (b_l, b_u) is the confidence interval. From Gideon et al. ([16]) we have $\sqrt{n}r_{gd}(\underline{x}, \underline{y} - \underline{x}\beta) \xrightarrow{d} N(0,1)$ and from Gideon and Li (1992) we have $\sqrt{n}r_{mf}(\underline{x}, \underline{y} - \underline{x}\beta) \xrightarrow{d} N(0,2/3)$. Then for large sample sizes, asymptotic $1 - v$ confidence intervals can be obtained by solving for b_l and b_u in

$$r_{gd}(\underline{x}, \underline{y} - \underline{x}b_l) = Z_{v/2} / \sqrt{n},$$

$$r_{gd}(\underline{x}, \underline{y} - \underline{x}b_u) = -Z_{v/2} / \sqrt{n} \tag{1.4.15}$$

where $Z_{v/2}$ is the upper $v/2$ percentile for a $N(0,1)$ random variable.

For b_l and b_u using r_{mf} solve

$$r_{mf}(\underline{x}, \underline{y} - \underline{x}b_l) = Z_{v/2} / \sqrt{3n/2}$$

$$r_{mf}(\underline{x}, \underline{y} - \underline{x}b_u) = -Z_{v/2} / \sqrt{3n/2} \tag{1.4.16}$$

It has been found that a bisection algorithm works well to find the estimate b for r_{gd} and r_{mf} in equation (1.4.2). We can find b^* and b^{**} in equation (1.4.11) using r_{gd} and r_{mf} as defined in (1.4.8) and using the max-min method for tied values. A confidence interval is obtained for b by again solving an equation like (1.4.2) with the same numerical algorithm except the right-hand side gets replaced by upper and lower critical values of the null distribution of the appropriate correlation as explained above. The null distribution of r_{gd} appears in Gideon and Hollister (1987).

1.4.4 Intercept and Residual Scale Estimates

For any correlation coefficient and in particular for the Greatest Deviation correlation coefficient r_{gd}, the simple linear regression equation is:

$$r_{gd}(\underline{x}, \underline{y} - b_1 \underline{x}) = 0. \tag{1.4.17}$$

Thus, the regressor variables are uncorrelated with the regression residuals. The intercept of the regression is estimated by taking the median of these residuals:

$$b_0 = median(\underline{y} - b_1 \underline{x}) \tag{1.4.18}$$

In order to estimate σ, solve the following equation for s ([16]):

$$r_{gd}(\underline{k}, res^0 - s * \underline{k}) = 0 \tag{1.4.19}$$

where res^0 is the vector of ordered residuals, $\underline{k}$ is the standard Gaussian order statistics.

1.5 Multiple Linear Regression with Nonparametric Correlation Coefficients

The estimations are not strongly affected by the outliers is called the robustness. Estimation of the parameters of a general linear model was proposed by using the nonparametric correlation coefficient, r_{gd} (Gideon, Rummel and Li [19], Rummel [40]). In these unpublished research papers, hypothesis and subhypothesis tests were also introduced by using this correlation as a multiple correlation coefficient. The efficiency of the estimation and the power of the new test procedure were studied using Monte Carlo simulations. Simulation studies showed that these procedures are more robust and efficient than the classical least square procedures when the underlying error distribution is not normal, and they compare well with existing robust methods. In general, any nonparametric correlation could be utilized in the same manner.

The least squares estimation procedure and the classical F test procedure for the multiple linear regression model can be restated by using Pearson's correlation coefficient r. We substitute r_{gd} for r in the determining normal equations and study the resulting effects on the estimation of the parameters. One reason for this is that it is known that the least squares estimation and classical F test procedures are not robust to different error assumptions. The Greatest Deviation Correlation Coefficient is resistant to outliers and we used this correlation coefficient in the estimation of parameters in the simple linear regression model. We found that the robustness of r_{gd} to outliers and nonnormality as a correlation coefficient induces robustness in the estimated parameters. We extended the estimation method from the simple linear model to the multiple linear model.

The general linear model can be written in matrix form as:

$$\underline{y} = X\underline{\beta} + \underline{\varepsilon} \tag{1.5.1}$$

where $\underline{y}$ is an $n \times 1$ vector of independent observations, X is an $n \times (p+1)$ matrix of known constants, $\underline{\beta}$ is a $(p+1) \times 1$ vector of unknown regression parameters and $E(\underline{\varepsilon}) = 0,\ E(\varepsilon\varepsilon') = \sigma^2 I_n, \sigma > 0$ where I_n is the identity matrix of order n.

Section 1.5.1 introduces a robust estimation method for the β parameters. Section 1.5.2 shows how to estimate the parameters α and σ ([19], [40]).

1.5.1 Estimation of Parameters

The least squares estimator of $\underline{\beta}$, say $\tilde{\underline{\beta}}$ can be obtained by solving the following equations:

$$
\begin{aligned}
&r(\underline{x}_1, \underline{y} - X\tilde{\underline{\beta}}) = 0,\\
&r(\underline{x}_2, \underline{y} - X\tilde{\underline{\beta}}) = 0,\\
&\cdot\\
&\cdot\\
&\cdot\\
&r(\underline{x}_p, \underline{y} - X\tilde{\underline{\beta}}) = 0,
\end{aligned}
\qquad (1.5.2)
$$

where r stands for Pearson's correlation coefficient and $X = (\underline{x}_1, \underline{x}_2, \ldots, \underline{x}_p)$, where $\underline{x}_1, \underline{x}_2, \ldots, \underline{x}_p$ are column vectors representing the p predictors.

It is known that Pearson's correlation coefficient r is not robust to outliers and nonnormality; therefore, the least squares estimates of the β's are not robust to outliers and nonnormality. The r_{gd} estimates of β's, denoted as $\tilde{\beta}_{r_{gd}}$, are defined by replacing Pearson's correlation coefficient r by the Greatest Deviation Correlation Coefficient r_{gd} in equations (1.5.2); i.e., by solving the following equations:

$$
\begin{aligned}
&r_{gd}(\underline{x}_1, \underline{y} - X\underline{\tilde{\beta}_{r_{gd}}}) = 0,\\
&r_{gd}(\underline{x}_2, \underline{y} - X\underline{\tilde{\beta}_{r_{gd}}}) = 0,\\
&\cdot\\
&\cdot\\
&\cdot\\
&r_{gd}(\underline{x}_p, \underline{y} - X\underline{\tilde{\beta}_{r_{gd}}}) = 0,
\end{aligned}
\qquad (1.5.3)
$$

where $\underline{\tilde{\beta}_{r_{gd}}} = (\tilde{\beta}_{r_{gd1}}, \tilde{\beta}_{r_{gd2}}, ..., \tilde{\beta}_{r_{gdp}})$. The equations in (1.5.3) will be called the r_{gd} normal equations.

There is no explicit solution, so a numerical algorithm is needed to solve these equations. Let w_1 and w_2 be $n \times 1$ data vectors from a continuous bivariate random variable. Because $r_{gd}(w_1, w_2 - w_1 b)$ is a non-increasing function of b for any fixed w_1 and w_2, $r_{gd}(w_1, w_2 - w_1 b) \to 1$ when b is a large negative number, and $r_{gd}(w_1, w_2 - w_1 b) \to -1$ when b is a large positive number (see Rummel, 1991), equations (1.5.3) have a solution. C and Splus computer programs which use a bisection numerical method to obtain a solution for b in this simple linear regression model have ben written.

For an intermediate set of possible solutions of (1.5.3) $\tilde{\beta}_i, i = 1,2,...., p$. let $y_i = y - \sum_{j \neq i} \tilde{\beta}_j x_j$. The ith equation in (1.5.3) is

$$r_{gd}(x_i, y_i - x_i \tilde{\beta}_i) = 0. \tag{1.5.4}$$

Now, $r_{gd}(x_i, y_i - x_i \tilde{\beta}_i)$ is a decreasing step function in $\tilde{\beta}_i$ and a bisection method is used to solve for $\tilde{\beta}_i$ in equation (1.5.4) where the bisection method depends on the possible jump points of the step function, $(y_{i,k} - y_{i,j})/(x_{i,k} - x_{i,j})$ where $y_i' = (y_{i,1}, \ldots, y_{i,n})$ and x_i' = $(x_{i,1}, \ldots, x_{i,n})$ and $(x_{i,k} - x_{i,j}) \neq 0$. Once $\hat{\beta}_i$ is found to satisfy Equation (1.5.4), the process is repeated at $i+1$. Thus, given a set of initial values ($\beta_1^0, \ldots, \beta_p^0$), the equations are solved sequentially for $i = 1,2,..., p$ to obtain a new set of values ($\beta_1^1, \ldots, \beta_p^1$). This *Gauss-Siedel* method has converged under a wide set of simulations and examples. It has difficulty when p is too near n. When $p = 1$, a unique solution can be defined by averaging the infinum and supremum of the solution set. Since r_{gd} is discrete-valued, the solution set for the $\hat{\beta}_i$'s when $p > 1$ is a region in p space, and currently, the solution is defined to be the first set of values that satisfies equation (1.5.3).

1.5.2 Estimate of Error and Intercept

For any correlation coefficient and in particular for the Greatest Deviation correlation coefficient r_{gd}, the multiple linear regression equations are:

$$r_{gd}(\underline{x_i}, \underline{y} - b_1\underline{x_1} - b_2\underline{x_2} - \ldots - b_p\underline{x_p}) = 0, i = 1,2,..,p. \qquad (1.5.5)$$

Thus, the regressor variables are uncorrelated with the regression residuals. The intercept of the regression is obtained by taking the median of these residuals:

$$b_0 = median(\underline{y} - b_1\underline{x_1} - b_2\underline{x_2} - \ldots - b_p\underline{x_p}) \qquad (1.5.6)$$

where n is the sample size, $\underline{x_1} = \begin{pmatrix} x_{11} \\ x_{12} \\ \cdot \\ \cdot \\ \cdot \\ x_{1n} \end{pmatrix}$, ..., $\underline{x_p} = \begin{pmatrix} x_{p1} \\ x_{p2} \\ \cdot \\ \cdot \\ \cdot \\ x_{pn} \end{pmatrix}$, and for simple linear regression, $p = 1$.

In order to estimate σ, solve the following equation for s:

$$r_{gd}(\underline{k}, res^0 - s * \underline{k}) = 0 \qquad (1.5.7)$$

where res^0 is the vector of the ordered residuals, $\underline{k}$ is the vector of standard Gaussian order statistics. For $z_{(i)}$, the ith order statistic from a $N(0,1)$ random sample, $k_i = E(z_{(i)})$, $i = 1,2,\ldots,n$.

Chapter 2

Generalized Linear Models and Estimation

2.1 Introduction

For several decades linear models of the form

$$\underline{y} = X\underline{\beta} + \underline{\varepsilon} \tag{2.1.1}$$

with the assumption that the elements of ε are $NID(0, \sigma^2)$ have formed the basis of most analyses of continuous data. Recent advances in statistical theory and computer software allow us to use methods analogous to those developed for linear models in the following situations:

- the response variables have distributions other than the normal distribution - they may even be categorical rather than continuous;
- the relationship between the response and explanatory variables need not be of the simple linear form in (2.1.1).

One of these advances has been the recognition that many of the 'nice' properties of the normal distributions are shared by a wider class of distribution called the exponential family of distributions. This chapter introduces the exponential family of distributions and defines generalized linear models. A second advance is the extension of the numerical methods for estimating parameters, from linear combinations like $X\beta$ in (2.1.1) to differentiable functions of linear combinations such as $g(X\beta)$.

In Chapter 1 we discussed linear regression with nonparametric correlation coefficients, such as r_{gd}. The parameters α and β can be estimated by least squares or the r_{gd} method. Gideon and Hollister (1987) introduced the Greatest Deviation Correlation Coefficient, r_{gd}. Gideon, Rummel, and Li (1993) used this correlation coefficient in the estimation of parameters in the simple linear regression and multiple linear regression models (Gideon, Li and Rummel [19] and Gideon, Rummel and Li [40]).

Let the vector notation $\underline{x}, \underline{y}$ denote the random bivariate data (x_i, y_i), $i = 1,2,\cdots n$. For Pearson's correlation coefficient r or the Greatest Deviation correlation coefficient r_{gd}, let $r(\underline{x},\underline{y})$ or $r_{gd}(\underline{x},\underline{y})$ be the value of the correlation coefficient on the data. Assume the response variable $\underline{y}$ and the independent variable $\underline{x}$, have a continuous bivariate distribution with simple linear regression relationship as in (2.1.1). The standard least squares approach in estimating β is to minimize a squared error distance function with the centered data. This is done by differentiating it with respect to β and equating the result to zero.

Letting b represent the estimate of β, then this is equivalent to choosing b to make the residuals $\underline{y} - \underline{x}b$ orthogonal to the vector $\underline{x}$, when the data are centered. This is equivalent to setting the Pearson's correlation coefficient of the uncentered vector $\underline{y} - \underline{x}b$ and $\underline{x}$ to zero (see Chapter 1):

$$r(\underline{x}, \underline{y} - \underline{x}b) = 0. \qquad (2.1.2)$$

In solving equation (2.1.2) with either Pearson's r or the Greatest Deviation correlation coefficient r_{gd} as b proceeds from $-\infty \; to + \infty$, $r(\underline{x}, \underline{y} - \underline{x}b)$ or $r_{gd}(\underline{x}, \underline{y} - \underline{x}b)$ proceeds monotonically from +1 to -1 (see Figure 1.1 and Figure 1.2 in Chapter 1).

This monotonicity allows us to find of b after a few iterations using an iterative computer program language such as S-Plus or C for the computation of r_{gd}, which can then be used to find b. Least squares estimation for generalized linear models and nonlinear models can be modified in a natural way to accommodate using the Greatest Deviation correlation coefficient r_{gd}. This chapter extends the method developed for simple linear models to generalized linear models by substituting r_{gd} for r in the determining normal equations.

2.2 Generalized Linear Models

2.2.1 Exponential Family:

Exponential Family of Distributions:

A distribution belongs to the one-parameter exponential family if it can be written in the form:

$$f(y,\theta) = s(y)t(\theta)e^{a(y)b(\theta)} \qquad (2.2.1)$$

where a, b, s, t are known functions, and θ is an unknown parameter.

(2.2.1) can be rewritten in the form:

$$f(y,\theta) = \exp[a(y)b(\theta) + c(\theta) + d(y)] \qquad (2.2.2)$$

where $s(y) = \exp[d(y)]$, and $t(\theta) = \exp[c(\theta)]$.

If $a(y) = y$, the distribution in (2.2.2) is said to be in canonical form and $b(\theta)$ is called the natural parameter of the distribution. If there are other parameters in addition to θ, they are regarded as nuisance parameters forming parts of the function $a, b. c$ and d. they are treated as though they are known.

Many well-known distributions belong to the exponential family. For example, the Poisson, Normal and binomial distributions can be written in the canonical form.

A. Poisson Distribution

$$f(y,\lambda) = \frac{\lambda^y e^{-\lambda}}{y!}, \quad y = 0,1,2,.... \tag{2.2.3}$$

This can be rewritten as:

$$f(y,\lambda) = \exp[y \log\lambda - \lambda - \log y!]$$

which is in the canonical form with $\log\lambda$ as the natural parameter.

B. Normal Distribution, $Y \sim N(\mu, \sigma^2)$

$$f(y,\mu) = \frac{1}{(2\pi\sigma^2)^{1/2}} \exp[-\frac{1}{2\sigma^2}(y-\mu)^2], \quad -\infty < y < \infty \tag{2.2.4}$$

The canonical form is

$$f(y,\mu) = \exp[-\frac{y^2}{2\sigma^2} + \frac{y\mu}{\sigma^2} - \frac{\mu^2}{2\sigma^2} - \frac{1}{2}\log(2\pi\sigma^2)]$$

with the natural parameter μ/σ^2.

C. Binomial Distribution, $Y \sim b(n,\pi)$

$$f(y,\pi) = \binom{n}{y}\pi^y(1-\pi)^{(n-y)}, \quad y = 0,1,2,...,n. \tag{2.2.5}$$

The canonical form is

$$f(y,\pi)=\exp\left[y\log\pi-y\log(1-\pi)+n\log(1-\pi)+\log\binom{n}{y}]\right]$$

with the natural parameter $\log\frac{\pi}{1-\pi}$, the log odds ratio.

These results are summarized in Table 2.1.

Table 2.1 Poisson, Normal and binomial distributions as members of the exponential family

Distribution	Natural parameters	c	d
Poisson	$\log\lambda$	$-\lambda$	$-\log y!$
Normal	μ/σ^2	$-\frac{1}{2}\mu^2/\sigma^2-\frac{1}{2}\log(2\pi\sigma^2)$	$-\frac{1}{2}y^2/\sigma$
Binomial	$\log(\frac{\pi}{1-\pi})$	$n\log(1-\pi)$	$\log\binom{n}{y}$

In order to get the normal equations for generalized linear models in Section 2.3.3, we need to find the expressions of $E[a(y)]$ and $Var[a(y)]$.

Consider a continuous random variable Y with the probability density function $f(y;\theta)$ depending on a single parameter θ. The log-likelihood function is the logarithm of $f(y,\theta)$:

$$l(\theta,y)=\log f(y,\theta)$$

The derivative $U = \frac{dl}{d\theta}$ is called the score. We can get the moments of U using the identity (Cox & Hinkley, [7]):

$$\frac{d\log f(y,\theta)}{d\theta} = \frac{1}{f(y,\theta)}\frac{df(y,\theta)}{d\theta}. \qquad (2.2.6)$$

If we take expectations of (2.2.6) we obtain

$$E(U) = \int \frac{d\log f(y,\theta)}{d\theta} f(y,\theta)dy = \int \frac{df(y,\theta)}{d\theta}dy$$

$$\frac{d}{d\theta}\int f(y,\theta)dy = \frac{d}{d\theta}1 = 0$$

This interchange of the integral and derivative works for any exponential family distribution due to the Lebesgue Dominated Convergence Theorem.

Hence, $E(U) = 0$ (Cox & Hinkley [7]) (2.2.7)

Also, if we take expectations of (2.2.6) and differentiate it with respect to θ. the order of these operations for any exponential family distribution can be interchanged due to the Lebesgue Dominated Convergence Theorem, then

$$\frac{d}{d\theta}\int \frac{d\log f(y,\theta)}{d\theta} f(y,\theta)dy = \frac{d^2}{d\theta^2}\int f(y,\theta)dy = 0$$

$$\int \frac{d^2\log f(y,\theta)}{d\theta^2} f(y,\theta)dy + \int [\frac{d\log f(y,\theta)}{d\theta}]^2 f(y,\theta)dy = 0$$

Therefore, $E[-\frac{d^2\log f(y,\theta)}{d\theta^2}] = E\{[\frac{d\log f(y,\theta)}{d\theta}]^2\}$

or $E[-U'] = E[U^2]$. (2.2.8)

Since $E(U) = 0$, the variance of U, which is called the information, is

$$Var(U) = E(U^2) = E(-U') \tag{2.2.9}$$

More generally, consider independent random variables $Y_1, Y_2, ..., Y_N$ whose probability distributions depend on parameters $\theta_1, \theta_2, ..., \theta_p$ where $p \le N$. Let $l_i(\theta; y_i)$ denote the log-likelihood function of $\theta = [\theta_1, ..., \theta_p]^T$ for Y_i. Then due to the independence, the log-likelihood function for $Y_1, Y_2, ..., Y_N$ is

$$l(\theta; y) = \sum_{i=1}^{N} l_i(\theta; y_i)$$

where $y = [y_1, ..., y_N]^T$. The total score with respect to θ_j is defined as

$$U_j = \frac{\partial l(\theta; y)}{\partial \theta_j} = \sum_{i=1}^{N} \frac{\partial l_i(\theta; y_i)}{\partial \theta_j}, \text{ with}$$

$$E\left[\frac{\partial l_i(\theta; y_i)}{\partial \theta_j}\right] = 0,$$

and so $\qquad E(U_j) = 0 \qquad$ for $j = 1, 2, ..., p$. $\qquad$ (2.2.10)

The information matrix, $J_{p \times p}$, is defined to be the variance-covariance matrix of the U_j's, where $J = E(U\,U^T)$, U is the vector of scores $U_1, U_2, ..., U_p$, i.e., $U = [U_1 ..., U_p]^T$, so J has elements

$$J_{jk} = E[U_j U_k] = E[\frac{\partial l}{\partial \theta_j} \frac{\partial l}{\partial \theta_k}] = E[-\frac{\partial^2 l_i}{\partial \theta_j \partial \theta_k}]. \tag{2.2.11}$$

To find the expected value and variance of $a(Y)$ we use the above results. From (2.2.2),

$$l = \log f = a(y)b(\theta) + c(\theta) + d(y)$$

so that $$U = \frac{dl}{d\theta} = a(y)b'(\theta) + c'(\theta),$$

and $$U' = \frac{d^2 l}{d\theta^2} = a(y)b''(\theta) + c''(\theta).$$

Thus $$E(U) = b'(\theta)E[a(Y)] + c'(\theta)$$

Since $E(U) = 0$,

$$E[a(Y)] = -c'(\theta)/b'(\theta). \qquad (2.2.12)$$

Also $$Var(U) = E(U^2) = [b'(\theta)]^2 \operatorname{var}[a(Y)]$$

and $$E(-U') = -b''(\theta)E[a(Y)] - c''(\theta)$$

$\Rightarrow$ $$Var[a(y)] = \{-b''(\theta)E[a(Y)] - c''(\theta)\}/[b'(\theta)]^2$$

$$= [b''(\theta)c'(\theta) - c''(\theta)b'(\theta)]/[b'(\theta)]^3. \qquad (2.2.13)$$

2.2.2 Generalized Linear Model

Let $Y_1, \cdots, Y_n$ be independent random variables, each with a distribution from the exponential family with the following properties:

- The distribution of each Y_i is of canonical form and depends on a single parameter θ_i, thus

$$f(y_i; \theta_i) = \exp[y_i b_i(\theta_i) + c_i(\theta_i) + d_i(y_i)]. \qquad (2.2.14)$$

- The distribution of all the Y_i's is of the same form so that the subscripts on b, c and d are not needed. Thus the joint probability density function of $Y_1, \cdots, Y_n$ is

$$f(y_1, \cdots, y_n; \theta_1, \cdots, \theta_n) = \exp[\sum_{i=1}^{n} y_i b(\theta_i) + \sum_{i=1}^{n} c(\theta_i) + \sum_{i=1}^{n} d(y_i)]. \qquad (2.2.15)$$

For a generalized linear model, we consider a smaller set of parameters:

$\beta_1, \beta_2, \cdots, \beta_p$ (where $p < n$) such that a linear combination of the β 's is equal to some function of the expected value μ_i of Y_i, i.e.,

$$g(\mu_i) = x_i^T \underline{\beta} \tag{2.2.16}$$

where g is a monotone, differentiable function called the link function;

$\underline{x_i}$ is a $p \times 1$ vector of explanatory variables corresponding to y_i; and

$\underline{\beta}$ is a $p \times 1$ the vector of parameters $= [\beta_1, \ldots, \beta_p]^T$

Thus, a generalized linear model has three components:

- response variables $Y_1, \cdots, Y_n$ which are assumed to share the same distribution from the exponential family;
- a set of parameters β and explanatory matrix: $X = [x_1^T, \ldots, x_n^T]^T$
- a monotone link function g such that

$$g(\mu_i) = x_i^T \underline{\beta} \quad \text{where } \mu_i = E(Y_i).$$

Such models form the core of this chapter.

2.3 Estimation in Generalized Linear Models

Two of the commonly used approaches to the statistical estimation of parameters are the method of maximum likelihood and method of least squares. This chapter begins by reviewing the principles of each of these methods and some properties of the estimators. In Section 2.3.3 we will discuss parameters estimation using the Greatest Deviation correlation coefficient r_{gd}.

2.3.1 Method of Maximum Likelihood

Let $Y_1,...,Y_N$ be N random variables with the joint probability density function $f(y_1,...y_N;\theta_1,...\theta_p)$ which depends on parameters $\theta_1,...\theta_p$. For brevity we denote $[y_1,...,y_N]^T$ by $\underline{y}$ and $[\theta_1,...,\theta_p]^T$ by $\underline{\theta}$.

Let Ω denote the parameter space, i.e. all possible values of the parameter vector $\underline{\theta}$. The maximum likelihood estimator of $\underline{\theta}$ is defined as the vector $\hat{\underline{\theta}}$ such that

$$L(\hat{\underline{\theta}};\underline{y}) \geq L(\underline{\theta};\underline{y}) \qquad \text{for all } \underline{\theta} \in \Omega.$$

Equivalently, if $l(\theta;y) = \log L(\theta;y)$ is the log-likelihood estimator, then $\hat{\theta}$ is the maximum likelihood estimator if

$$l(\hat{\underline{\theta}};\underline{y}) \geq l(\underline{\theta};\underline{y}) \qquad \text{for all } \underline{\theta} \in \Omega.$$

The most convenient way to obtain the maximum likelihood estimator is to examine all the local maxima of $l(\underline{\theta};\underline{y})$. These maxima are

(i) the solution of $\dfrac{\partial l(\underline{\theta};\underline{y})}{\partial \theta_j} = 0, \qquad j = 1,..., p,$

such that $\underline{\theta}$ belongs to Ω and the matrix of second derivatives

$\dfrac{\partial^2 l(\underline{\theta};\underline{y})}{\partial \theta_j \partial \theta_k}$ is negative definite; and

(ii) any values of $\underline{\theta}$ at the edges of the parameter space Ω which correspond to maxima of $l(\underline{\theta};\underline{y})$.

The value $\hat{\underline{\theta}}$ giving the largest of the local maxima is the maximum likelihood estimator. For models considered in this chapter there is usually a unique maximum given by $\partial L / \partial \underline{\theta} = 0$.

An important property of maximum likelihood estimators is that if $\psi(\underline{\theta})$ is any function of the parameters θ, then the maximum likelihood estimator of ψ is

$$\hat{\psi} = \psi(\hat{\underline{\theta}}).$$

This invariance property follows from the definition of $\hat{\underline{\theta}}$.

2.3.2 Method of Least Squares

Let $Y_1,...Y_N$ be random variables with expected values

$$E(Y_i) = \mu_i = \mu_i(\underline{\beta}), \quad i = 1,...,N,$$

where $\underline{\beta} = [\beta_1,...,\beta_p]^T$ $(p < N)$ are the parameters to be estimated. Consider the model:

$$Y_i = \mu_i + e_i, \quad i = 1,...,N,$$

where e_i is the ith random error.

The method of least squares consists of finding estimators $\hat{\underline{\beta}}$ which minimize the sum of squares of the error terms

$$S = \sum_{i=1}^{N} e_i^2 = \sum_{i=1}^{N} [Y_i - \mu_i(\underline{\beta})]^2.$$

In matrix notation this is

$$S = (\underline{y} - \underline{\mu})^T (\underline{y} - \underline{\mu})$$

where $\underline{y} = [Y_1,...,Y_N]^T$ and $\underline{\mu} = [\mu_1,...,\mu_N]^T$.

Generally the estimator $\hat{\underline{\beta}}$ is obtained by differentiating S with respect to each element β_j of $\underline{\beta}$ and solving the simultaneous equations

$$\frac{\partial S}{\partial \beta_j} = 0, \qquad j = 1,...p.$$

It is necessary to check that the solutions correspond to minima (i.e. the matrix of second derivatives is positive definite) and to identify the global minimum from among these solutions and any local minima at the boundary of the parameter space.

2.3.3 Estimation using Pearson's Correlation Coefficient and the Greatest Deviation Correlation Coefficient r_{gd}

We wish to obtain maximum likelihood estimators of the parameters in $\underline{\beta}$ for the generalized linear models defined in Section 2.2.2. The log-likelihood function for independent responses $Y_1,...,Y_N$ is

$$l(\underline{\theta},\underline{y}) = \sum y_i b(\theta_i) + \sum c(\theta_i) + \sum d(y_i)$$

where $$E(Y_i) = \mu_i = -{c'(\theta_i)}\big/{b'(\theta_i)} \tag{2.3.1}$$

and $$g(\mu_i) = x_i^T \underline{\beta} = \eta_i, \tag{2.3.2}$$

where g is monotone and differentiable.

From (2.2.13),

$$Var(Y_i) = [b''(\theta_i)c'(\theta_i) - c''(\theta_i)b'(\theta_i)]/[b'(\theta_i)]^3 \tag{2.3.3}$$

The score with respect to parameter β_j is defined as

$$U_j = \frac{\partial l(\underline{\theta}, \underline{y})}{\partial \beta_j} = \sum_{i=1}^{N} \frac{\partial l_i}{\partial \beta_j}$$

where $$l_i = y_i b(\theta_i) + c(\theta_i) + d(y_i) \qquad (2.3.4)$$

To obtain U_j, we use

$$\frac{\partial l_i}{\partial \beta_j} = \frac{\partial l_i}{\partial \theta_i} \frac{\partial \theta_i}{\partial \mu_i} \frac{\partial \mu_i}{\partial \beta_j} \qquad (2.3.5)$$

By differentiating (2.3.4) and substituting (2.3.1)

$$\frac{\partial l_i}{\partial \theta_i} = y_i b'(\theta_i) + c'(\theta_i) = b'(\theta_i)(y_i - \mu_i) \, . \qquad (2.3.6)$$

By differentiating (2.3.1) and substituting (2.3.3)

$$\frac{\partial \mu_i}{\partial \theta_i} = -\frac{c''(\theta_i)}{b'(\theta_i)} + \frac{c'(\theta_i) b''(\theta_i)}{[b'(\theta_i)]^2} = b'(\theta_i) Var(Y_i) \, . \qquad (2.3.7)$$

By differentiating (2.3.2)

$$\frac{\partial \mu_i}{\partial \beta_j} = \frac{\partial \mu_i}{\partial \eta_i} \frac{\partial \eta_i}{\partial \beta_j} = x_{ij} \frac{\partial \mu_i}{\partial \eta_i} \, .$$

Hence, $$\frac{\partial l_i}{\partial \beta_j} = \frac{\partial l_i}{\partial \theta_i} \frac{\partial \mu_i}{\partial \beta_j} / \frac{\partial \mu_i}{\partial \theta_i} = \frac{(y_i - \mu_i) x_{ij}}{Var(Y_i)} \left(\frac{\partial \mu_i}{\partial \eta_i} \right) \qquad (2.3.8)$$

and therefore $$U_j = \sum_{i=1}^{N} \frac{(y_i - \mu_i) x_{ij}}{Var(Y_i)} \left(\frac{\partial \mu_i}{\partial \eta_i} \right) . \qquad (2.3.9)$$

The elements of the information matrix are defined by

$$J_{jk} = E(U_j U_k) \, .$$

From (2.3.8), for each Y_i the contribution to J_{jk} is

$$E\left[\frac{\partial l_i}{\partial \beta_j}\frac{\partial l_i}{\partial \beta_k}\right] = E\left[\frac{(y_i-\mu_i)^2 x_{ij}x_{ik}}{\{Var(Y_i)\}^2}(\frac{\partial \mu_i}{\partial \eta_i})^2\right]$$

$$= \frac{x_{ij}\,x_{ik}}{Var(Y_i)}(\frac{\partial \mu_i}{\partial \eta_i})^2 .$$

Therefore $$J_{jk} = \sum_{i=1}^{N}\frac{x_{ij}\,x_{ik}}{Var(Y_i)}(\frac{\partial \mu_i}{\partial \eta_i})^2 . \qquad (2.3.10)$$

From (2.3.9), the equations formed by setting $U_j = 0$, $j = 1,...p$ are non-linear equations and they have to be solved with iterative numerical techniques.

Using the Newton-Raphson method the m th approximation for b is given by

$$b^{(m)} = b^{(m-1)} - \left[\frac{\partial^2 l}{\partial \beta_j \partial \beta_k}\right]^{-1}_{\beta=b^{(m-1)}} U^{(m-1)}, \qquad (2.3.11)$$

where $\left[\frac{\partial^2 l}{\partial \beta_j \partial \beta_k}\right]_{\beta=b^{(m-1)}}$ is the matrix of the second derivatives of l evaluated at $\beta = b^{(m-1)}$ and $U^{(m-1)}$ is the vector of the first derivatives $U_j = \frac{\partial l}{\partial \beta_j}$ evaluated at $\beta = b^{(m-1)}$. (**Note**: this is the multidimemential analogue of the Newton-Raphson method for finding a solution of the equation $f(x) = 0$, with m th step:

$$x^{(m)} = x^{(m-1)} - \frac{f[x^{(m-1)}]}{f'[x^{(m-1)}]}.)$$

An alternative procedure which is sometimes simpler than the Newton-Raphson method is called the method of scoring ([35]). It involves replacing the matrix of second derivatives in (2.3.11) by the matrix of expected values:

$$E\left[\frac{\partial^2 l}{\partial\beta_j\partial\beta_k}\right].$$

The information matrix:

$J = E[U\,U^T]$ has the elements:

$$J_{jk} = E[U_j U_k] = E\left[\frac{\partial l}{\partial\beta_j}\frac{\partial l}{\partial\beta_k}\right] = -E\left[\frac{\partial^2 l}{\partial\beta_j\partial\beta_k}\right].$$

Thus (2.3.11) is replaced by

$$b^{(m)} = b^{(m-1)} + [J^{(m-1)}]^{-1} U^{(m-1)} \tag{2.3.12}$$

where $J^{(m-1)}$ denotes the information matrix evaluated at $b^{(m-1)}$. Multiplication by $J^{(m-1)}$ in (2.3.12) gives

$$J^{(m-1)} b^{(m)} = J^{(m-1)} b^{(m-1)} + U^{(m-1)}. \tag{2.3.13}$$

For generalized linear models the (j,k) th element is written as in (2.3.10).

Thus the information matrix J can be written as

$$J = X^T W X$$

where W is the $N \times N$ diagonal matrix with the elements

$$W_{ii} = \frac{1}{Var(Y_i)}\left(\frac{\partial\mu_i}{\partial\eta_i}\right)^2.$$

The expression on the right hand side of (2.3.13) is the vector with elements

$$\sum_k \sum_i \frac{x_{ij}x_{ik}}{Var(Y_i)}\left(\frac{\partial\mu_i}{\partial\eta_i}\right)^2 b_k^{(m-1)} + \sum_i \frac{(y_i-\mu_i)x_{ij}}{Var(Y_i)}\left(\frac{\partial\mu_i}{\partial\eta_i}\right)$$

evaluated at $b^{(m-1)}$; this follows from (2.3.10) and (2.3.11). Thus the right hand side of (2.3.13) can be written as

$$J^{(m-1)}b^{(m-1)} + U^{(m-1)} = X^T W z$$

where z has the elements

$$z_i = \sum_k x_{ik} b_k^{(m-1)} + (y_i - \mu_i)(\frac{\partial \eta_i}{\partial \mu_i}) \qquad (2.3.14)$$

with μ_i and $\frac{\partial \eta_i}{\partial \mu_i}$ evaluated at $b^{(m-1)}$.

Hence the iterative equation can be written as $X^T W X b^{(m)} = X^T W z$. (2.3.15)

Next, we want to obtain estimates of the parameters using the Greatest Deviation correlation coefficient r_{gd}.

Let $E(y_i) = \mu_i$, $g(\mu_i) = \eta_i = \underline{x_i^T \beta}$ for $i = 1,2,\cdots,p$,

$$X_{n\times p} = [\underline{x_1},\ldots,\underline{x_n}]^T, \qquad (2.3.16)$$

and let W = the diagonal weight matrix where $W_{ii} = \frac{(\frac{\partial u_i}{\partial \eta_i})^2}{\sigma_y^2}$, $W = D(W_{ii})$, D indicates a diagonal matrix and x_i^T = vector of the i^{th} observations on all explanatory variables.

(2.3.14) can be rewritten as $z_i = (x_i^T b^{(m-1)}) + (y_i - \mu_i)(\frac{\partial \eta_i}{\partial u_i})$, $i = 1,2,\cdots,n$,

or $$\underline{z} = X b^{m-1} + D(\frac{\partial \eta_i}{\partial u_i})(\underline{y} - \underline{u}) \qquad (2.3.17)$$

Let $W_{n\times n}^{1/2} = D(\sqrt{W_{ii}})$ and $X^* = W^{1/2}X$, $Z^* = W^{1/2}\underline{z}$. (2.3.18)

At step m in the iteration, using (2.3.15):

$$X^T W X \underline{b}^{(m)} = X^T W z$$

$$\Rightarrow \qquad X^T W^{1/2} W^{1/2} X \underline{b}^{(m)} = X^T W^{1/2} W^{1/2} \underline{z}$$

$$\Rightarrow \qquad X^{*T} X^* \underline{b}^{(m)} = X^{*T} \underline{z}^* \qquad (2.3.19)$$

This gives the normal equations for generalized linear models and can be solved by the correlation method using Pearson's r. If we replace r by the Greatest Deviation correlation coefficient r_{gd} we have an iterative r_{gd} method as follows:

With Pearson's r, we had:

$$r(\underline{x}_i^*, \underline{z}^* - X^* \underline{b}^{(m)}) = 0, \qquad (2.3.20)$$

We can solve equation (2.3.19) for $b^{(m)}$ using the Greatest Deviation correlation coefficient r_{gd} as:

$$r_{gd}(\underline{x}_i^*, \underline{z}^* - X^* \underline{b}^{(m)}) = 0, \quad for \quad i = 1, 2, \cdots, p. \qquad (2.3.21)$$

2.4 An Example of Simple Linear Regression for Poisson Distributions

The data in Table 2.2 are counts y_i observed at various values of a covariate x (Annette J. Dobson (1990) "An Introduction to Generalized Linear Models"). This example is a simple linear regression with Poisson responses.

Table 2.2: Poisson regression data

y_i	2	3	6	7	8	9	10	12	15
x_i	-1	-1	0	0	0	0	1	1	1

The data are plotted in Figure 2.1

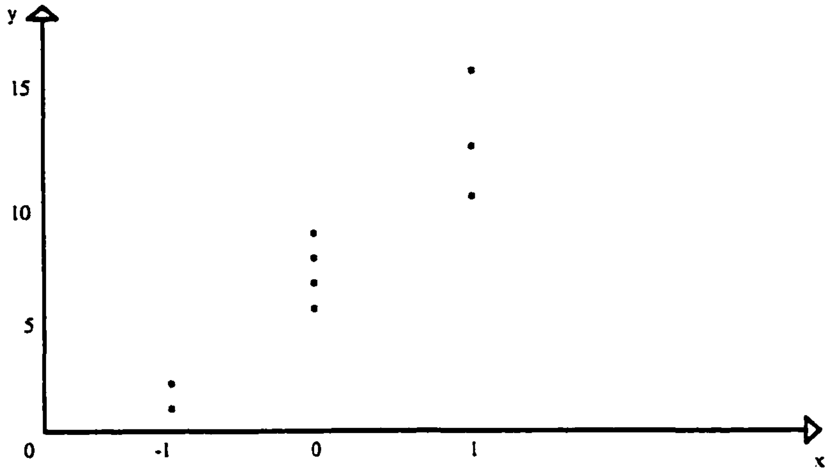


Figure 2.1 Poisson regression data

Assume that the responses Y_i are Poisson random variables. For the Poisson distribution. the expected values and variances of the Y_i's are equal:

$$E(Y_i) = Var(Y_i),\ i = 1,2,\ldots,n.$$

Let us model the relationship between Y_i and X_i with a straight line.

$$E(Y_i) = \mu_i = \beta_0 + \beta_1 x_i = \underline{x}_i^T \underline{\beta}$$

where $\underline{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix}$ and $\underline{x}_i = \begin{bmatrix} 1 \\ x_i \end{bmatrix}$ for $i = 1,\ldots,9$.

The link function here is $g(\mu_i) = \mu_i = \underline{x}_i^T \underline{\beta} = \eta_i$.

Therefore $\partial \mu_i / \partial \eta_i = 1$, $W_{ii} = \dfrac{1}{Var(Y_i)} = \dfrac{1}{\beta_0 + \beta_1 x_i}$,

and from (2.3.18)

$$\underline{z} = X b^{m-1} + \underline{y} - \underline{u} = \underline{y},$$

$$W^{1/2} = D(\sqrt{w_{ii}}) = D\left(\frac{1}{\sqrt{\beta_0 + \beta_1 x_i}}\right),$$

$$X^* = W^{1/2}X = \begin{bmatrix} \frac{1}{\sqrt{\beta_0+\beta_1 x_i}} & & 0 \\ & \ddots & \\ 0 & & \frac{1}{\sqrt{\beta_0+\beta_1 x_9}} \end{bmatrix} \begin{bmatrix} 1 & X_1 \\ . & . \\ . & . \\ . & . \\ 1 & X_9 \end{bmatrix},$$

$$\underline{z}^* = W^{1/2}\underline{z} = \begin{bmatrix} \frac{1}{\sqrt{\beta_0+\beta_1 x_1}} & & 0 \\ & \ddots & \\ 0 & & \frac{1}{\sqrt{\beta_0+\beta_1 x_9}} \end{bmatrix} \begin{bmatrix} y_1 \\ . \\ . \\ . \\ y_9 \end{bmatrix} = \begin{bmatrix} \frac{y_1}{\sqrt{\beta_0+\beta_1 x_1}} \\ . \\ . \\ . \\ \frac{y_9}{\sqrt{\beta_0+\beta_1 x_9}} \end{bmatrix},$$

$$\Rightarrow X^{*T} X^* \underline{b}^{(m)} = X^{*T} \underline{z}^*,$$

$$\Rightarrow \underline{b}^{(m)} = (X^{*T} X^*)^{-1} X^{*T} \underline{z}^*$$

We can then estimate $\underline{b}$ iteratively from the following equation:

$$r_{gd}(\underline{x}_j^*, \underline{z}^* - X^*\underline{b}^{(m)}) = 0, \quad for\ j = 1,2.$$

The iterative process stops when increments in the elements of the $\underline{b}$ vector are small (<0.000001).

We can choose initial values $b_0^{(0)} = 50, b_1^{(0)} = 4.935$. Successive approximations are shown in Table 2.3.

The r_{gd}-based and r-based estimates are given for comparison below:

r_{gd} method to estimate $\underline{b}$: $b_0 = 7.7659, b_1 = 4.935$ (Table 2.2)

least squares method to estimate $\underline{b}$: $b_0 = 7.4516, b_1 = 4.9353$ (Table 2.3).

The two regression fits are shown in Figure 2.2 and Figure 2.3.

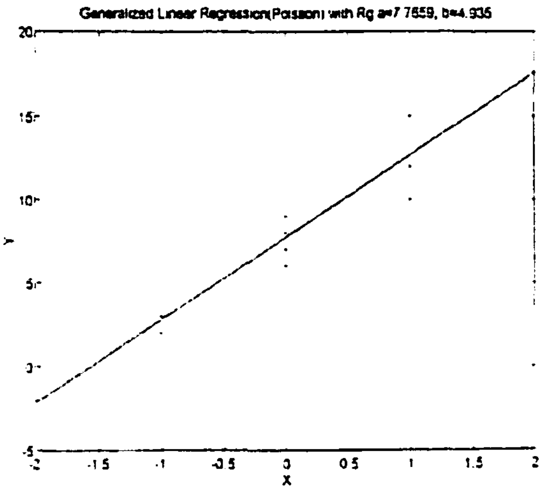


Figure 2.2 r_{gd} method

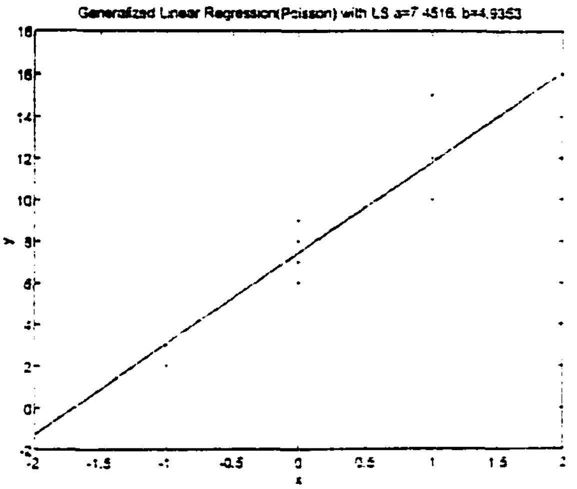


Figure 2.3 least squares method

Table 2.3 Successive approximations of regression coefficients by r_{gd}:

m	0	1	2	3	4	5	6	7	8	9
$b_0^{(m)}$	50	10.990	8.030	7.788	7.768	7.766	7.766	7.7659	7.7659	7.7659
$b_1^{(m)}$	4.9354	4.9354	4.9353	4.9353	4.9353	4.9352	4.9352	4.9352	4.9350	4.9350

Table 2.4 Successive approximations of regression coefficients by least squares:

m	0	1	2	3
$b_0^{(m)}$	7	7.45	7.4516	7.4516
$b_1^{(m)}$	5	4.937	4.9353	4.9353

Table 2.5 Comparison of r_{gd} and glm:

x_i	y_i	r_{gd} : $\hat{y}$	glm: $\hat{y}$
-1	2	2.831	2.5163
-1	3	2.831	2.5163
0	6	7.766	7.4516
0	7	7.766	7.4516
0	8	7.766	7.4516
0	9	7.766	7.4516
1	10	12.701	12.3869
1	12	12.701	12.3869

In this example we fit a simple generalized linear model for Poisson responses. The r_{gd} and least squares methods gave the similar results (see Table 2.5. Figure 2.2 and Figure 2.3).

2.5 An Example of Simple Linear Regression with r_{td}

Kenneth Lange and Jannet S. Sinsheimer studied the robust regression applications of independent, normal distributions to robust regression ([33], 1993). Lange and Sinsheimer studied the properties of normal/independent distributions and presented several results. Consider a positive random variable U and an independent k-variate normal random vector Z with mean 0 and nonsingular covariance matrix Ω. If μ is any constant k-vector, then $Y = \mu + U^{-1}Z$ is said to be normal/independent ([33]). Certain families of normal/independent distributions are particularly attractive for adaptive, robust regression. EM algorithms were discussed for use with robust regression based on the t, slash, and contaminated normal families. The examples illustrated the performance of the different methods on real data and simulated data. They concluded that the slash and t methods perform similarly. The contaminated normal and least squares are more suspect.

The Slash Distribution ([33]):

The multivariate version of the slash distribution (Rogers and Tukey 1972) has scale variable U with density $h(u) = vu^{v-1}$ on $[0,1]$ for $v > 0$. The reciprocal moments

$$E(U^{-m}) = \frac{v}{v-m}$$

exist for $m < v$, and the density of the slash is given by the integral

$$\frac{v}{(2\pi)^{k/2}\,|\,\Omega\,|^{1/2}}\int_0^1 u^{(k/2)+v-1}e^{-u\delta^2/2}du\,,$$ where Ω is a nonsingular covariance matrix.

The Contaminated Normal Distribution ([33]):

For the multivariate contaminated normal (Tukey 1960), the scale variable U is concentrated at the two points $\lambda < 1$ and 1 with masses ϕ and $1-\phi$. Clearly, $E(U^{-m}) = \phi\lambda^{-m} + 1 - \phi$, and $\Pr(\delta^2 \le r) = \phi\Pr(\chi_k^2 \le \lambda r) + (1-\phi)\Pr(\chi_k^2 \le r)$. The density of Y is the mixture

$$\frac{1}{(2\pi)^{k/2}\,|\,\Omega\,|^{1/2}}\left[\phi\lambda^{k/2}e^{-\lambda\delta^2/2} + (1-\phi)e^{-\delta^2/2}\right].$$

The Slash and Contaminated Normal distributions will be used in the following example.

Example:

In this example the real data set comes from Lange and Sinsheimer. We will use the nonparametric correlation coefficient r_{gd} method to estimate the parameters, and then compare it with other methods. Table 2.6 shows the average births and deaths by hour over a 30-year period at a certain hospital in Brussels.

Table 2.6 Birth-Death Data Set

hour (i)	the number of births x_i	the number of deaths y_i at hour i
1	142	228
2	173	253
3	130	230
4	122	242
5	111	213
6	112	217
7	99	248
8	88	207
9	130	228
10	137	311
11	48	110
12	94	257
13	97	233
14	88	217
15	91	237
16	104	281
17	100	233
18	121	204
19	97	194
20	133	199
21	115	220
22	120	231
23	224	243
24	4	14

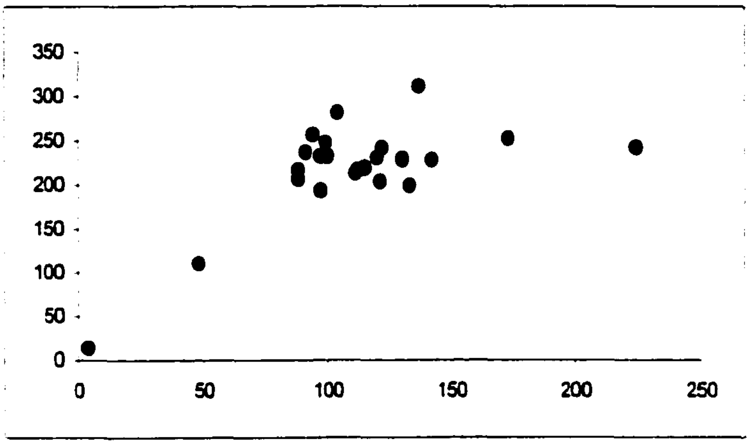


Figure 2.4 A plot of y vs. x

Based on the Figure 2.4, we postulate the linear regression model $\mu_i = E(y_i) = \alpha + \beta x_i$ for $i = 1,...,24$.

This model was fit using a variety of methods, with the fits summarized in Table 2.7.

Table 2.7 Linear model for birth/death data with r_{gd} and other methods

Parameter estimates	r_{gd}	Logistic	Slash	t	Contaminated normal	Normal (LS)	Normal minus 2 outliers*
α	198.188	147.3	202.7	203.0	201.9	114.95	212.3
β	0.25	0.6645	0.2050	0.2008	0.2040	0.9296	0.1730

* The points (4,14) and (48,110) are outliers.

In this example, the influence of outliers causes β to be significantly greater than 0 for the least squares and logistic methods. Deaths by hour, however, should not be correlated with births by hour. When we used the nonparametric correlation coefficient r_{gd} methodology, the estimated correlation diminished greatly. The β estimators for the slash, t and contaminated normal also show this feature and show the similar results as

r_{gd}. The least squares and logistic methods fail to downweight the outliers sufficiently, as would be expected (Figures 2.5, 2.6, 2.7, 2.8, 2.9, 2.10, 2.11).

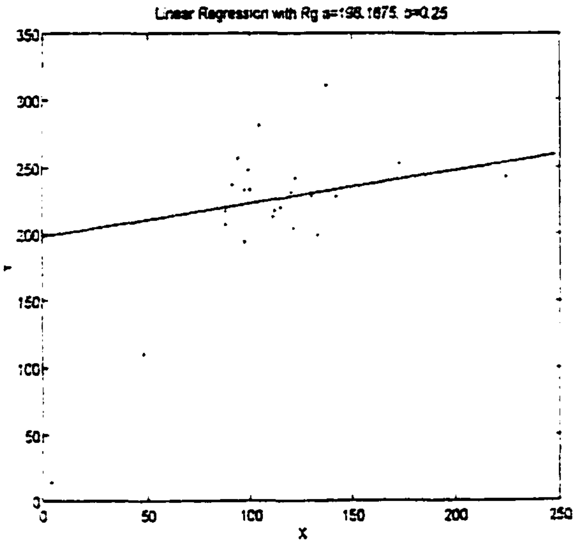


Figure 2.5 r_{gd} method

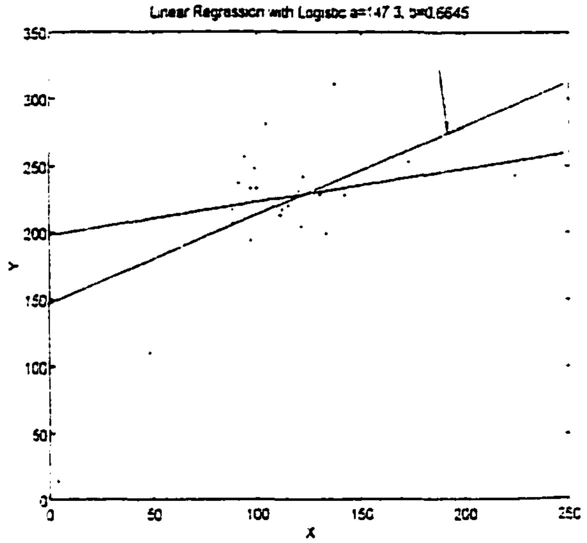


Figure 2.6 Logistic method

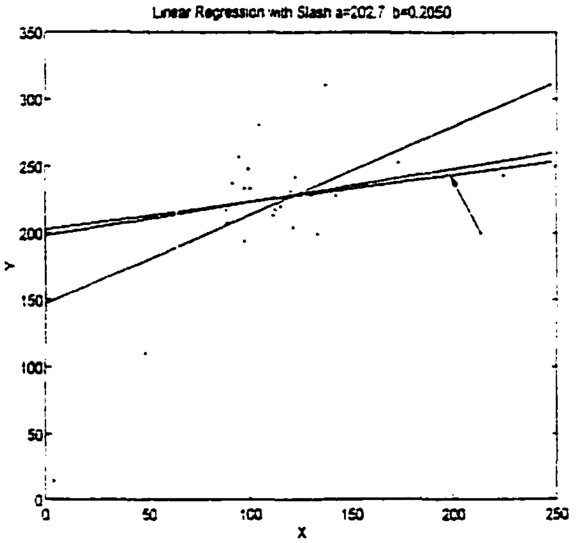


Figure 2.7 Slash method

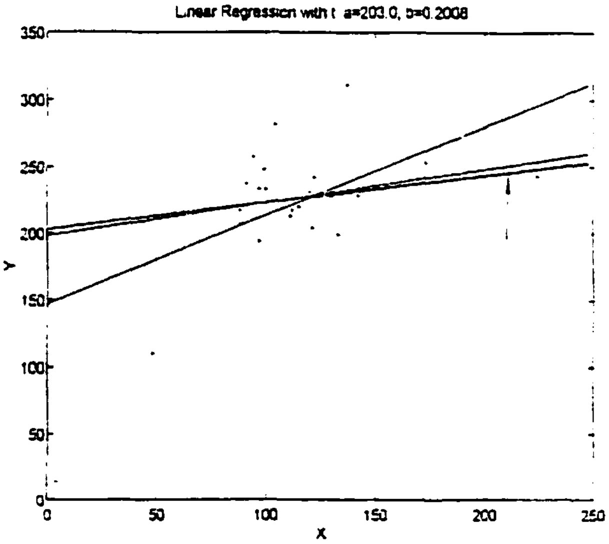


Figure 2.8 t method

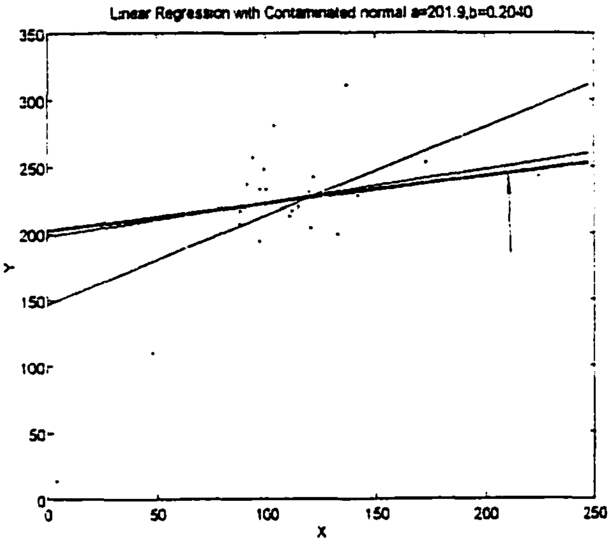


Figure 2.9 Contaminated normal

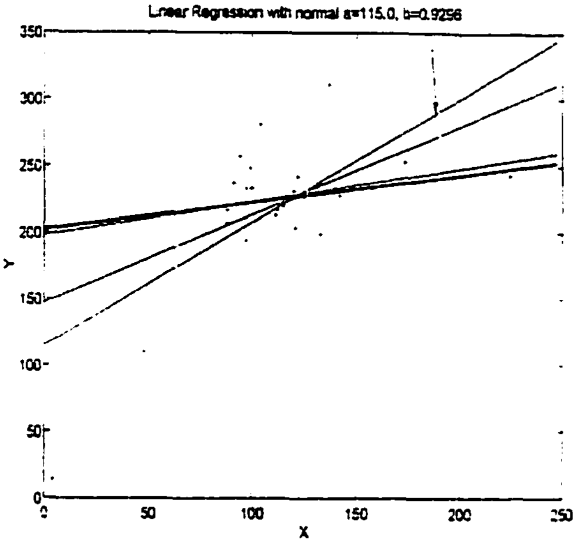


Figure 2.10 Normal distribution

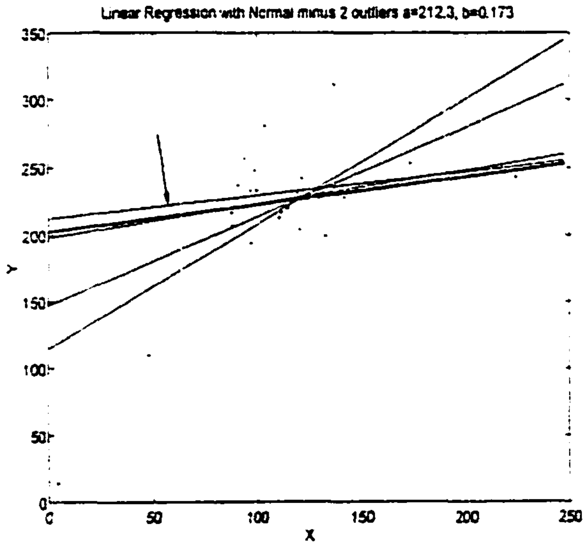


Figure 2.11 Normal minus 2 outliers

In conclusion, the Greatest Deviation correlation coefficient r_{gd} method on this example is more robust and less sensitive to the extreme outliers than the least squares method. The r_{gd}, slash and t methods perform similarly. The least squares method is more suspect.

2.6 Logistic Regression

In this section we consider the special case of generalized linear models in which the outcome variable is measured on a binary scale.

The specific form of the logistic regression model we will use is as follows:

$$\pi(x) = \frac{e^{\alpha+\beta x}}{1+e^{\alpha+\beta x}}, \text{ or equivalently:} \qquad (2.6.1)$$

$$\log[\frac{\pi(x)}{1-\pi(x)}] = \alpha + \beta x = \underline{x}_i^T\underline{\beta}. \qquad (2.6.2)$$

The difference between the linear and logistic regression models concerns the conditional distribution of the outcome variable. In the linear regression model, we assume that an observation of the outcome variable may be expressed as $y = E(Y\,|\,x) + \varepsilon$. The most common assumption is that the error ε follows a normal distribution with mean zero and some variance that is constant across levels of the independent variable. It follows that the conditional distribution of the outcome variable given x will be normal with mean $\mu = E(Y\,|\,x)$, and a variance that is a constant. This is not the case with a dichotomous outcome variable. In this situation, we may express the value of the outcome variable given x as $y = \pi(x) + \varepsilon$. Here the quantity ε may assume one of two possible values. If

$y = 1$ then $\varepsilon = 1 - \pi(x)$, and if $y = 0$ then $\varepsilon = -\pi(x)$. Thus, ε has a distribution with mean zero and variance equal to $\pi(x)[1-\pi(x)]$. That is, the conditional distribution of the outcome variable follows a binomial distribution with probability given by the conditional mean, $\pi(x)$.

2.6.1 Fitting the Logistic Regression Model

Suppose we have a sample of n independent pairs of observations (x_i, y_i), $i = 1,2,...,n$, where y_i denotes the value of a dichotomous outcome variable and x_i is the value of the independent variable for the ith subject. To fit the logistic regression model in equation (2.6.1) to a set of data requires that we estimate the values of α and β, the unknown parameters.

The maximum liklihood method is the standard approach to estimation for the logistic regression model. For k distinct x_i values where there are n_i outcomes for x_i, $i = 1,....,k$, the likelihood function for $\underline{\beta}$ is given as:

$$L(\underline{\beta}) = \prod_{i=1}^{k} \pi_i^{y_i}(1-\pi_i)^{n_i - y_i}, \tag{2.6.3}$$

where y_i = binomial observation at x_i, n_i of them, k groups.

The log-likelihood function is:

$$\log L = \sum_{i=1}^{k} \{y_i \log \pi_i + (n_i - y_i)\log(1-\pi_i)\} \tag{2.6.4}$$

$$\Rightarrow \quad \log L(\alpha,\beta) = \sum_{i=1}^{k} y_i \log(\frac{\pi_i}{1-\pi_i}) + \sum_{i=1}^{k} n_i \log(1-\pi_i)$$

$$= \sum_{i=1}^{k} y_i(\alpha + \beta x_i) - \sum_{i=1}^{k} n_i \log\{1 + \exp[\alpha + \beta x_i]\}$$

Differentiating with respect to α :

$$\frac{\partial \log L}{\partial \alpha} = \sum_{i=1}^{k} y_i - \sum_{i=1}^{k} \frac{n_i \exp[\alpha + \beta x_i]}{1 + \exp[\alpha + \beta x_i]}$$

$$= \sum_{i=1}^{k} (y_i - n_i \pi_i) = 0 \qquad (2.6.6)$$

Differentiating with respect to β :

$$\frac{\partial \log L}{\partial \beta} = \sum_{i=1}^{k} y_i x_i - \sum_{i=1}^{k} \frac{n_i x_i}{1 + \exp[\alpha + \beta x_i]}$$

$$= \sum_{i=1}^{k} y_i x_i - \sum_{i=1}^{k} n_i x_i \pi_i = 0 \qquad (2.6.7)$$

$$= \sum_{i=1}^{k} x_i (y_i - n_i \pi_i) = 0$$

i.e., $\underline{x} \perp (\underline{y} - \underline{\hat{y}})$.

The residual vector is given by: $\underline{res} = y_i - n_i \hat{\pi}_i = \underline{y} - \underline{\hat{y}}$

where $\hat{\pi}_i = \dfrac{1}{\{1 + \exp[-(a + bx_i)]\}}$. (2.6.8)

Note: Logistic regression is a type of generalized linear model with link function $\eta = \log\{\pi/(1-\pi)\}$.

From equation (2.3.18), we have $\underline{z}^* - X^*\underline{b} = W^{1/2}[X\underline{b} + D(\frac{\partial \eta_i}{\partial u_i})(\underline{y} - \underline{u})] - W^{1/2}X\underline{b}$

$= W^{1/2} D(\frac{\partial \eta_i}{\partial u_i})(\underline{y} - \underline{u}) = W^{1/2} D(\frac{\partial \eta_i}{\partial u_i})(\underline{y} - n_i \underline{\pi})$. For the logistic regression model, we can rewrite equation (2.3.22) as $r_{gd}(\underline{x}, \underline{y} - \underline{\hat{y}}) = 0$, i.e., $r_{gd}(\underline{x}, \underline{res}) = 0$.

If we use the nonparametric correlation coefficient r_{gd} method, α and β are found by solving:

$$\begin{aligned} &r_{gd}(\underline{x}, \underline{res}) = 0 \\ &median(\underline{res}) = 0 \end{aligned} \qquad \text{where } \underline{res} = \text{the residual vector.} \qquad (2.6.9)$$

2.6.2 Testing for the Significance of the Coefficients

In logistic regression, comparison of observed to predicted values can be based on the log likelihood function defined in equation (2.6.4).

One way to compare observed to predicted values using the likelihood function is based on the following deviance statistic:

$$D = -2\log\left[\frac{(likelihood\ of\ the\ current\ \text{mod}\,el)}{(likelihood\ of\ the\ saturated\ \text{mod}\,el)}\right]. \qquad (2.6.10)$$

Let $\hat{\pi}_i$ = MLE of π_i under the model of interest.

For the maximal model, we take the π_i s as the parameters to be estimated. Then,

$$\frac{\partial l}{\partial \pi_i} = \frac{y_i}{\pi_i} - \frac{n_i - y_i}{1 - \pi_i},$$

so the solution of $\partial l / \partial \pi_i = 0$ is y_i / n_i.

The predicted response in the i^{th} group under the saturated model is: $\tilde{\pi} = y_i / n_i$.

$$\log L(saturated\ model) = \sum_{i=1}^{k}\{y_i \log\frac{y_i}{n_i} + (n_i - y_i)\log(1-\frac{y_i}{n_i})\}$$

$$D = -2\{\log L(current\ model) - \log L(saturated\ model)\}$$

$$= 2\{\sum_{i=1}^{k}\{y_i(\log\frac{y_i}{n_i} - \log\frac{\hat{y}_i}{n_i}) + (n_i - y_i)[\log(1-\frac{y_i}{n_i}) - \log(1-\frac{\hat{y}_i}{n_i})]\} \qquad (2.6.11)$$

$$= 2\sum_{i=1}^{k}[y_i \log(\frac{y_i}{\hat{y}_i}) + (n_i - y_i)\log(\frac{n_i - y_i}{n_i - \hat{y}_i})].$$

This function behaves in much the same way as the residual sum of squares or weighted residual sum of squares in ordinary linear models.

Under H_0 (current model is true), D = Deviance = 2[log(saturated model) - log(current model)] has an asymptotically χ^2_{k-p} distribution, where p is the number of parameters in the current model. Assuming k cells in the full model and 2 parameters in the reduced model, α and β, there are $k-2$ degrees of freedom for D. However, "asymptotically" here means k is fixed and $n_i \to \infty$ for each i. If k is increasing, but the n_i remains bounded, then this asymptotic result does not hold. In fact, if $n_i = 1$ for all i, then D is meaningless as a goodness-of-fit statistic.

2.6.3 Examples of Logistic Regression

Example 1. Table 2.8 shows data on the number of insects dead after five hours's exposure to gaseous carbon disulphide at various concentrations (Annette J. Dobson [12]).

Table 2.8

Dose, x_i	Number of insects , n_i	Number killed, y_i
1.6907	50	6
1.7242	60	13
1.7552	62	18
1.7842	56	28
1.8113	63	52
1.8369	59	53
1.8610	62	61
1.8839	60	60

Figure 2.12 shows a plot of π_i vs. x_i.

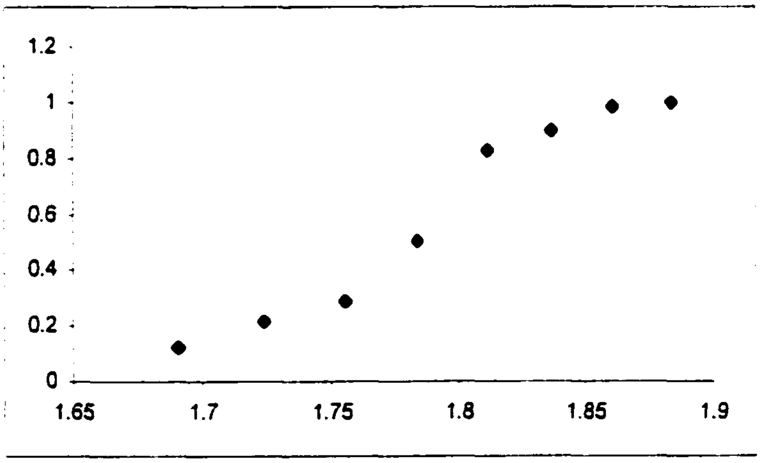


Figure 2.12 Plot of π_i vs. x_i

We begin by fitting the logistic model:

$$\pi(x) = \frac{e^{\alpha+\beta x}}{1+e^{\alpha+\beta x}} \Leftrightarrow \log[\frac{\pi(x)}{1-\pi(x)}] = \alpha + \beta x = \underline{x}_i^T \underline{\beta}$$

The maximum likelihood method results in the following normal equations for α and β:

$$\sum_{i=1}^{k}(y_i - n_i\hat{\pi}_i) = 0$$

$$\sum_{i}^{k} x_i(y_i - n_i\hat{\pi}_i) = 0$$

Now $y_i - n_i\hat{\pi}_i$ is the ith group residual, res_i, so that the first equation says $\sum_{i=1}^{k} res_i = 0$ and the second equation says $cor(\underline{x}, \underline{res}) = 0$, i.e., Pearson's correlation of the x vector with the residual vector is zero. Replacing Pearson's correlation by r_{gd}, the two equations are

$$median(\underline{res}) = 0,$$
$$r_{gd}(\underline{x}, \underline{res}) = 0.$$

Table 2.9 shows the parameter estimates of α and β using the r_{gd} method.

$$\left.\begin{array}{l} b_0^{(209)} = -60.63 \\ b_1^{(209)} = 34.34 \end{array}\right. \text{ after 209 iterations.}$$

We used C and S-plus functions to estimate the parameters α and β. The computations converged after 209 iterative steps.

Table 2.9 Fitting the logistic model to the beetle mortality data by r_{gd}

m	0	1	10	100	......	209
$b_0^{(m)}$	-63.7	-62.68	-62.59	-61.69	......	-60.63
$b_1^{(m)}$	36.27	32.29	32.38	33.28	......	34.34

The parameter estimators of α and β using the least squares method were: $b_0 = -59.8, b_1 = 33.672$. Since there is no outlier in this logistic example, the r_{gd} method and least squares method give similar results, but r_{gd} method is better than least squares on the left and right tails of the observations (see Figures 2.15 and 2.16). But, if data are simulated with two outliers for these data: (1.89, 0.2903) and (1.99, 0.5), The results of these two parameter estimation methods are quite different (see Table 2.10 and Figures 2.13 and 2.14).

Table 2.10 Comparison of r_{gd} and least squares methods

	without outlier	with 2 outliers
r_{gd} method	$b_0 = -60.63$ $b_1 = 34.34$	$b_0 = -56.7$ $b_1 = 32.27$
LS method	$b_0 = -59.80$ $b_1 = 33.672$	$b_0 = -26.134$ $b_1 = 14.746$

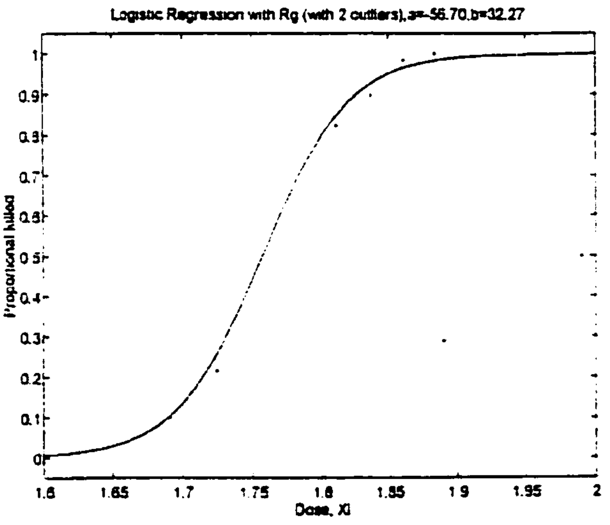


Figure 2.13 r_{gd} with 2 outliers

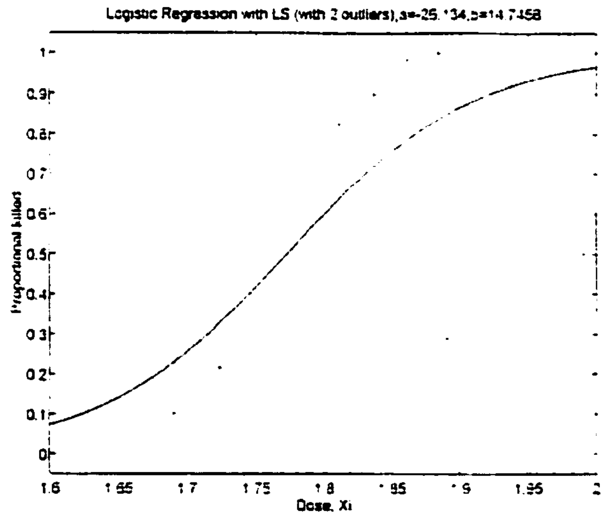


Figure 2.14 Least squares with 2 outliers

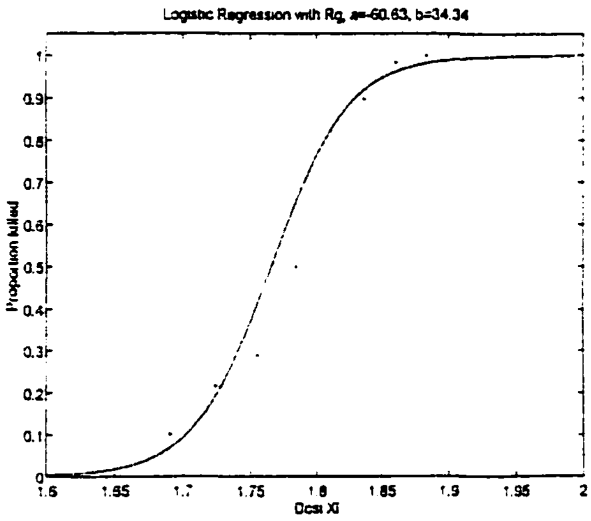


Figure 2.15 r_{gd} without outlier

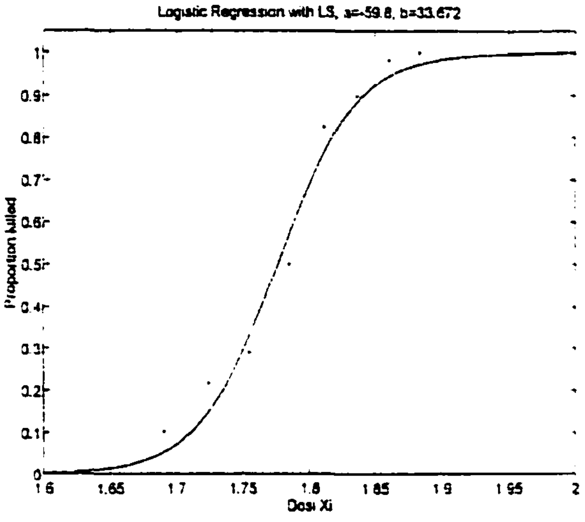


Figure 2.16 Least squares without outlier

The data set with 2 outliers (Table 2.10) shows the advantage of the robust r_{gd} regression method. The influence of outliers causes b to change significantly for the least squares method. The least squares method fails to downweight the outliers sufficiently (Figures 2.13 and 2.14).

Example 2: The data for this second example come from 175 Atlanta Braves games from the 1992 season. For each game, define

$$\text{the response variable } y = \begin{cases} 1 \textit{ for a win} \\ 0 \textit{ for a loss} \end{cases}.$$

For each game, the number of Atlanta hits minus the number of Opponent hits was computed. A frequency table for this variable x, called the hit difference, the variable y called the number of wins in n games, are given in Table 2.11.

Table 2.11 (Data A) Original Data:

y	x	n
0	-10	2
0	-9	2
0	-8	1
0	-7	5
1	-6	6
1	-5	5
3	-4	12
5	-3	13
1	-2	10
9	-1	20
14	0	18
11	1	15
9	2	13
11	3	13
10	4	11
9	5	9
5	6	5
5	7	5
4	8	4
2	9	2
1	11	1
1	12	1
1	13	1
1	16	1

We grouped the ends of data, see Table 2.12 (Data B).

Table 2.12 (Data B) Ends of data grouped

wins	rundiff	weights	wins/weights
0	-7	10	0
1	-6	6	0.17
1	-5	5	0.20
3	-4	12	0.25
5	-3	13	0.38
1	-2	10	0.10
9	-1	20	0.45
14	0	18	0.78
11	1	15	0.73
9	2	13	0.69
11	3	13	0.85
10	4	11	0.91
9	5	9	1.00
5	6	5	1.00

15	7	15	1.00

It was desired to study how the probability of winning a game is related to x. Logistic regression was used to model this relationship.

Let $\pi(x)$ = the probability of winning a game given x, the hit difference. It will be expected that $\pi(0)$ would be $1/2$.

The logistic model is given by: $\pi(x) = \frac{1}{1+\exp[-(\alpha+\beta x)]}$ or logit $\pi(x)$ = $\log(\frac{\pi(x)}{1-\pi(x)}) = \alpha + \beta x$

The maximum likelihood method leads to the following normal equations for α and β:

$$\sum_{i=1}^{k}(y_i - n_i\hat{\pi}_i) = 0$$

$$\sum_{i=1}^{k}x_i(y_i - n_i\hat{\pi}_i) = 0$$

where there are k groups of data based on the k distinct x_i values, and $\hat{\pi}_i$ = $\frac{1}{1+\exp[-(\alpha+\beta x_i)]}$.

For Data A, we fit the logistic model using both the r_{gd} method and least squares method. The results of these estimation methods appear in the left column of Table 2.13.

For Data A, the r_{gd} method and least squares method are similar in their regression fits (Figures 2.17 and 2.18). However, if the outlying data point (5, 0.1) is added to the data, the resulting parameter estimates change as seen in the right column of Table 2.13.

Table 2.13 Comparison of r_{gd} and least squares method

	without outlier	with one outlier
r_{gd} method	a = 0.50415, b = 0.47415	a = 0.472, b = 0.447
LS method	a = 0.4216, b=0.4635	a = 0.30671, b = 0.33423

Table 2.13 shows the advantage of the r_{gd} robust regression method. The influence of the outlier causes a and b to change significantly for the least squares method. The least squares method fails to downweight the outlier sufficiently for logistic regression (Figures 2.19 and 2.20).

Consider testing:

$$H_0: \beta=0$$

$$H_a: \beta\neq 0$$

Recall the deviance statistic, defined as:

Deviance = 2[log(saturated model) - log(constrained model)].

We use the r_{gd} method to compute the Null Deviance and Residual Deviance as follows:

Null Deviance = 2[log(saturated model) - log(intercept only)] = 90.7, $d.f. = 23$

Residual Deviance = 2[log(saturated model) - log(intercept+slope)] = 11.5, $d.f. = 22$

Null Deviance - Residual Deviance = 2[log(intercept+slope) - log(intercept only)]

= 90.7-11.5

= 79.2, which is χ_1^2 if H_0 is true.

The p-value for this test is:

$p(\chi_1^2 \geq 10.83) = 0.001$, so the slope is significant, i.e., $\beta\neq 0$.

For Data B, we also fit the logistic model using the r_{gd} and least squares methods.

The resulting estimates and likelihoods evaluated at those estimates are given below:

r_{gd} method: $a = 0.572$, $b = 0.422$, $\hat{L} = 0.572 + 0.422x$

least squares: $a = 0.41779$, $b = 0.45514$, $\hat{L} = 0.41779 + 0.45514x$

For Data B, it appears from Figure 2.21 and 2.22 that the r_{gd} method supplies a better fit than the least squares method.

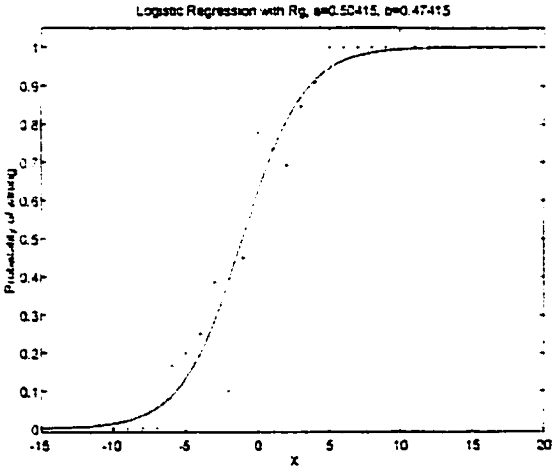


Figure 2.17 Logistic regression with r_{gd}

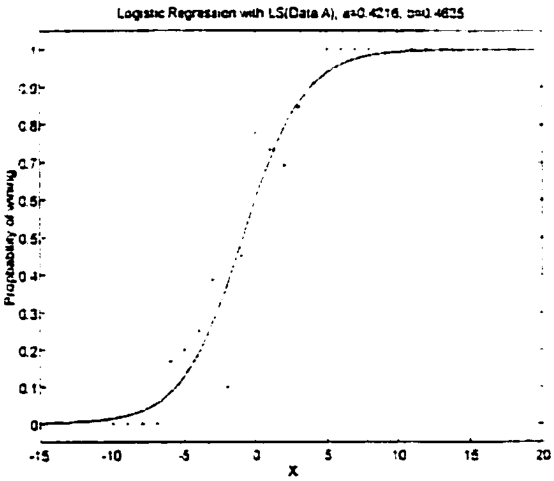


Figure 2.18 Logistic regression with LS

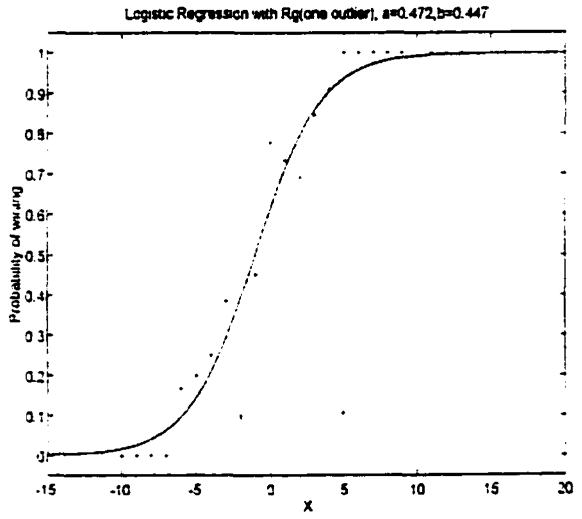


Figure 2.19 Logistic regression with r_{gd} (outlier)

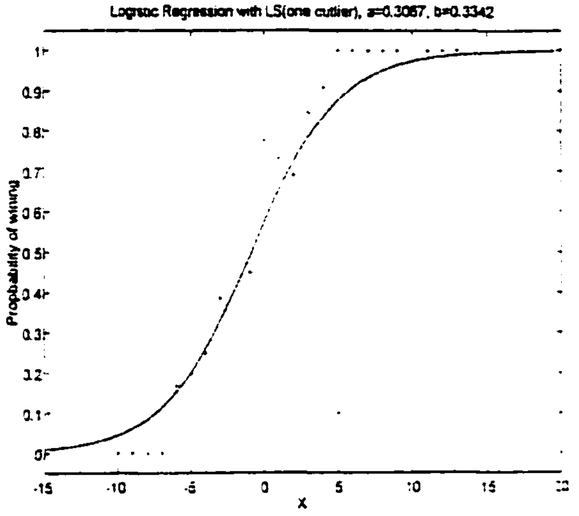


Figure 2.20 Logistic regression with LS (outlier)

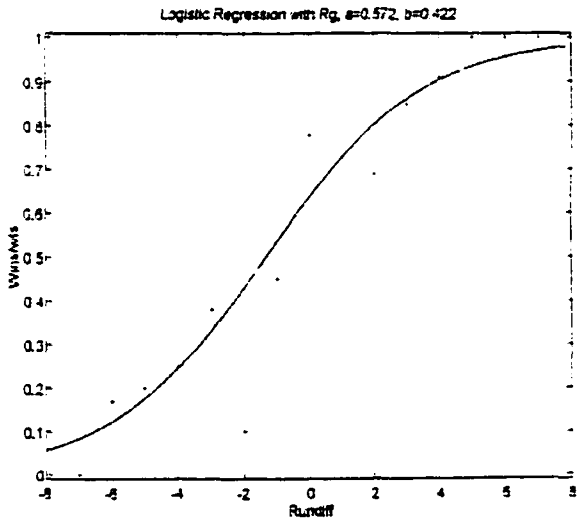


Figure 2.21 Logistic regression with r_{gd}

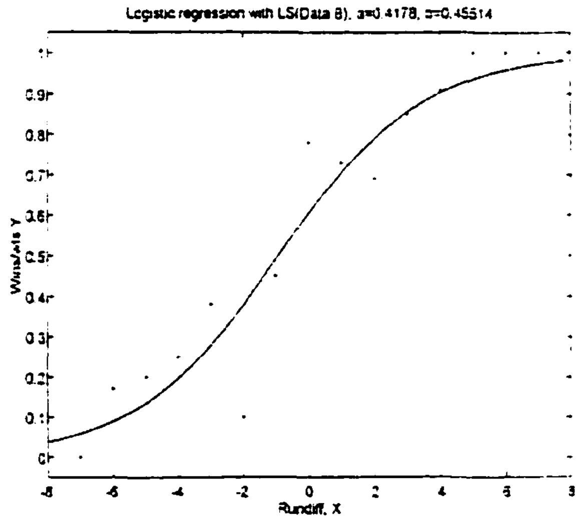


Figure 2.22 Logistic regression with LS

Chapter 3

Nonlinear Models and Estimation

3.1 Introduction

The general linear model can be written as:

$$Y = \beta_0 + \beta_1 x_1 + \ldots + \beta_n x_n + \varepsilon . \qquad (3.1.1)$$

Any model which is not of the form (3.1.1) will be called a nonlinear model. In general. whenever a linear regression model does not appear to adequately represent the relationship between variables, then a nonlinear regression model might be appropriate. Nonlinear estimation is a general fitting procedure that will estimate the parameters defining any kind of relationship between a response variable, and a list of explanatory variables. In general, all regression models may be expressed in the form:

$$E(y \mid \underline{x}) = f(x_1, x_2, \ldots, x_n) \qquad (3.1.2)$$

In most general terms, we are interested in whether and how a response variable is related to a list of explanatory variables.

Nonlinear estimation allows us to specify essentially any type of regression model. Some common nonlinear models are probit, logit, and exponential growth or decay models. We can also use any number of fitting techniques to estimate the model parameters. More precisely, we can use standard least squares estimation, maximum likelihood estimation or define some "loss function" to be minimized. In this chapter we will use the nonparametric correlation coefficient r_{gd} method to estimate the parameter in several nonlinear models. Some common nonlinear models are reviewed below:

(1) Growth Rate Model

Some regression models which cannot be transformed into linear ones, can only be estimated via nonlinear estimation. The Growth Rates are often affected by many variables (other than time), and we can expect a considerable amount of random residual fluctuation around the fitted line. If we add this error or residual variability to the model, it might be written as follows:

$$Growth = \exp(-b_1 * Age) + error. \tag{3.1.2}$$

In this additive error model, we assume that the error variability is independent of age, i.e., that the amount of residual error variability is the same at any age. Because the error term in this model is additive, we can no longer linearize this model by taking the logarithm of both sides. If for a given data set, we were to log-transform the variable *Growth* anyways and fit the simple linear model, then we would find that the residuals from the analysis would no longer be evenly distributed over the range of ages; and thus, a standard linear regression analysis would no longer be appropriate. Therefore, the parameters for this model should be estimated using nonlinear estimation techniques.

(2) General Exponential Growth Model

The general exponential growth model, is similar to the example that we previously considered:

$$y = b_0 + b_1 * \exp(b_2 * x) + error, \text{ where } b_0, b_1, b_2 > 0 \tag{3.1.3}$$

This model is commonly used in studies of any kind of population growth. An example where this model would be adequate is when we want to describe healthcare insured membership as a function of time

(3) Models for Binary Responses

We studied binary response models in Chapter 2. It is not uncommon that a dependent or response variable is binary in nature, i.e., it can have only two possible values. For example, patients either do or do not recover from an injury; job applicants either succeed or fail at an employment test, etc. In all of these cases, we are interested in estimating a model that describes the relationship between one or more explanatory variables to the binary response variable.

Logistic regression

We studied logistic regression in Chapter 2. In the logistic regression model, the predicted values for the response variable will never be less than (or equal to) 0, or greater than (or equal to) 1, regardless of the values of the explanatory variables. The general form of the logistic regression model is given below:

$$E(y \mid \underline{x}) = \exp(b_0 + b_1 * x_1 + ... + b_n * x_n) / \{1 + \exp(b_0 + b_1 * x_1 + ... + b_n * x_n)\}. \qquad (3.1.4)$$

We can easily recognize that, regardless of the regression coefficients or the magnitude of the x values, this model will always produce expected values (expected y's) in the range of 0 to 1.

Suppose we think of the binary response variable y in terms of its underlying continuous probability π for a given $\underline{x}$, ranging from 0 to 1.

$$\eta = \log\{\pi/(1-\pi)\}. \tag{3.1.5}$$

The function η is also called the link function. Note that η can theoretically assume any value between minus and plus infinity. Since the logit transform solves the issue of the 0/1 boundaries for the original response variable (probability), we could use those logistic transformed values as the responses in an ordinary linear regression equation. In fact, if we perform the logistic transform on both sides of the logit regression equation (3.1.4), we obtain the standard linear regression model:

$$\eta = b_0 + b_1 * x_1 + b_2 * x_2 + \ldots + b_n * x_n + \varepsilon. \tag{3.1.6}$$

We have listed and described some common nonlinear models. In this chapter we will use the Greateast Deviation correlation coefficient r_{gd} method to estimate the parameters in nonlinear models.

3.2 Nonlinear Model Estimation

3.2.1 Loss Functions

Some common nonlinear estimation procedures are:

(1) Least Squares

We have reviewed some common nonlinear models in the previous section. Now, the question arises as to how the parameters in these models are estimated. In the most general terms, least squares estimation is aimed at minimizing the sum of squared deviations of the observed values for the response variable from those predicted by the model.

In standard multiple regression we estimate the regression coefficients by "finding" those coefficients that minimize the residual variance (sum of squared residuals) around the regression line. Any deviation of an observed value from a predicted value signifies some loss in the accuracy of our prediction, possibly, due to random error. Therefore, the goal of least squares estimation is to minimize a loss function; specifically, this loss function is defined as the sum of the squared deviations about the predicted values. When this function is at its minimum, then we get the parameter estimates (regression coefficients). Because of the particular loss function that yielded those estimates, we can call the estimates least squares estimates.

There are several common function minimization methods that can be used to minimize various types of loss functions.

(2) Weighted Least Squares

In addition to least squares regression, weighted least squares estimation is a commonly used estimation technique. Ordinary least squares techniques assume that the residual variance around the regression line is the same across all values of the independent variables. In another words, it is assumed that the error variance in the measurement of each case is identical. Often, this is not a realistic assumption; in particular, violations frequently occur in business, economic, or biological applications.

For example, suppose we wanted to study the relationship between the projected cost of construction projects, and the actual cost. This may be useful in order to gage the expected cost overruns. In this case it is reasonable to assume that the absolute magnitude (dollar amount) by which the estimates are off, is proportional to the size of the project and hence nonconstant. Thus, we might use a weighted least squares loss function to fit a linear regression model. Specifically, the loss function would be Loss = (Observed-Predicted)^2 * (1/x^2).

In this equation, the loss function first specifies the standard least squares loss function, and then weights this loss by the inverse of the squared value of the explanatory variable (x) for each case. The larger the project (x) the less weight is placed on the deviation from the predicted value (cost).

(3) Maximum Likelihood

An alternative to the least squares loss function is to maximize the likelihood or log-likelihood function (or to minimize the negative log-likelihood function). In most general terms, the likelihood function is defined as the product of the individual probability functions:

$$L = \prod_{i=1}^{n} f(y_i, \underline{\theta}). \tag{3.2.1}$$

Maximum Likelihood requires a distributional assumption (normal distribution) on the errors.

3.2.2 Function Minimization Algorithms

Now that we have discussed different regression models, and the loss functions that can be used to estimate them, we want to know how to minimize the loss functions to find the best fitting set of parameters, and how to estimate the standard errors of these parameter estimates. One very efficient algorithm that approximates the second-order derivatives of the loss function to guide the search for the minimum (i.e., for the best parameter estimates, given the respective loss function) is the quasi-Newton method. In addition, there are several other general function minimization algorithms that follow different search strategies (which do not depend on the second-order derivatives). These strategies are sometimes more effective for estimating loss functions with local minima; therefore, these methods are often particularly useful for finding appropriate starting values for the estimation via the quasi-Newton method.

Start Values, Step Sizes, Convergence Criteria

A common aspect of most nonlinear estimation procedures is that they require the user to specify some starting values for the parameters, initial step sizes for the iterative search, and a criterion for convergence. These methods will begin with a particular set of initial estimates, which will be changed in some systematic manner from iteration to iteration; in the first iteration, the step size determines by how much the parameters will changed. Finally, the convergence criterion determines when the iteration process will stop. For example, the process may stop when the improvements in the loss function from iteration to iteration are less than a specific amount.

Quasi-Newton Method

The slope of a function at a particular point can be computed as the first-order derivative of the function at that point. The "slope of the slope" is the second-order derivative, which tells us how fast the slope is changing at the respective point, and in which direction. The quasi-Newton method will, at each step, evaluate the loss function at different points in order to estimate the first-order derivatives and second-order derivatives. It will then use this information to follow a path towards the minimum of the loss function.

Nonparametric Correlation Coefficient r_{gd} Method

In Chapters 1 and 2 we illustrated the use of the Greatest Deviation correlation coefficient r_{gd}, which is robust to outliers, in fitting multiple linear and generalized linear regression models. This chapter will extend the method from linear and generalized linear models to nonlinear models. The following sections will illustrate the method of steepest descent with r_{gd} for the estimation of nonlinear model parameters.

3.3 Least Squares in Nonlinear Regression.

Suppose we have a nonlinear model of the following form:

$$Y = f(x_1,\cdots,x_k;\beta_1,\cdots,\beta_p)+\varepsilon . \tag{3.3.1}$$

Let

$\underline{x} = (x_1,\cdots,x_k)'$ = the observed values for k explanatory variables.

$\underline{\beta} = (\beta_1,\cdots,\beta_p)'$ = the vector of fixed but unknown model parameters.

Then (3.3.1) can be written as

$$y = f(\underline{x},\underline{\beta})+\varepsilon \tag{3.3.2}$$

Assume there are n independent observations:

$y_u, x_{1u}, x_{2u},\cdots, x_{ku}$, for $u = 1,2,..,n$,

$$y_u = f(\underline{x_u};\underline{\beta})+\varepsilon_u \tag{3.3.3}$$

where $\underline{x}_u = (x_{1u}, x_{2u}, \cdots, x_{ku})'$

The assumption of normality and independence of errors can now be written as $\underline{\varepsilon} \sim N(\underline{0}, I\sigma^2)$, for $\sigma^2 > 0$ a fixed but unknown constant.

We define the error sum of squares for the nonlinear model as:

$$s(\underline{\beta}) = \sum_{u=1}^{n} [y_u - f(\underline{x}_u, \underline{\beta})]^2$$

The least squares estimate $\hat{\underline{\beta}}$ is a value of $\underline{\beta}$ which minimizes $s(\underline{\beta})$.

The least squares estimate of $\underline{\beta}$ is also the maximum likelihood estimate of $\underline{\beta}$ (since $\underline{\varepsilon} \sim N(\underline{0}, I\sigma^2)$).

The likelihood function can be written as:

$$L(\underline{\beta}, \sigma^2) = (2\pi\sigma^2)^{-n/2} \exp\left[\frac{-s(\underline{\beta})}{2\sigma^2}\right] \tag{3.3.4}$$

where $s(\underline{\beta}) = \sum_{u=1}^{n} [Y_u - f(\underline{x}_u, \underline{\beta})]^2$

If σ^2 is known, maximizing $L(\underline{\beta}, \sigma^2)$ with respect to $\underline{\beta}$ is equivalent to minimizing $s(\underline{\beta})$ with respect to $\underline{\beta}$.

Differentiating $s(\underline{\beta})$ with respect to $\underline{\beta}$ yields the following set of normal equations:

$$\sum_{u=1}^{n} \{Y_u - f(\underline{x}_u, \hat{\underline{\beta}})\} \left[\frac{\partial f(\underline{x}_u, \underline{\beta})}{\partial \beta_i}\right]_{\beta = \hat{\beta}} = 0, \text{ for } i = 1,2,\ldots p. \tag{3.3.5}$$

3.4 Nonlinear Least Squares using the Newton and Steepest Descent Method

A common unconstrained minimization problem requiring iterative techniques involves fitting a nonlinear model to data. If least squares is appropriate, then this problem can be written in the form:

$$\min F(x_1,\ldots,x_n) = \min \sum_{j=1}^{m}[f_j(x_1,\ldots,x_n)]^2 \tag{3.4.1}$$

where the nonlinear function $f_j(x_1,\ldots,x_n)$ represents the residual for the j^{th} data point. Nonlinear least squares is an unconstrained minimization problem, which can be solved by a number of iterative numerical techniques, such as Newton's method or the method of steepest descent. These techniques are illustrated below:

Newton's method

Consider the Taylor's series approximation:

$$F(x+p) \approx F(x) + p^T \nabla F(x) + \tfrac{1}{2} p^T \nabla^2 F(x) p \equiv F(x) + Q(p) \tag{3.4.2}$$

where the higher order term is a quadratic function in p. To obtain the step p, we now minimize the remainder term $Q(p)$ as a function of p by forming its gradient with respect to p:

$$\nabla_p Q(p) = \nabla_p (p^T \nabla F(x) + \tfrac{1}{2} p^T \nabla^2 F(x) p) = \nabla F(x) + \nabla^2 F(x) p, \tag{3.4.3}$$

and set it equal to zero, giving:

$$\nabla^2 F(x) p = -\nabla F(x). \tag{3.4.4}$$

This is a set of n linear equations in the n unknowns $p = (p_1,\ldots,p_n)^T$. These linear equations are called the Newton equations. Thus at the $(k+1)$ th step:

$$x_{k+1} = x_k + p = x_k - [\nabla^2 F(x_k)]^{-1} \nabla F(x_k). \tag{3.4.5}$$

The steepest descent method can be used to produce a convergent method:

Steepest Descent Method

A steepest descent algorithm for performing nonlinear least squares is summarized below:

Given an initial value x_0, set $k = 0$.

1. At the k th step, compute $F_k = F(x_k)$ and $\nabla F_k = \nabla F(x_k)$, the function and gradient values at x_k. Test for convergence. If converged, i.e., $| x_k - x_{k-1} |<$ specific amount, stop.

2. Compute a descent direction p, i.e., a direction p such that $F(x_k + \varepsilon \cdot p) < F_k$ for ε small. This is equivalent to requiring that $p^T \nabla F_k < 0$.

3. Line search: Find α > 0 such that $F(x_k + \alpha \cdot p) < F_k$. Set $x_{k+1} = x_k + \alpha \cdot p$, return to step 1.

It is sometimes possible to guarantee that this Newton method will produce a decent direction Suppose that inverse Hessian matrix $\nabla^2 F^{-1}$ is positive definite, i.e., it satisfies the condition $z^T \nabla^2 F^{-1} z > 0$ for all z ≠0. In this case, the Newton direction is guaranteed to reduce F. To see this, note that for some $\varepsilon > 0$:

$$F(x + \varepsilon p) = F(x) + \nabla F^T (\varepsilon p) + o(\varepsilon^2)$$

$$= F(x) + \varepsilon \nabla F^T (-\nabla^2 F^{-1} \nabla F) + o(\varepsilon^2) \quad \text{(From (3.4.4))}$$

$$= F(x) - \varepsilon \nabla F^T \nabla^2 F^{-1} \nabla F + o(\varepsilon^2). \qquad (3.4.6)$$

Since $\nabla^2 F^{-1}$ is positive definite, $\nabla F^T \nabla^2 F^{-1} \nabla F > 0$ as long as $\nabla F \neq 0$. Thus if ε is small and $\nabla F \neq 0$, then $F(x + \varepsilon p) < F(x)$, i.e., p is in a downhill direction.

If we have $B_k \approx \nabla^2 F_k = \nabla^2 F(x_k)$, then the step at the k-th iteration will be defined by

$$B_k p = -\nabla F_k .$$

This step p will be used within the steepest descent method above. After the line search obtains $x_{k+1} = x_k + \alpha \cdot p$, B_k is updated to produce the new approximate B_{k+1}, using the values of x_{k+1} and $\nabla F(x_{k+1})$.

The new Hessian approximation will be chosen so that

$$B_{k+1}(x_{k+1} - x_k) = \nabla F_{k+1} - \nabla F_k \tag{3.4.7}$$

Use of this approximation is called the quasi-Newton method. The advantages of the quasi-Newton method over Newton's method are (1) it is possible to choose B_k to be positive definite so that a descent direction is always obtained; (2) only gradient values are used, avoiding the calculation of derivatives.

In this chapter we use the quasi Newton's method and steepest descent method for the estimation of nonlinear model parameters and compare with the r_{gd} method.

3.5 Parameter Estimation with Pearson's Correlation Coefficient and r_{gd}

The Taylor series approximation in (3.4.2) can be rewritten as:

$$f(\underline{x_u}, \underline{\beta}) \approx f(\underline{x_u}, \underline{\beta_0}) + \sum_{i=1}^{p} \left[\frac{\partial f(\underline{x_u}, \underline{\beta})}{\partial \beta_i} \right]_{\beta = \beta_0} (\beta_i - \beta_{i0}) . \tag{3.5.1}$$

Let $\quad f_u^o = f(\underline{x_u}, \underline{\beta_0})$,

$\theta_i^0 = \beta_i - \beta_{i0}$,

$$z^0_{iu} = \left[\frac{\partial f(\underline{x_u}, \underline{\beta})}{\partial \beta_i}\right]_{\beta=\beta_0} \quad \text{, so that:} \tag{3.5.2}$$

$$y_u - f^o_u = \sum_{i=1}^{p} \theta^o_i Z^o_{iu} + \varepsilon_u \, .$$

This rearrangement results in the following normal equations:

$$Z_0^T Z_0 \underline{\theta_o} = Z_0^T \underline{y_0} \, , \tag{3.5.3}$$

where $$Z_0 = \begin{bmatrix} z^0_{11} & z^0_{12} & \cdot & \cdot & z^0_{1p} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ z^0_{n1} & z^0_{n2} & \cdot & \cdot & z^0_{np} \end{bmatrix}_{n\times p} ,$$

$$\underline{\theta_o} = \begin{bmatrix} \theta^0_1 \\ \cdot \\ \cdot \\ \cdot \\ \theta^0_p \end{bmatrix}_{p\times 1} , \quad \text{and} \quad \underline{y_o} = \begin{bmatrix} y_1 - f^0_1 \\ \cdot \\ \cdot \\ \cdot \\ y_u - f^0_u \end{bmatrix}_{u\times 1}$$

We can solve the normal equations by the correlation method using Pearson's r (least squares):

$$r(\underline{z_j}, \underline{y_o} - Z_o \underline{\theta_o}) = 0 \, , \tag{3.5.4}$$

If we replace r by r_{gd} then we have an iterative r_{gd} method as follows:

$$r_{gd}(\underline{z_j}, \underline{y_o} - Z_o \underline{\theta_o}) = 0 \qquad \text{for } j = 1,2,\ldots,p \, . \tag{3.5.5}$$

3.6 Examples of Nonlinear Regression with the r_{gd} and Least Squares Methods

Example 1: The example which follows is taken from an investigation performed at Procter and Gamble and reported by H. Smith and S. D. Dubey in "Some reliability problems in the chemical industry," (*Applied Regression Analysis*, N. R. Draper and H. Smith). We illustrate how a solution can be obtained for the parameters in a nonlinear model using the r_{gd} method. The investigation involved a product A which must have a fraction 0.50 of Available Chlorine at the time of manufacture. The fraction of Available Chlorine in the product decreases with time. The data are given in Table 3.1.

Table 3.1 Per Cent of Available Chlorine in a Unit of Product

Length of Time since produced (weeks)	Available Chlorine
8	0.49, 0.49
10	0.48, 0.47, 0.48, 0.47
12	0.46, 0.46, 0.45, 0.43
14	0.45, 0.43, 0.43
16	0.44, 0.43, 0.43
18	0.46, 0.45
20	0.42, 0.42, 0.43
22	0.41, 0.41, 0.40
24	0.42, 0.40, 0.40
26	0.41, 0.40, 0.41
28	0.41, 0.40
30	0.40, 0.40, 0.38
32	0.41, 0.40
34	0.40
36	0.41, 0.38
38	0.40, 0.40
40	0.39
42	0.39

It was postulated that with y = available chlorine and x = length of time since produced (weeks) a nonlinear model of the form

$$y = \alpha + (0.49 - \alpha)e^{-\beta(x-8)} + \varepsilon \qquad (3.6.1)$$

would suitably account for the variation observed in the data, for $x \geq 8$. The problem is to estimate the parameters α and β of the non-linear model (3.6.1) using the data given in the table.

To linearize the model into the form (3.6.1) we need to evaluate the first derivative of

$$f(\underline{x_u}, \underline{\beta}) = \alpha + (0.49 - \alpha)e^{-\beta(\underline{x_u} - 8)} \ . \tag{3.6.2}$$

where $\underline{x_u}$ = length of time since produced (weeks).

Differentiatng:

$$\begin{aligned} \frac{\partial f}{\partial \alpha} &= 1 - e^{-\beta(x_u - 8)}, \\ \frac{\partial f}{\partial \beta} &= -(0.49 - \alpha)(x_u - 8)e^{-\beta(x_u - 8)}. \end{aligned} \tag{3.6.3}$$

The resulting Taylor series expansion at the m th step is:

$$\begin{aligned} y_u - f^{(m)}(\underline{x_u}) \approx {} & [1 - e^{-\beta_m(x_u - 8)}](\alpha - \alpha_m) + \\ & [-(0.49 - \alpha_m)(x_u - 8)e^{-\beta_m(x_u - 8)}](\beta - \beta_m) \, . \end{aligned} \tag{3.6.4}$$

In matrix form, this can be expressed as:

$$\underline{y} - \underline{f} = Z^{(m)} \begin{bmatrix} \alpha - \alpha_m \\ \beta - \beta_m \end{bmatrix}.$$

Premultiplying both sides by $Z^{(m)}$ gives the normal equations:

$Z^{(m)^T} Z^{(m)} \underline{\theta} = Z^{(m)^T} (\underline{y} - \underline{f})$ where:

$$Z^{(m)}=\begin{bmatrix} 1-e^{-\beta_m(x_1-8)} & -(0.49-\alpha_m)(x_1-8)\,e^{-\beta_m(x_1-8)} \\ \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ 1-e^{-\beta_m(x_u-8)} & -(0.49-\alpha_m)(x_u-8)\,e^{-\beta_m(x_u-8)} \\ \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ 1-e^{-\beta_m(x_{44}-8)} & -(0.49-\alpha_m)(x_{44}-8)\,e^{-\beta_m(x_{44}-8)} \end{bmatrix}$$

and $\underline{\theta}=\begin{bmatrix}\alpha-\alpha_m \\ \beta-\beta_m\end{bmatrix}$.

From (3.5.4), we then need to solve:

$$r_{gd}(\underline{z}_j,(\underline{y}-\underline{f}^{(m)})-Z\underline{\theta})=0$$

or $$r_{gd}(\underline{z}_j,(\underline{y}-\underline{f}^{(m)})-Z\underline{\theta})=0 \quad j=1,2 \qquad (3.6.5)$$

The results of nonlinear regression show as follows:

r_{gd} method: $\mu_x=0.3902+(0.49-0.3902)\exp(-0.1028*(x-8))$.

LS with steepest descent method:

$$\mu_x=0.3901+(0.49-0.3901)\exp(-0.1016*(x-8)).$$

There are no outliers in this example; hence, the r_{gd} and least squares methods give similar results (Figures 3.1 and 3.2). Beginning the iteration with initial guesses of $\alpha_0=0.32$ and $\beta_0=0.04$ and applying equation (3.6.5) iteratively, the r_{gd} estimates converged after 14 steps and least squares estimates after 4 steps. The iterations for both methods are summarized in Table 3.2.

Table 3.2 Iterative steps for least squares and r_{gd}

	least squares		r_{gd} method	
iteration(j)	α_j	β_j	α_j	β_j
0	0.30	0.02	0.32	0.04
1	0.8416	0.1007	0.3478	0.0568
2	0.3901	0.1004	0.3546	0.1005
3	0.3901	0.1016	0.3581	0.1243
4	0.3901	0.1016	0.3602	0.1237
...			...	...
13			0.3902	0.1028
14			0.3902	0.1028

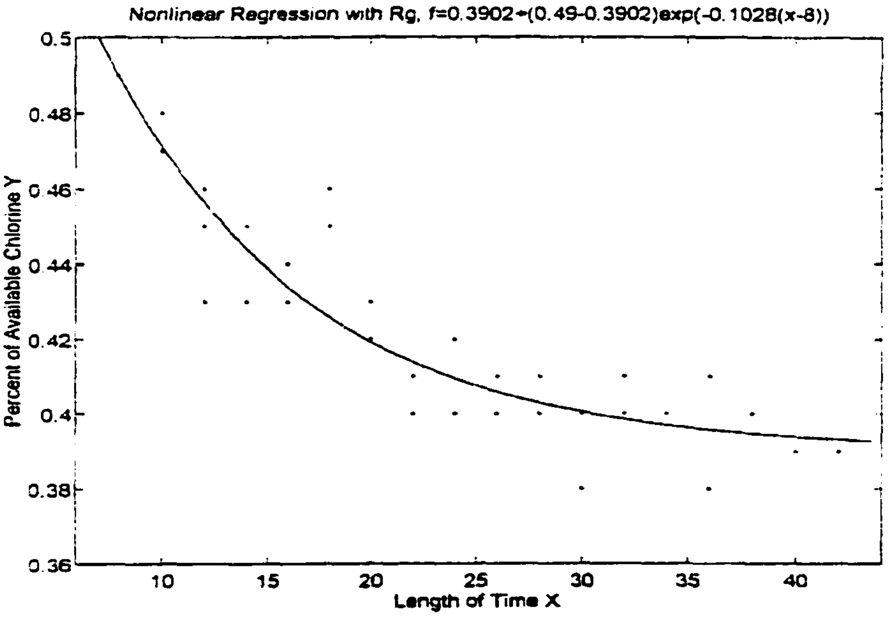


Figure 3.1 Nonlinear Regression with r_{gd}

In the above example, suppose there exists one outlier, at X=18 (length of time since produced) and Y=0.8 (available chlorine).

The estimation with r_{gd} and least squares using steepest descent methods gave the following results, as summarized in Table 3.3:

r_{gd} method: $\mu_t = 0.3896 + (0.49 - 0.3896)\exp(-0.1022 * (x-8))$

LS with Newton and steepest descent method:

$$\mu_t = 0.3445 + (0.49 - 0.3445)\exp(-0.039056 * (x-8))$$

Table 3.3 Comparison of r_{gd} and LS (Newton or steepest descent method)

	with no outlier	with one outlier
r_{gd} method	$\hat{\alpha} = 0.3902$ $\hat{\beta} = 0.1028$	$\hat{\alpha} = 0.3896$ $\hat{\beta} = 0.1022$
LS method	$\hat{\alpha} = 0.3901$ $\hat{\beta} = 0.1016$	$\hat{\alpha} = 0.3445$ $\hat{\beta} = 0.0391$

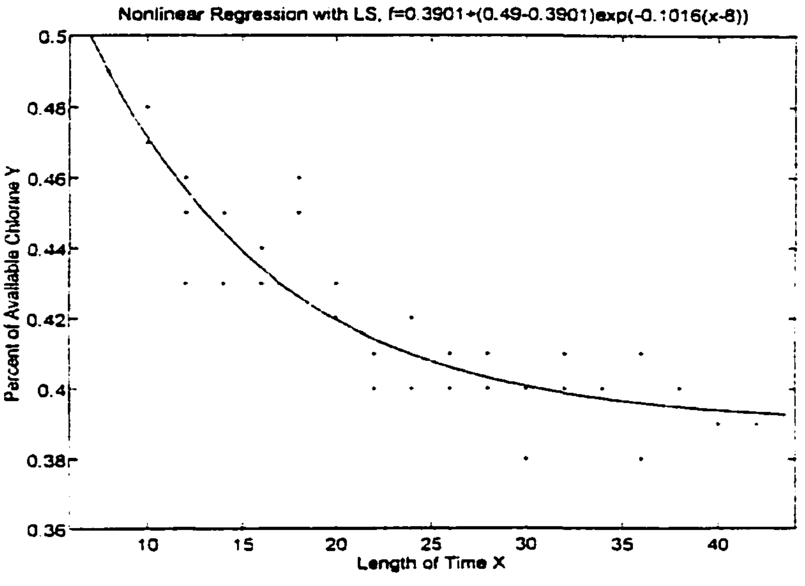


Figure 3.2 Nonlinear Regression with LS

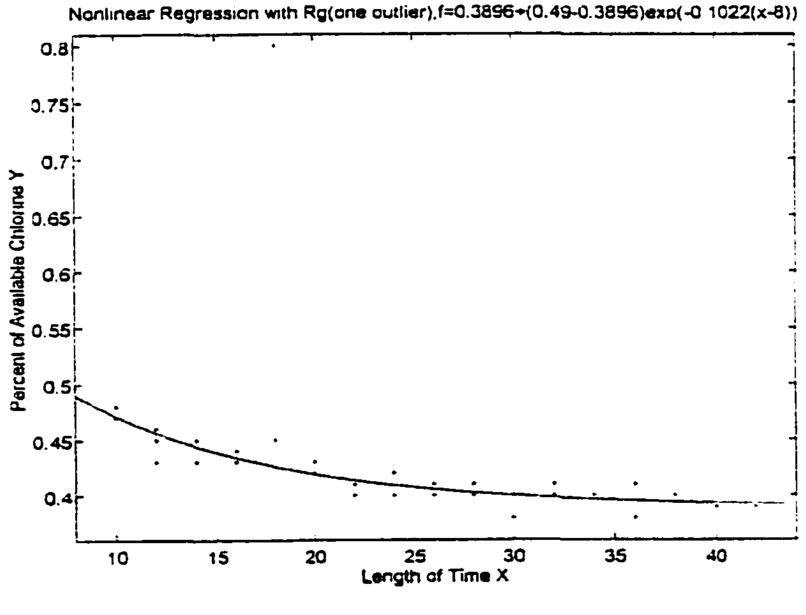


Figure 3.3 Nonlinear Regression with r_{gd} (one outlier)

In this particular example where there is one outlier, the r_{gd} method is clearly more robust to the effects of the outlier than least squares (see Figures 3.3 and 3.4). The influence of the outlier causes α and β to change significantly with least squares. The least squares method fails to downweight the outlier sufficiently. This example illustrates the advantage of the robust r_{gd} regression method.

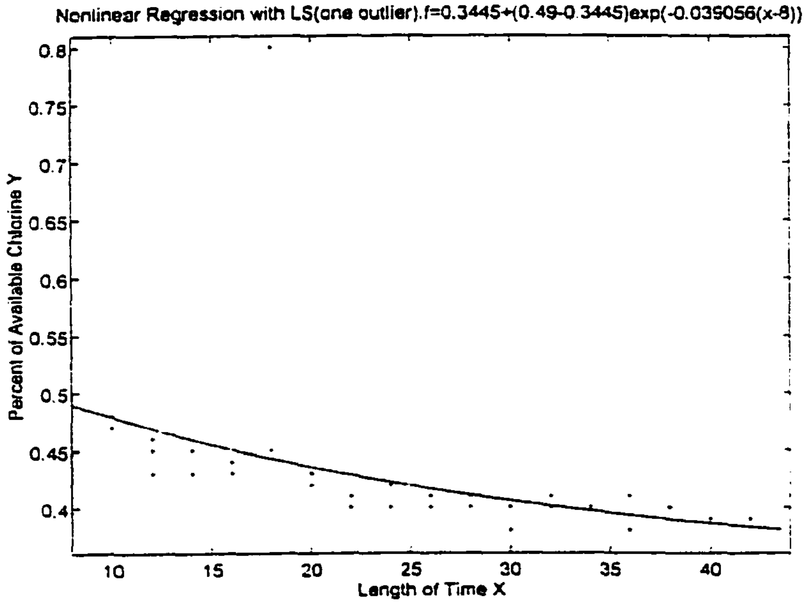


Figure 3.4 Nonlinear Regression with LS (one outlier)

Example 2. The data set for this second example comes from canine myocardium blood-flow calibration (Kenneth Lange and Janet S. Sinsheimer [33], 1993). The 251 cases relate a medically invasive measurement of blood flow x_t to a non-invasive measurement of extraction times blood flow y_t based on positron tomography (We received the original data sets from Lang and Sinsheimer).

Based on a scatter plot, we postulate the nonlinear mean function $\mu_t = x_t(1-\alpha e^{-\beta/x_t})$ where μ_t = mean of y_t.

Table 3.4 Blood Flow Calibration

Parameter estimates	r_{gd}	Logistic	Slash	t	Contaminated normal	Normal	Normal minus 4 outliers
α	0.774	0.7513	0.7435	0.7457	0.7444	0.7818	0.7399
β	270.40	279.3	271.8	274.7	270.8	306.0	267.2

In this example, the r_{gd}, logistic, slash, t, contaminated normal, and normal minus 4 outliers perform well for blood-flow calibration data. The normal (LS) method, however, performed poorly due to the 4 outliers. The parameter estimates for each case are given in table 3.4.

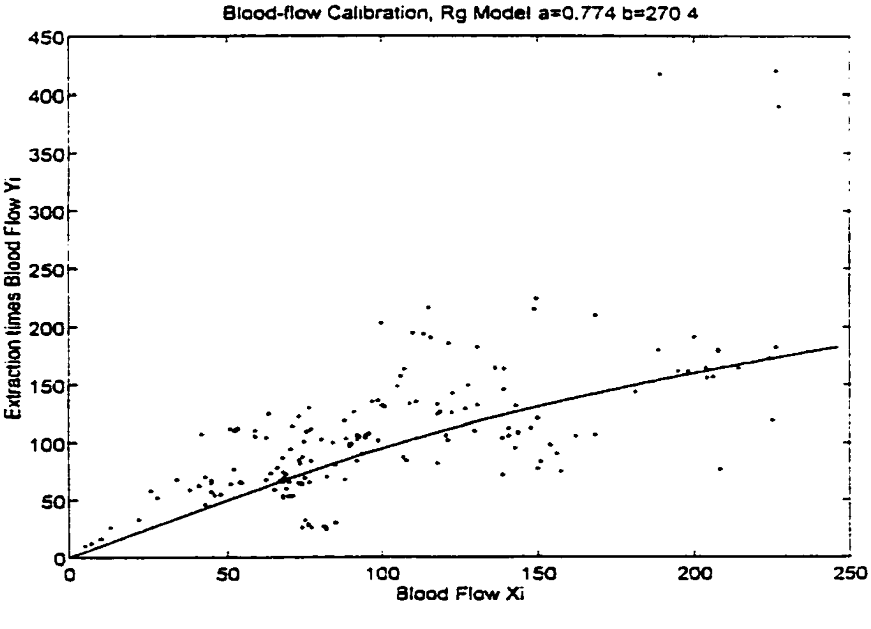


Figure 3.5 Blood-flow Calibration, r_{gd} model

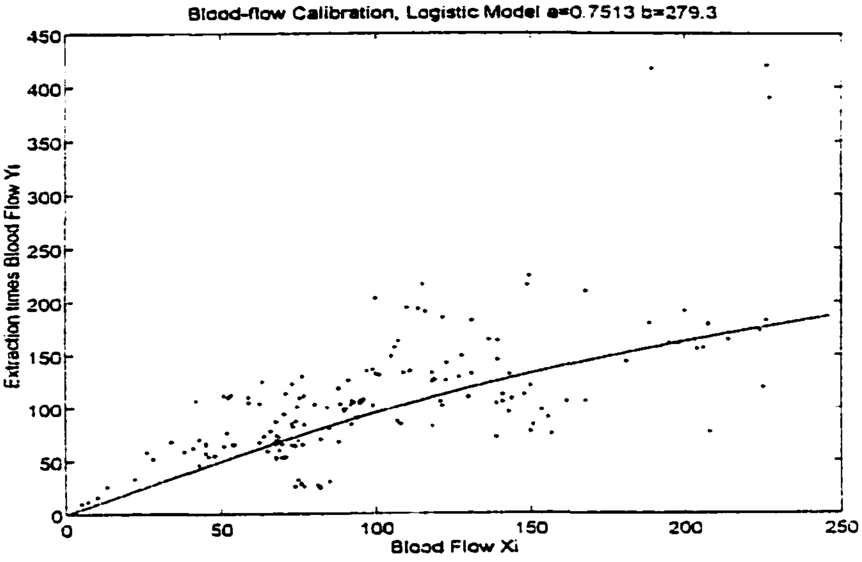


Figure 3.6 Blood-flow Calibration, Logistic model

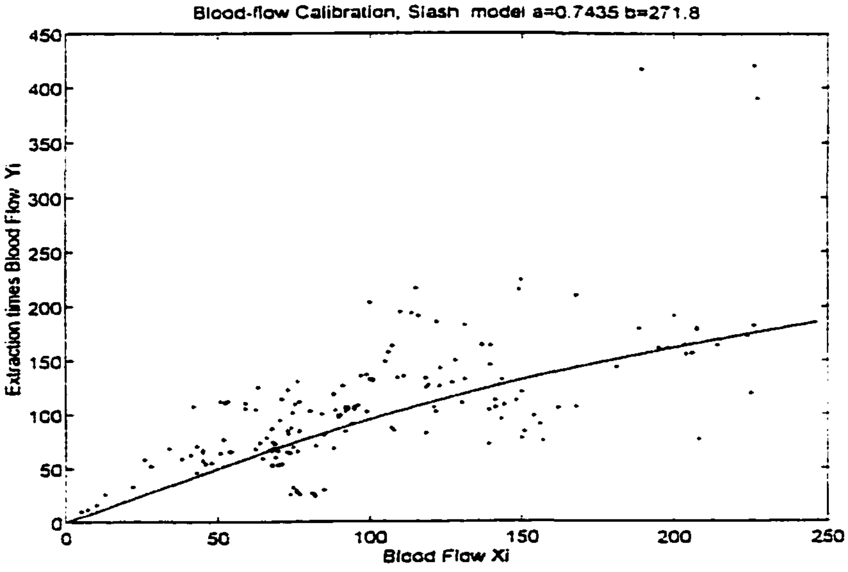


Figure 3.7 Blood-flow Calibration, Slash model

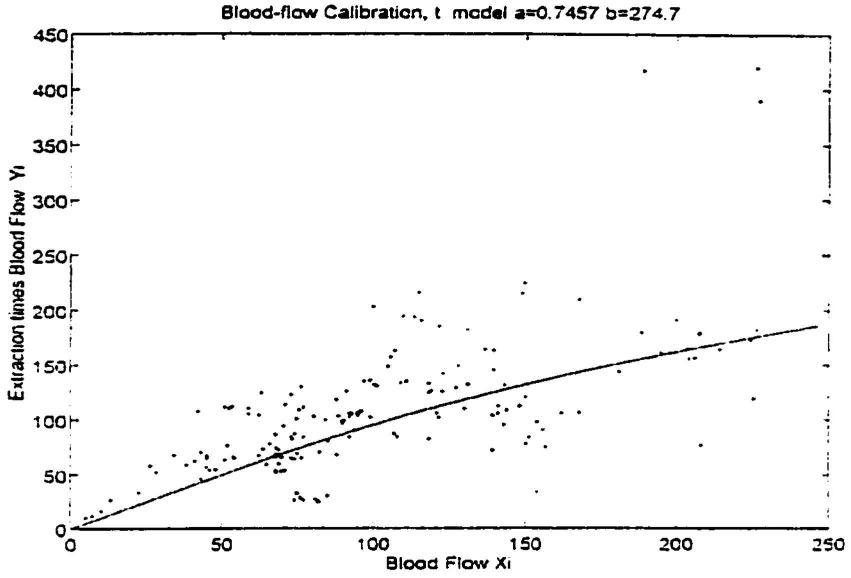


Figure 3.8 Blood-flow Calibration, *t* model

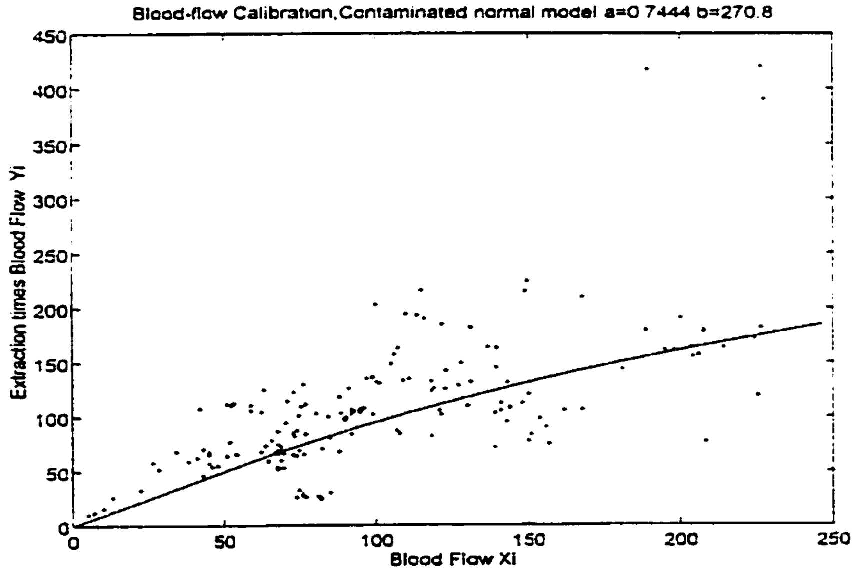


Figure 3.9 Blood-flow Calibration, Contaminated normal model

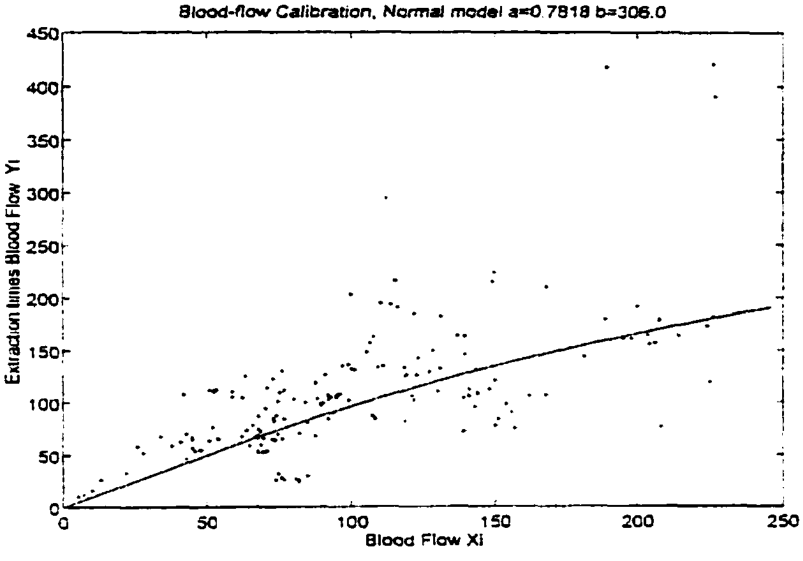


Figure 3.10 Blood-flow Calibration Normal model

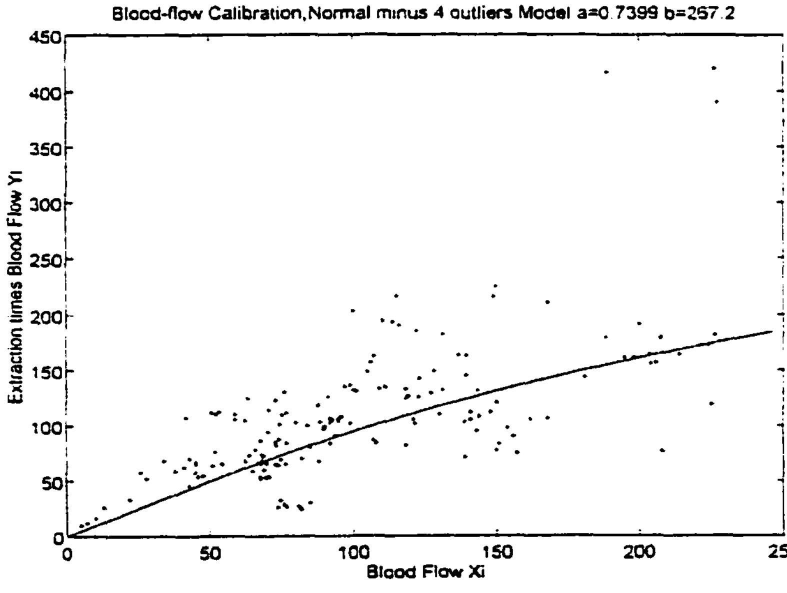


Figure 3.11 Blood-flow Calibration, Normal minus 4 outliers model

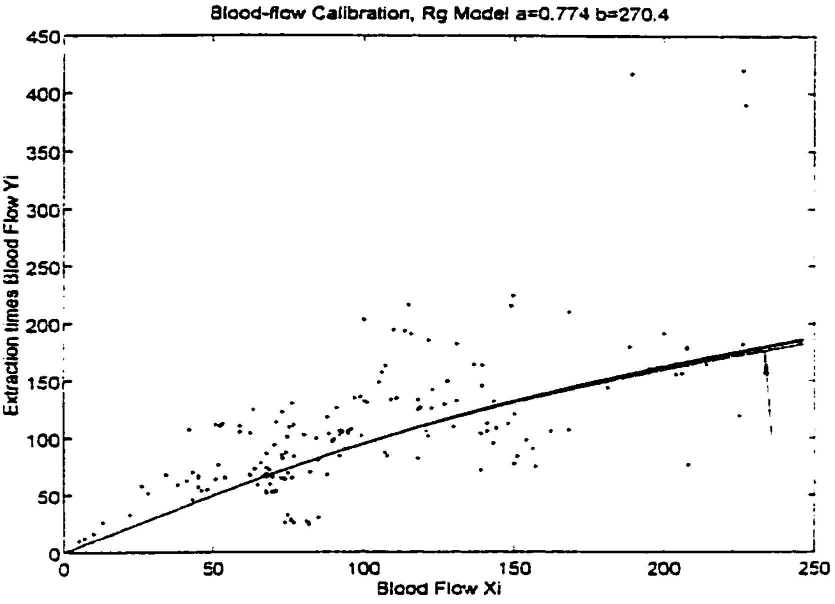


Figure 3.12 Blood-flow Calibration, r_{gd} model

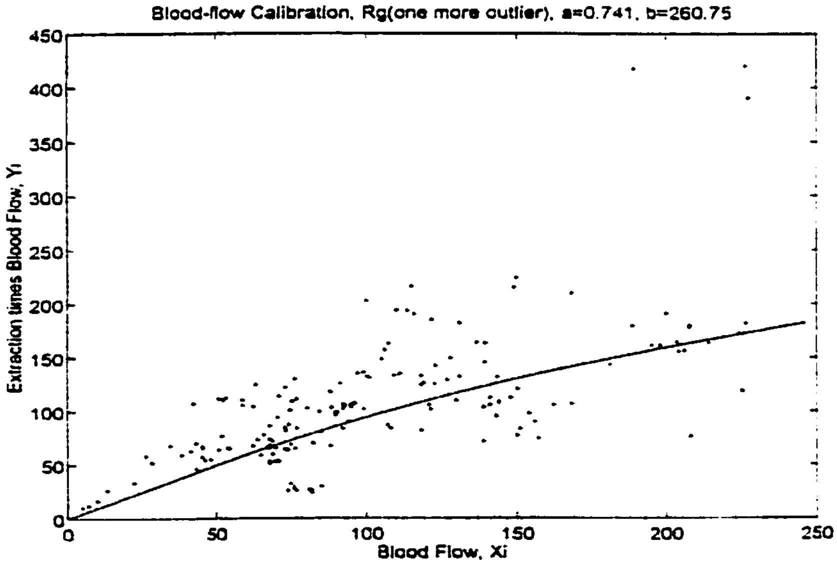


Figure 3.13 Blood-flow Calibration, r_{gd} model (another outlier)

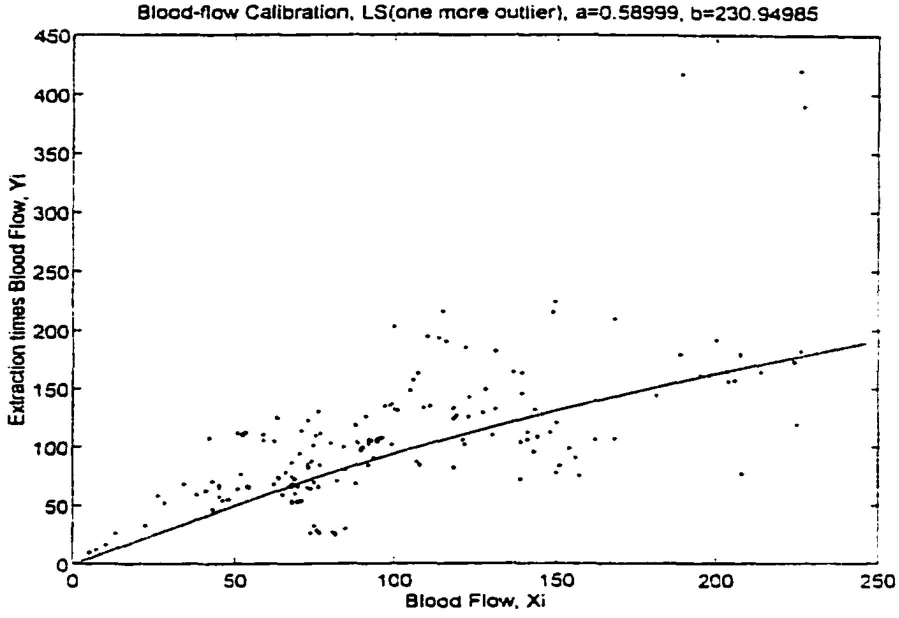


Figure 3.14 Blood-flow Calibration, LS (another outlier)

Figures 3.5, 3.6, 3.7, 3.8, 3.9, 3.10, 3.11, 3.12 shows the Greatest Deviation correlation coefficient r_{gd} estimation method and other methods.

Since the Greatest Deviation r_{gd} is a nonparametric correlation coefficient which is robust to outliers, the estimates arising from its use in nonlinear regression are also resistant to outliers. In this example, suppose that there exists another outlier $X = 934$ (the 251th observation) and $y = 905.7$. Estimation using the r_{gd} method and least squares method gives the following results:

r_{gd} method: $\mu_t = x_t(1 - 0.7410 e^{-260.750/x_t})$.

LS with Newton and steepest descent method:

$$\mu_i = x_i(1-0.58999e^{-230.94985/x_i}).$$

The influence of one more outlier causes α and β to change significantly for least squares with the steepest descent method (see Table 3.5 and Figures 3.13, 3.14).

Table 3.5 Comparison with r_{gd} and least squares

	with 4 outliers	with 5 outliers
r_{gd} method	$\hat{\alpha} = 0.774$ $\hat{\beta} = 270.40$	$\hat{\alpha} = 0.741$ $\hat{\beta} = 260.75$
LS with the steepest descent method	$\hat{\alpha} = 0.7818$ $\hat{\beta} = 360.0$	$\hat{\alpha} = 0.58999$ $\hat{\beta} = 230.950$

C and Splus functions were used to estimate the parameters for these nonlinear models.

Chapter 4

Times Series Model and Estimation

4.1 Introduction

A time series is a collection of random variables, say $\{Y_t\}, t = 1,\ldots,N$, ordered in time. A time series might be observations on economic variables over time, which can originate from various fields of economic and business. Examples of such variables are inflation rates, stock market indices, unemployment rates, and market shares. Forecasts for such variables are often needed to set policy targets. For example, the forecast for the next year's inflation rate can lead to a change in the monetary policy of a central bank. A forecast of a company's market share in the next few months may lead to changes in the allocation of the advertising budget. Time series data can display a wide variety of patterns. Typically, many macroeconomic aggregates such as industrial production. consumption, and wages show upward trending patterns. Stock markets can crash with decreases in daily returns that can be as large as -20%, while such markets do not tend to boom with similarly sized increases in returns.

ARIMA stands for *A*uto*R*egressive *I*ntegrated *M*oving *A*verage. ARIMA models are flexible and widely used models in time series analyses. As a first step in an ARIMA process the raw time series is examined to identify one of the many available models that we will tentatively select as the best representation of the time series. The second step in the process is to estimate the parameters of the tentative model. The third step in the

ARIMA modeling process is to assess the quality of the model in order to determine whether the correct model has been chosen. The final step in the ARIMA modeling process is to actually forecast using the chosen model. Figure 4.1 shows the ARIMA (Box-Jenkins) process.

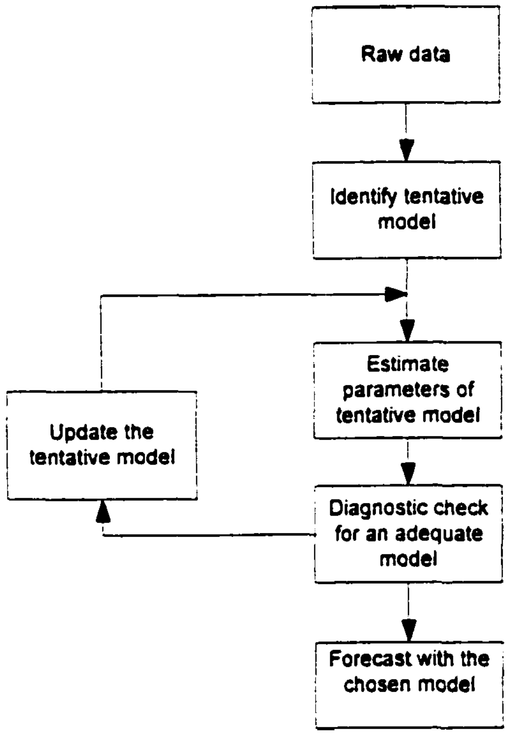

Raw data
Identify tentative model
Estimate parameters of tentative model
Update the tentative model
Diagnostic check for an adequate model
Forecast with the chosen model

Figure 4.1 ARIMA (Box-Jenkins) process

In summary, there are two general goals of time series analysis: (a) identifying the nature of the phenomenon represented by the sequence of observations, and (b) forecasting. Both of these goals require that the pattern of observed time series data is identified and described. Once the pattern is established, we can extrapolate the identified pattern to predict future events.

Most time series patterns can be described in terms of two basic classes of components: trend and seasonality. The trend represents a general systematic linear or nonlinear component that changes over time and does not repeat or at least does not repeat within the time range captured by our data. The seasonality may have a formally similar nature; however, it repeats itself in systematic intervals over time. Those two general classes of time series components may coexist in real-life data. For example, sales of a company can rapidly grow over years but they still follow consistent seasonal patterns. In this chapter, we use several sample series for the illustration of the concepts and models. Some simulation data also appear in this chapter.

In time series analysis, it is assumed that the data consist of a systematic pattern, random noise and possibly outliers which can often make the pattern difficult to identify. We know that the ordinary least squares and maximum likelihood estimation techniques are not robust to outliers, which often lead to specification of the time series model. The estimation process of r_{gd} regression adapts the robustness of the corresponding parametric correlation coefficient. The robust r_{gd} method can ensure that a few outliers have not allowed a misspecification of the time series model. Parameter estimates obtained for several data sets and through simulation show that the r_{gd} method compares favorably with the classical least squares or maximum likelihood estimation methods when the data are well behaved, but performs robustly when the data have numerous suspect data.

In this chapter we will discuss parameter estimation and forecasting using the nonparametric correlation coefficient r_{gd} method.

4.2 ARIMA Model

ARIMA (Box-Jenkins) models are flexible and widely used models in time series analyses. ARIMA models work well for a large variety of time series. The methods used to estimate the parameters of ARIMA models can be computationally intensive.

4.2.1 Expectations, Autocorrelation, and Stationarity

(A) Expectations and Stochastic Processes

Suppose we have observed a sample of size T of some random variable Y_t:

$$\{y_1, y_2, ..., y_T\} \qquad (4.2.1)$$

For example, consider a collection of T independent and identically distributed (i.i.d) variables:

$$\{\varepsilon_1, \varepsilon_2, ..., \varepsilon_T\} \quad \text{where } \varepsilon_t \sim N(0, \sigma^2) \qquad (4.2.2)$$

This is referred to as a sample of size T from a Gaussian white noise process.

Let $\{y_t^{(1)}\}_{t=-\infty}^{\infty}, \{y_t^{(2)}\}_{t=-\infty}^{\infty}, ..., \{y_t^{(I)}\}_{t=-\infty}^{\infty}$ be I sequences and consider selecting the observation associated with date t from each sequence:

$$\{y_t^{(1)}, y_t^{(2)}, ..., y_t^{(I)}\}$$

This would be described as a sample of I realizations of the random variable Y_t. This random variable Y_t has an unconditional density, denoted by $f_{Y_t}(y_t)$.

Under a normality assumption,

$$f_{Y_t}(y_t) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp(\frac{-y_t^2}{2\sigma^2}), \quad -\infty < y_t < \infty,$$

with mean: $$E(Y_t) = \int_{-\infty}^{\infty} y_t f_{Y_t}(y_t) dy_t \,. \tag{4.2.3}$$

If $\{Y_t\}_{t=-\infty}^{\infty}$ represents the sum of a constant μ plus a Gaussian white noise process $\{\varepsilon_t\}_{t=-\infty}^{\infty}$, then the resulting model is:

$$Y_t = \mu + \varepsilon_t \,, \tag{4.2.4}$$

where $$E(Y_t) = \mu + E(\varepsilon_t) = \mu \,. \tag{4.2.5}$$

The variance of the random variable Y_t (denoted γ_{0t}), is defined by

$$\gamma_{0t} = E(Y_t - \mu_t)^2 = \int_{-\infty}^{\infty} (y_t - \mu_t)^2 f_{Y_t}(y_t) dy_t \,. \tag{4.2.6}$$

(B) Autocovariance, Autocorrelation and Partial Autocorrelation

Given a particular realization such as $\{y_t^{(1)}\}_{t=-\infty}^{\infty}$ on a time series process, consider constructing a vector

$$X_t^{(1)} = \begin{bmatrix} y_t^{(1)} \\ y_{t-1}^{(1)} \\ \cdot \\ \cdot \\ \cdot \\ y_{t-j}^{(1)} \end{bmatrix} .$$

The j th autocovariance of Y_t is given by:

$$\gamma_{jt} = E(Y_t - \mu_t)(Y_{t-j} - \mu_{t-j})$$

$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty}(y_t - \mu_t)(y_{t-j} - \mu_{t-j}) \times f_{Y_t,Y_{t-1},\ldots,Y_{t-j}}(y_t, y_{t-1}, \ldots, y_{t-j})dy_t dy_{t-1}\ldots dy_{t-j}\,. \qquad (4.2.7)$$

For the process in (4.2.4) the autocovariances are all zero for $j \neq 0$:

$$\gamma_{jt} = E(Y_t - \mu)(Y_{t-j} - \mu) = E(\varepsilon_t \varepsilon_{t-j}) = 0 \text{ for } j \neq 0$$

The patterns of time series can be examined via the autocorrelation function (ACF) which consists of the serial correlation coefficients for consecutive lags in a specified range of lags. A useful diagnostic is plot of the ACF versus the lag.

The sample autocorrelation function at lag k, r_k, is defined by

$$r_k = \frac{\sum_{t=k+1}^{n}(Y_t - \bar{Y})(Y_{t-k} - \bar{Y})}{\sum_{t=1}^{n}(Y_t - \bar{Y})^2} \quad \text{for } k = 0,1,2,\ldots. \qquad (4.2.8)$$

Another useful method to examine serial dependencies is to examine the partial autocorrelation function (PACF) - an extension of the autocorrelation function, where the dependence on the intermediate elements (those within the lag) is removed. In other words the partial autocorrelation is similar to autocorrelation, except that when calculating it, the autocorrelations with all the elements within the lag are removed (Box & Jenkins, 1976; McDowall, McCleary, Meidinger, & Hay, 1980).

The partial autocorrelation function for lag k, ϕ_{kk}, is defined by

$$\phi_{kk} = Cor(Y_t, Y_{t-k} \mid Y_{t-1}, Y_{t-2}, \ldots, Y_{t-k+1}), \qquad (4.2.9)$$

i.e., ϕ_{kk} is the correlation coefficient in the bivariate distribution of Y_t, Y_{t-k} conditional on $Y_{t-1}, Y_{t-2}, Y_{t-k+1}$.

Levinson (1947) and Durbin (1960) gave an efficient recursive equation for obtaining ϕ_{kk}:

$$\phi_{kk} = \frac{\rho_k - \sum_{j=1}^{k-1} \phi_{k-1,j}\rho_{k-j}}{1 - \sum_{j=1}^{k-1} \phi_{k-1,j}\rho_j} \qquad (4.2.10)$$

where $\phi_{kj} = \phi_{k-1,j} - \phi_{kk}\phi_{k-1,k-j}$ for $j = 1,2,\dots,k-1$.

The recursive begins with $\phi_{11} = \rho_1$.

(C) Stationarity

If neither the mean μ_t nor the autocovariences γ_{jt} dependent on the time t, then the process for Y_t is said to be covariance-stationary or weakly stationary:

$$E(Y_t) = \mu \qquad \text{for all } t,$$

$$E(Y_t - \mu)(Y_{t-j} - \mu) = \gamma_j \qquad \text{for all } t \text{ and any } j.$$

A process is said to be strictly stationary if, for any values of $j_1, j_2, \dots, j_n$, the joint distribution of $(Y_t, Y_{t-j_1}, Y_{t-j_2}, \dots, Y_{t-j_n})$ depends only on the intervals separating the dates $(j_1, j_2, \dots, j_n)$ and not on the time itself (t).

4.2.2 Moving Average Processes

(A) The First-Order Moving Average Process

Let $\{\varepsilon_t\}$ be white noise and consider the process

$$Y_t = \mu + \varepsilon_t + \theta\varepsilon_{t-1}, \qquad (4.2.11)$$

where $\varepsilon_t \sim i.i.d\ N(0,\sigma^2)$

The expectation of Y_t is given by

$$E(Y_t) = E(\mu + \varepsilon_t + \theta\varepsilon_{t-1}) = \mu + E(\varepsilon_t) + \theta E(\varepsilon_{t-1}) = \mu . \qquad (4.2.12)$$

The variance of Y_t is

$$E(Y_t - \mu)^2 = E(\varepsilon_t + \theta\varepsilon_{t-1})^2$$

$$=(1+\theta^2)\sigma^2 . \qquad (4.2.13)$$

The first autocovariance is

$$E(Y_t - \mu)(Y_{t-1} - \mu) = E(\varepsilon_t + \theta\varepsilon_{t-1})(\varepsilon_{t-1} + \theta\varepsilon_{t-2})$$

$$= \theta\sigma^2 . \qquad (4.2.14)$$

Higher autocovariances are all zero:

$$E(Y_t - \mu)(Y_{t-j} - \mu) = E(\varepsilon_t + \theta\varepsilon_{t-1})(\varepsilon_{t-j} + \theta\varepsilon_{t-j-1}) = 0 \quad \text{for } j > 1. \qquad (4.2.15)$$

The j th autocorrelation of a covariance-stationary process is defined as the j th autocovariance divided by the variance:

$\rho_j = \gamma_j / \gamma_0$, resulting from: (4.2.16)

$$Corr(Y_t, Y_{t-j}) = \frac{Cov(Y_t, Y_{t-j})}{\sqrt{Var(Y_t)}\sqrt{Var(Y_{t-j})}} = \frac{\gamma_j}{\sqrt{\gamma_0}\sqrt{\gamma_0}} = \rho_j .$$

The first autocorrelation for the $MA(1)$ process is given by

$$\rho_1 = \frac{\theta\sigma^2}{(1+\theta^2)\sigma^2} = \frac{\theta}{(1+\theta^2)} . \qquad (4.2.17)$$

The method of moments is one of the easiest, if not the most efficient, methods for obtaining parameter estimates in MA models. The method consists of equating sample moments to theoretical moments and solving the resulting equations to obtain estimates of unknown parameters.

Equating ρ_1 to r_1 using equation (4.2.17):

$$r_1 = \frac{\hat{\theta}}{(1+\hat{\theta}^2)}$$

$$\Rightarrow \quad \hat{\theta} = \frac{1 \pm \sqrt{1-4r_1^2}}{2r_1}.$$

Only one of the solutions satisfies the invertibility condition $|\theta| < 1$, namely

$$\hat{\theta} = \frac{1-\sqrt{1-4r_1^2}}{2r_1}. \qquad (4.2.18)$$

If the time series Y_t is nearly Gaussian, then ρ can be estimated robustly using r_{gd}, via $\hat{r} = \sin(\frac{\pi}{2} r_{gd})$. The population relationship for bivariate normal is $\rho = \sin(\frac{\pi}{2}\rho_{gd})$, where $\rho_{gd} = \frac{2}{\pi}\sin^{-1}(\rho)$, the population parameter ρ_{gd} was developed by Gideon et al ([22], 1987). If we use the nonparametric correlation coefficient r_{gd}, then a robust estimate of θ is:

$$\hat{\theta}_{gd} = \frac{1-\sqrt{1-4\sin^2(\frac{\pi}{2} r_{gd})}}{2\sin(\frac{\pi}{2} r_{gd})}. \qquad (4.2.19)$$

Maximum likelihood estimation

Conditional Likelihood Function:

Let $\underline{\theta} = (\mu, \theta, \sigma^2)'$ denote the population parameters to be estimated for the MA(1) model, then:

$$f_{Y_t|\varepsilon_{t-1}}(y_t \mid \varepsilon_{t-1};\underline{\theta}) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left[\frac{-(y_t-\mu-\theta\varepsilon_{t-1})^2}{2\sigma^2}\right]. \qquad (4.2.20)$$

We assume that $\varepsilon_0 = 0$. Then:

$(Y_1 \mid \varepsilon_0 = 0) \sim N(\mu,\sigma^2)$.

Given the observation y_1, the value of ε_1 is then known with certainty as well:

$\varepsilon_1 = y_1 - \mu$, and hence:

$$f_{Y_2|Y_1,\varepsilon_0=0}(y_2 \mid y_1,\varepsilon_0 = 0;\underline{\theta}) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left[\frac{-(y_2-\mu-\theta\varepsilon_1)^2}{2\sigma^2}\right].$$

Since ε_1 is known with certainty, ε_2 can be calculated by:

$\varepsilon_2 = y_2 - \mu - \theta\varepsilon_1$.

Proceeding in this fashion, it is clear that given the initial knowledge that $\varepsilon_0 = 0$, the full sequence $\{\varepsilon_1,\varepsilon_2,...,\varepsilon_T\}$ can be calculated from $\{y_1,y_2,...,y_T\}$ by iterating

$$\varepsilon_t = y_t - \mu - \theta\varepsilon_{t-1}. \qquad (4.2.21)$$

for $t = 1,2,...,T$, starting from $\varepsilon_0 = 0$. The conditional density of the t th observation can be calculated from (4.2.20) as

$$f_{Y_t|Y_{t-1},Y_{t-2},...,Y_1,\varepsilon_0=0}(y_t \mid y_{t-1},y_{t-2},...,y_1,\varepsilon_0 = 0;\underline{\theta})$$

$$= f_{Y_t|\varepsilon_{t-1}}(y_t \mid \varepsilon_{t-1};\underline{\theta})$$

$$= \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left[\frac{-\varepsilon_t^2}{2\sigma^2}\right]. \qquad (4.2.22)$$

The sample likelihood would then be the product of these individual densities:

$$f_{Y_T,Y_{T-1},\dots,Y_1|\varepsilon_0=0}(y_T, y_{T-1},\dots,y_1 \mid \varepsilon_0 = 0;\underline{\theta})$$

$$= f_{Y_1|\varepsilon_0=0}(y_1 \mid \varepsilon_0 = 0;\theta)\prod_{t=2}^{T} f_{Y_t|Y_{t-1},\dots,Y_1,\varepsilon_0=0}(y_t \mid y_{t-1}, y_{t-2},\dots,y_1,\varepsilon_0 = 0;\underline{\theta}).$$

The conditional log likelihood is:

$$\ell(\underline{\theta}) = \log f_{Y_T,Y_{T-1},\dots,Y_1|\varepsilon_0=0}(y_T, y_{T-1},\dots,y_1 \mid \varepsilon_0 = 0;\underline{\theta}) \qquad (4.2.23)$$

$$= -\frac{T}{2}\log(2\pi) - \frac{T}{2}\log(\sigma^2) - \sum_{t=1}^{T}\frac{\varepsilon_t^2}{2\sigma^2}.$$

For a particular numerical value of $\underline{\theta}$, we thus calculate the sequence of ε's implied by the data from (4.2.21). The conditional log likelihood (4.2.23) is then a function of the sum of squares of these ε's. The log likelihood is a fairly complicated nonlinear function of μ and θ, so that an analytical expression for the maximum likelihood estimates of μ and θ is not readily calculated. We can use numerical optimization methods to find the value of $\hat{\underline{\theta}}$ that maximizes $\ell(\underline{\theta})$.

(B) The q th-Order Moving Average Process

A q th-order moving average process, $MA(q)$ is given by

$$Y_t = \mu + \varepsilon_t + \theta_1\varepsilon_{t-1} + \theta_2\varepsilon_{t-2} + \dots + \theta_q\varepsilon_{t-q} \qquad (4.2.24)$$

with mean:

$$E(Y_t) = E(\mu + \varepsilon_t + \theta_1\varepsilon_{t-1} + \theta_2\varepsilon_{t-2} + \dots + \theta_q\varepsilon_{t-q})$$

$$= \mu .$$

The variance of Y_t in an $MA(q)$ process is:

$$\gamma_0 = E(Y_t - \mu)^2$$

$$= E(\varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \ldots + \theta_q \varepsilon_{t-q})^2 \tag{4.2.25}$$

$$= (1 + \theta_1{}^2 + \theta_2{}^2 + \ldots + \theta_q{}^2)\sigma^2 . \tag{4.2.26}$$

For $j = 1,2,\ldots,q$, the covariances are given as:

$$\gamma_j = E[(\varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \ldots + \theta_q \varepsilon_{t-q})$$

$$\times (\varepsilon_{t-j} + \theta_1 \varepsilon_{t-j-1} + \theta_2 \varepsilon_{t-j-2} + \ldots + \theta_q \varepsilon_{t-j-q})]$$

$$= E[\theta_j \varepsilon_{t-j}^2 + \theta_{j+1}\theta_1 \varepsilon_{t-j-1}^2 + \theta_{j+2}\theta_2 \varepsilon_{t-j-2}^2 + \ldots, + \theta_q \theta_{q-j} \varepsilon_{t-q}^2] \tag{4.2.27}$$

$$\gamma_j = \begin{cases} [\theta_j + \theta_{j+1}\theta_1 + \theta_{j+2}\theta_2 + \ldots + \theta_q \theta_{q-j}] \cdot \sigma^2 & \text{for} \quad j = 1,2,\ldots,q \\ 0 & \text{for} \quad j > q \end{cases} . \tag{4.2.28}$$

For an $MA(2)$ process, the variance and covariances are:

$$\begin{aligned} \gamma_0 &= [1 + \theta_1^2 + \theta_2^2]\sigma^2 \\ \gamma_1 &= [\theta_1 + \theta_2\theta_1]\sigma^2 \\ \gamma_2 &= [\theta_2]\sigma^2 \\ \gamma_3 &= \gamma_4 = \ldots = 0 \end{aligned} \tag{4.2.29}$$

with autocorrelations:

$$\begin{aligned} \rho_1 &= \frac{\theta_1 + \theta_1\theta_2}{1 + \theta_1^2 + \theta_2^2}, \\ \rho_2 &= \frac{\theta_2}{1 + \theta_1^2 + \theta_2^2}, \ \rho_3 = \rho_4 = \ldots = 0. \end{aligned} \tag{4.2.30}$$

For any values of $(\theta_1, \theta_2, \ldots, \theta_q)$, the $MA(q)$ process is thus covariance-stationary.

Assume that

$$\varepsilon_0 = \varepsilon_{-1} = \ldots = \varepsilon_{-q+1} = 0 . \tag{4.2.31}$$

From these starting values we can iterate on

$$\varepsilon_t = y_t - \mu - \theta_1 \varepsilon_{t-1} - \theta_2 \varepsilon_{t-2} - \ldots - \theta_q \varepsilon_{t-q} \quad \text{for } t = 1,2,\ldots,T.$$

Let $\underline{\varepsilon_0}$ denote the $(q \times 1)$ vector $(\varepsilon_0, \varepsilon_{-1}, \ldots, \varepsilon_{-q+1})'$.

The conditional log likelihood is then

$$\ell(\underline{\theta}) = \log f_{Y_T, Y_{T-1}, \ldots, Y_1 | \varepsilon_0 = 0}(y_T, y_{T-1}, \ldots, y_1 \mid \underline{\varepsilon_0} = 0; \underline{\theta})$$

$$= -\frac{T}{2}\log(2\pi) - \frac{T}{2}\log(\sigma^2) - \sum_{t=1}^{T} \frac{\varepsilon_t^2}{2\sigma^2} \tag{4.2.32}$$

where $\underline{\theta} = (\mu, \theta_1, \theta_2, \ldots, \theta_q, \sigma^2)'$.

The log likelihood is again a fairly complicated nonlinear function of $\underline{\theta}$, so that an analytical expression for the maximum likelihood estimates of $\underline{\theta}$ is not readily calculated. We also can use a numerical optimization method to find the value of $\hat{\underline{\theta}}$ that maximizes $\ell(\underline{\theta})$.

4.2.3 Autoregressive Processes

(A) The First-Order Autoregressive Process

A first-order autoregression satisfies the following difference equation:

$$Y_t = c + \phi Y_{t-1} + \varepsilon_t, \tag{4.2.33}$$

where $\varepsilon_t \sim i.i.d\ N(0, \sigma^2)$

In the case where $|\phi| < 1$, this is a stationary process for Y_t.

Repeated substitution using (4.2.33) yields:

$$Y_t = (c + \varepsilon_t) + \phi \cdot (c + \varepsilon_{t-1}) + \phi^2 \cdot (c + \varepsilon_{t-2}) + \phi^3 \cdot (c + \varepsilon_{t-3}) + \ldots$$

$$= [c/(1-\phi)] + \varepsilon_t + \phi \varepsilon_{t-1} + \phi^2 \varepsilon_{t-2} + \phi^3 \varepsilon_{t-3} + \ldots \,. \tag{4.2.34}$$

Taking expectations using (4.2.34)

$$E(Y_t) = [c/(1-\phi) + 0 + 0 + ...]$$

so that the mean of an $AR(1)$ process is

$$\mu = c/(1-\phi). \tag{4.2.35}$$

The variance is

$$\begin{aligned}\gamma_0 &= E(Y_t - \mu)^2 \\ &= E(\varepsilon_t + \phi\varepsilon_{t-1} + \phi^2\varepsilon_{t-2} + \phi^3\varepsilon_{t-3} + ...)^2 \\ &= (1 + \phi^2 + \phi^4 + \phi^6 + ...)\cdot\sigma^2 \\ &= \sigma^2/(1-\phi^2),\end{aligned} \tag{4.2.36}$$

while the j th autocovariance is

$$\begin{aligned}\gamma_j &= E(Y_t - \mu)(Y_{t-j} - \mu) \\ &= E[\varepsilon_t + \phi\varepsilon_{t-1} + \phi^2\varepsilon_{t-2} + ... + \phi^j\varepsilon_{t-j} + \phi^{j+1}\varepsilon_{t-j-1} + ...] \\ &\qquad \times[\varepsilon_{t-j} + \phi\varepsilon_{t-j-1} + \phi^2\varepsilon_{t-j-2} + ..] \\ &= [\phi^j + \phi^{j+2} + \phi^{j+4} + ...]\cdot\sigma^2 \\ &= [\phi^j/(1-\phi^2)]\cdot\sigma^2.\end{aligned} \tag{4.2.37}$$

It follows from (4.2.36) and (4.2.37) that the autocorrelation function follows a pattern of geometric decay:

$$\rho_j = \gamma_j/\gamma_0 = \phi^j, \quad j = 0,1,....$$

Maximum Likelihood Estimates for the Gaussian AR(1) Process:

The primary principle on which estimation will be based is maximum likelihood.

Let $\underline{\theta} = (c, \phi, \sigma^2)'$. (4.2.38)

Suppose we have observed a sample of size $T\ (y_1, y_2, \ldots, y_T)$. The approach will be to calculate the joint probability density:

$$f_{Y_T, Y_{T-1}, \ldots, Y_1}(y_T, y_{T-1}, \ldots, y_1; \underline{\theta}). \qquad (4.2.39)$$

The maximum likelihood estimator (MLE) of $\underline{\theta}$ is the value for which this sample is most likely to have been observed, i.e., it is the value of $\underline{\theta}$ that maximizes (4.2.39).

Consider the probability distribution of Y_1, the first observation in the sample.

From (4.2.35) and (4.2.36), $E(Y_1) = \mu = c/(1-\phi)$ and $\gamma_0 = E(Y_1 - \mu)^2 = \sigma^2/(1-\phi^2)$.

The density of the first observation is

$$f_{Y_1}(y_1; \underline{\theta}) = f_{Y_1}(y_1; c, \phi, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2/(1-\phi^2)}} \exp\left[\frac{-\{y_1 - [c/(1-\phi)]\}^2}{2\sigma^2/(1-\phi^2)}\right]. \qquad (4.2.40)$$

Consider the distribution of the second observation Y_2 conditional on observing $Y_1 = y_1$.

From (4.2.33),

$$Y_2 = c + \phi Y_1 + \varepsilon_2, \text{ so that} \qquad (4.2.41)$$

$$f_{Y_2|Y_1}(y_2 \mid y_1; \underline{\theta}) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[\frac{-(y_2 - c - \phi y_1)^2}{2\sigma^2}\right]. \qquad (4.2.42)$$

In general,

$$f_{Y_t|Y_{t-1}, Y_{t-2}, \ldots, Y_1}(y_t \mid y_{t-1}, y_{t-2}, \ldots, y_1; \underline{\theta}) = f_{Y_t|Y_{t-1}}(y_t \mid y_{t-1}; \underline{\theta}) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[\frac{-(y_t - c - \phi y_{t-1})^2}{2\sigma^2}\right]$$

(4.2.43)

The joint density of the first t observations is then

$$f_{Y_t,Y_{t-1},Y_{t-2},\ldots,Y_1}(y_t, y_{t-1}, y_{t-2},\ldots, y_1;\underline{\theta})$$

$$= f_{Y_t|Y_{t-1}}(y_t \mid y_{t-1};\theta)\cdot f_{Y_{t-1},Y_{t-2},\ldots,Y_1}(y_{t-}, y_{t-2},\ldots, y_1;\underline{\theta})\,. \qquad (4.2.44)$$

The likelihood of the complete sample (sample size T) can thus be calculated as

$$f_{Y_T,Y_{T-1},Y_{T-2},\ldots,Y_1}(y_t, y_{t-1}, y_{t-2},\ldots, y_1;\underline{\theta}) = f_{Y_1}(y_1;\theta)\prod_{t=2}^{T} f_{Y_t|Y_{t-1}}(y_t \mid y_{t-1};\underline{\theta}) \qquad (4.2.45)$$

The log likelihood function is thus:

$$\ell(\theta) = \log f_{Y_1}(y_1;\underline{\theta}) + \sum\nolimits_{t=2}^{T} \log f_{Y_t|Y_{t-1}}(y_t \mid y_{t-1};\underline{\theta})\,. \qquad (4.2.46)$$

Substituting (4.2.40) and (4.2.43) into (4.2.46), the log likelihood for a sample size T from a Gaussian AR(1) process is:

$$\ell(\underline{\theta}) = -\frac{1}{2}\log(2\pi) - \frac{1}{2}\log[\sigma^2/(1-\phi^2)] - \frac{\{y_1 - [c/(1-\phi)]\}^2}{2\sigma^2/(1-\phi^2)} - [(T-1)/2]\log(2\pi)$$

$$-[(T-1)/2]\log(\sigma^2) - \sum\nolimits_{t=2}^{T}\left[\frac{(y_t - c - \phi y_{t-1})^2}{2\sigma^2}\right]. \qquad (4.2.47)$$

The *MLE* $\hat{\underline{\theta}}$ is the value for which (4.2.47) is maximized. In principle, this requires differentiating (4.2.47) and setting the result equal to zero. Maximization of (4.2.47) requires iterative or numerical procedures. An alternative to numerical maximization of the exact likelihood function is to regard the value of y_1 as deterministic and maximize the likelihood conditioned on the first observation.

The joint density of $Y_t,\ldots,Y_2$ given Y_1 is:

$$f_{Y_t,Y_{t-1},\ldots,Y_2|Y_1}(y_T, y_{T-1},\ldots, y_2 \mid y_1;\underline{\theta}) = \prod_{t=2}^{T} f_{Y_t|Y_{t-1}}(y_t \mid y_{t-1};\underline{\theta}) \qquad (4.2.48)$$

The loglikelihood of $\underline{\theta}$ is thus:

$$\log f_{Y_t,Y_{t-1},\ldots,Y_2|Y_1}(y_T,y_{T-1},\ldots,y_2 \mid y_1;\underline{\theta}) = \log\prod_{t=2}^{T} f_{Y_t|Y_{t-1}}(y_t \mid y_{t-1};\underline{\theta})$$

$$= -[(T-1)/2]\log(2\pi)-[(T-1)/2]\log(\sigma^2)-\sum_{t=2}^{T}\left[\frac{(y_t-c-\phi y_{t-1})^2}{2\sigma^2}\right]. \qquad (4.2.49)$$

Maximization of (4.2.49) with respect to c and ϕ is equivalent to minimization of

$$\sum_{t=2}^{T}(y_t-c-\phi y_{t-1})^2 .$$

The conditional maximum likelihood estimates of c and ϕ are given by

$$\begin{bmatrix}\hat{c}\\ \hat{\phi}\end{bmatrix}=\begin{bmatrix}T-1 & \Sigma y_{t-1}\\ \Sigma y_{t-1} & \Sigma y_{t-1}^2\end{bmatrix}^{-1}\begin{bmatrix}\Sigma y_t\\ \Sigma y_{t-1}y_t\end{bmatrix}$$

Let $\quad \underline{\hat{\beta}}=\begin{bmatrix}\hat{c}\\ \hat{\phi}\end{bmatrix},\ X=\begin{bmatrix}T-1 & \Sigma y_{t-1}\\ \Sigma y_{t-1} & \Sigma y_{t-1}^2\end{bmatrix},\ \underline{z}=\begin{bmatrix}\Sigma y_t\\ \Sigma y_{t-1}y_t\end{bmatrix}$

Therefore, $\quad X\underline{\hat{\beta}}=\underline{z}$,

and $\quad X^T X\underline{\hat{\beta}}=X^T\underline{z}$.

These normal equations can be solved by the correlation method using Pearson's r or r_{gd} by solving $b^{(m)}$ in the following equation for r or r_{gd}:

$$r(\underline{x}_i,\underline{z}-X\underline{b}^{(m)})=0 \text{ or } r_{gd}(\underline{x}_i,\underline{z}-X\underline{b}^{(m)})=0, \quad \textit{for } i=1,2. \qquad (4.2.50)$$

(B) The Second-Order Autoregressive Process

A second-order autoregression model, denoted $AR(2)$, satisfies:

$$Y_t=c+\phi_1 Y_{t-1}+\phi_2 Y_{t-2}+\varepsilon_t \qquad (4.2.51)$$

Taking the expectation of (4.2.51):

$$E(Y_t) = c + \phi_1 E(Y_{t-1}) + \phi_2 E(Y_{t-2}) + E(\varepsilon_t) \tag{4.2.52}$$

implying that

$$\mu = c + \phi_1 \mu + \phi_2 \mu + 0$$

$$\Rightarrow \qquad \mu = c/(1 - \phi_1 - \phi_2).$$

To find the second moments, write (4.2.51) as

$$(Y_t - \mu) = \phi_1 (Y_{t-1} - \mu) + \phi_2 (Y_{t-2} - \mu) + \varepsilon_t. \tag{4.2.53}$$

Multiply both sides by $(Y_{t-j} - \mu)$ and take expectations to give:

$$\gamma_j = \phi_1 \gamma_{j-1} + \phi_2 \gamma_{j-2} \quad \text{for } j = 1,2,\ldots. \tag{4.2.54}$$

The autocorrelations are then found by dividing both sides of (4.2.54) by γ_0:

$$\rho_j = \phi_1 \rho_{j-1} + \phi_2 \rho_{j-2} \quad \text{for } j = 1,2,\ldots. \tag{4.2.55}$$

For $j=1$, $\rho_1 = \phi_1 / (1 - \phi_2)$.

For $j=2$, $\rho_2 = \phi_1 \rho_1 + \phi_2$

$$= \frac{\phi_2 (1 - \phi_2) + \phi_1^2}{1 - \phi_2}. \tag{4.2.56}$$

The method of moments replaces ρ_1 by r_1 and ρ_2 by r_2 to obtain

$$\hat{\phi}_1 = \frac{r_1 (1 - r_2)}{1 - r_1^2} \text{ and } \hat{\phi}_2 = \frac{r_2 - r_1^2}{1 - r_1^2} \tag{4.2.57}$$

If the time series is Gaussian, then $\hat{r} = \sin(\frac{\pi}{2} r_{gd})$.(see Gideon, 1987). Using the nonparametric correlation coefficient r_{gd}, the estimates are:

$$\hat{\phi}_1 = \frac{\sin(\frac{\pi}{2} r_{gd1})[1-\sin(\frac{\pi}{2} r_{gd2})]}{1-\sin^2(\frac{\pi}{2} r_{gd1})}, \quad \hat{\phi}_2 = \frac{\sin(\frac{\pi}{2} r_{gd2})-\sin^2(\frac{\pi}{2} r_{gd1})]}{1-\sin^2(\frac{\pi}{2} r_{gd1})}. \qquad (4.2.58)$$

(C) The p th-Order Autoregressive Process

A p th-order moving average process, $AR(p)$, is given by

$$Y_t = c + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \ldots + \phi_p Y_{t-p} + \varepsilon_t \qquad (4.2.59)$$

with $\varepsilon_t \sim i.i.d.\ N(0,\sigma^2)$.

The Yule-Walker equations:

$$\rho_j = \phi_1 \rho_{j-1} + \phi_2 \rho_{j-2} + \ldots + \phi_p \rho_{j-p} \quad \text{for } j = 1,2,\ldots. \qquad (4.2.60)$$

Replace ρ_k by r_k in the Yule-Walker equations are:

$$\begin{cases} r_1 = \hat{\phi}_1 + r_1\hat{\phi}_2 + \ldots + r_{p-1}\hat{\phi}_p \\ r_2 = r_1\hat{\phi}_1 + \hat{\phi}_2 + \ldots + r_{p-2}\hat{\phi}_p \\ \cdot \\ \cdot \\ \cdot \\ r_p = r_{p-1}\hat{\phi}_1 + r_{p-2}\hat{\phi}_2 + \ldots + \hat{\phi}_p \end{cases} \qquad (4.2.61)$$

The above linear equations can be solved for $(\hat{\phi}_1, \hat{\phi}_2, \ldots, \hat{\phi}_p)$ in terms of $(r_1, r_2, \ldots, r_p)$.

If we replace r_k by $\sin(\frac{\pi}{2} r_{gdk})$, $k = 1,..,p$ in (4.2.61), we can obtain the r_{gd}-based estimator for ϕ. These linear equations are solved for $(\hat{\phi}_1, \hat{\phi}_2, \ldots, \hat{\phi}_p)$ in terms of $(r_{gd1}, r_{gd2}, \ldots, r_{gdp})$.

Maximum Likelihood Estimation:

Let $\underline{\theta} = (c, \phi_1, \phi_2, \ldots, \phi_p, \sigma^2)'$.

Let $\sigma^2 V_p$ be the $(p \times p)$ variance-covariance matrix of $(Y_1, Y_2, ..., Y_p)$:

$$\sigma^2 V_p = \begin{bmatrix} E(Y_1-\mu)^2 & E(Y_1-\mu)(Y_2-\mu) & \dots & E(Y_1-\mu)(Y_p-\mu) \\ E(Y_1-\mu)(Y_2-\mu) & E(Y_2-\mu)^2 & \dots & E(Y_2-\mu)(Y_p-\mu) \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & & \cdot \\ E(Y_1-\mu)(Y_p-\mu) & E(Y_2-\mu)(Y_p-\mu) & \dots & E(Y_p-\mu)^2 \end{bmatrix} \qquad (4.2.62)$$

The density of the first p observations is that of a $N(\underline{\mu}_p, \sigma^2 V_p)$ variable.

$$f_{Y_p, Y_{p-1}, Y_{p-2}, \dots, Y_1}(y_p, y_{p-1}, y_{p-2}, \dots, y_1; \theta)$$

$$= (2\pi)^{-p/2} \left|\sigma^{-2} V_p^{-1}\right|^{1/2} \exp\left[-\frac{1}{2\sigma^2}(\underline{y_p} - \underline{\mu_p})' V_p^{-1} (\underline{y_p} - \underline{\mu_p})\right]$$

$$= (2\pi)^{-p/2} (\sigma^{-2})^{p/2} \left|V_p^{-1}\right|^{1/2} \exp\left[-\frac{1}{2\sigma^2}(\underline{y_p} - \underline{\mu_p})' V_p^{-1} (\underline{y_p} - \underline{\mu_p})\right]. \qquad (4.2.63)$$

For the remaining observations in the sample, $(y_{p+1}, y_{p+2}, ..., y_T)$, the prediction-error decomposition can be used. Conditional on the first $t-1$ observations, the t th observation is Gaussian with mean $c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + ... + \phi_p y_{t-p}$ and variance σ^2. Only the p most recent observations matter for this distribution.

Hence for $t > p$,

$$f_{Y_t|Y_{t-1}, Y_{t-2}, \dots, Y_1}(y_t \mid y_{t-1}, y_{t-2}, \dots, y_1; \underline{\theta})$$

$$= f_{Y_t|Y_{t-1}, Y_{t-2}, \dots, Y_{t-p}}(y_t \mid y_{t-1}, y_{t-2}, \dots, y_{t-p}; \underline{\theta})$$

$$= \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[\frac{-(y_t - c - \phi_1 y_{t-1} - \phi_2 y_{t-2} - \dots - \phi_p y_{t-p})^2}{2\sigma^2}\right].$$

The likelihood function for the complete sample is then:

$$f_{Y_T,Y_{T-1},Y_{T-2},\ldots,Y_1}(y_t, y_{t-1}, y_{t-2},\ldots,y_1;\underline{\theta})$$

$$= f_{Y_p,Y_{p-1},\ldots,Y_1}(y_p, y_{p-1},\ldots,y_1;\theta) \times \prod_{t=p+1}^{T} f_{Y_t|Y_{t-1},\ldots,Y_{t-p}}(y_t \mid y_{t-1}, y_{t-2},\ldots y_{t-p};\theta). \qquad (4.2.64)$$

The log likelihood function is:

$$\ell(\theta) = -\frac{T}{2}\log(2\pi) - \frac{T}{2}\log(\sigma^2) + \frac{1}{2}\log\left|V_p^{-1}\right| - \frac{1}{2\sigma^2}(\underline{y}_p - \underline{\mu}_p)'V_p^{-1}(\underline{y}_p - \underline{\mu}_p)$$

$$-\sum_{t=p+1}^{T}\frac{(y_t - c - \phi_1 y_{t-1} - \phi_2 y_{t-2} - \ldots - \phi_p y_{t-p})^2}{2\sigma^2}. \qquad (4.2.65)$$

Conditional Maximum Likelihood Estimates:

Maximization of the exact log likelihood for an AR(p) process (4.2.65) must be accomplished numerically. In contrast, the log of the likelihood conditional on the first p observations assumes the simpler form:

$$\log f_{Y_T,Y_{T-1},\ldots,Y_{p+1}|Y_p,\ldots,Y_1}(y_T, y_{T-1},\ldots,y_{p+1} \mid y_p,\ldots,y_1;\underline{\theta})$$

$$= -\frac{T-p}{2}\log(2\pi) - \frac{T-p}{2}\log(\sigma^2) - \sum_{t=p+1}^{T}\frac{(y_t - c - \phi_1 y_{t-1} - \phi_2 y_{t-2} - \ldots - \phi_p y_{t-p})^2}{2\sigma^2}. \qquad (4.2.66)$$

The values of $c, \phi_1, \phi_2, \ldots, \phi_p$ are the same as those that minimize

$$\sum_{t=p+1}^{T}(y_t - c - \phi_1 y_{t-1} - \phi_2 y_{t-2} - \ldots - \phi_p y_{t-p})^2 .$$

The conditional maximum likelihood estimates of these parameters can be obtained from an OLS regression of y_t on a constant and p of its own lagged values.

It is easy to see that the least squares estimators of $c, \phi_1, \phi_2, \ldots, \phi_p$, say $\hat{\underline{\phi}}$ can be obtained by solving the following equations:

$$\begin{aligned} &r(y_{t-1}, z - X\hat{\underline{\phi}}) = 0 \\ &r(y_{t-2}, z - X\hat{\underline{\phi}}) = 0 \\ &\qquad\quad\cdot \\ &\qquad\quad\cdot \\ &\qquad\quad\cdot \\ &r(y_{t-p}, z - X\hat{\underline{\phi}}) = 0 \end{aligned} \tag{4.2.67}$$

where $z = y_t$ and $X = (1, y_{t-1}, y_{t-2}, \ldots, y_{t-p})'$, $\hat{\underline{\phi}} = (c, \phi_1, \phi_2, \ldots, \phi_p)'$ and r stands for Pearson's correlation coefficient.

Replacing r in these equations by the nonparametric correlation coefficient r_{gd}, an r_{gd}-based fit can be obtained for the time series parameters:

$$\begin{aligned} &r(y_{t-1}, z - X\hat{\underline{\phi}}) = 0 \\ &r(y_{t-2}, z - X\hat{\underline{\phi}}) = 0 \\ &\qquad\quad\cdot \\ &\qquad\quad\cdot \\ &\qquad\quad\cdot \\ &r(y_{t-p}, z - X\hat{\underline{\phi}}) = 0. \end{aligned} \tag{4.2.68}$$

4.2.4 Mixed Autoregressive Moving Average Process

An $ARMA(p,q)$ process includes both autoregressive and moving average terms:

$$Y_t = c + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \ldots + \phi_p Y_{t-p} + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \ldots + \theta_q \varepsilon_{t-q} \tag{4.2.69}$$

where $\varepsilon_t \sim i.i.d\ N(0, \sigma^2)$.

First, consider an $ARMA(1,1)$ process:

$$Y_t = \phi Y_{t-1} + \varepsilon_t + \theta\varepsilon_{t-1}, \quad \text{where } \varepsilon_t \sim i.i.d\; N(0,\sigma^2). \qquad (4.2.70)$$

If we multiply both sides of (4.2.70) by Y_{t-k} and take expectations, we have:

$$\gamma_0 = \phi\gamma_1 + E(\varepsilon_t Y_t) + \theta E(\varepsilon_{t-1}Y_t), \qquad \text{for } k=0$$

$$\Rightarrow \quad \gamma_0 = \phi\gamma_1 + [1+\theta(\phi+\theta)]\sigma^2 \qquad \text{for } k=0$$

$$\gamma_1 = \phi\gamma_0 + \theta\sigma^2 \qquad \text{for } k=1$$

$$\gamma_k = \phi\gamma_{k-1} \qquad \text{for } k\geq 2$$

$$\Rightarrow \quad \gamma_0 = \frac{(1+2\theta\phi+\theta^2)}{1-\phi^2}\sigma^2,$$

$$\gamma_k = \frac{(1+\theta\phi)(\phi+\theta)}{1-\phi^2}\phi^{k-1}\sigma^2 \qquad \text{for } k\geq 1,$$

$$\text{and} \quad \rho_k = \frac{(1+\theta\phi)(\phi+\theta)}{1+2\theta\phi+\theta^2}\phi^{k-1} \qquad \text{for } k\geq 1. \qquad (4.2.71)$$

For the r_{gd} method, since $\hat{r} = \sin(\frac{\pi}{2}r_{gd})$, noting that ${}^{\rho_2}\!/_{\rho_1} = \phi$, we can first estimate ϕ as:

$$\hat{\phi} = \frac{r_2}{r_1} \Rightarrow \hat{\phi} = \frac{\sin(\frac{\pi}{2}r_{gd2})}{\sin(\frac{\pi}{2}r_{gd1})}.$$

$$\text{We can solve } \sin(\frac{\pi}{2}r_{gd1}) = \frac{(1+\theta\hat{\phi})(\hat{\phi}+\theta)}{1+2\theta\hat{\phi}+\theta^2}\hat{\phi}^{k-1} \text{ for } \hat{\theta}. \qquad (4.2.72)$$

Maximum Likelihood Estimation (ARMA(p,q) Process):

Let $\quad \underline{\theta} = (c,\phi_1,\phi_2,\ldots,\phi_p,\theta_1,\theta_2,\ldots,\theta_q,\sigma^2)'$.

Taking initial values for $\underline{y_0} \equiv (y_0, y_{-1}, \dots, y_{-p+1})'$ and $\underline{\varepsilon_0} \equiv (\varepsilon_0, \varepsilon_{-1}, \dots, \varepsilon_{-q+1})'$, the sequence $\{\varepsilon_1, \varepsilon_2, \dots, \varepsilon_T)$ can be calculated from $\{y_1, y_2, \dots, y_T\}$ by iterating on

$$\varepsilon_t = y_t - c - \phi_1 y_{t-1} - \phi_2 y_{t-2} - \dots - \phi_p y_{t-p} - \theta_1 \varepsilon_{t-1} - \theta_2 \varepsilon_{t-2} - .. - \theta_q \varepsilon_{t-q} \tag{4.2.73}$$

for $t = 1, 2, \dots, T$.

The conditional log likelihood is

$$\ell(\theta) = \log f_{Y_T, Y_{T-1}, \dots Y_1 | Y_0, \varepsilon_0}(y_T, y_{T-1}, \dots, y_1 \mid y_0, \varepsilon_0; \theta)$$

$$= -\frac{T}{2}\log(2\pi) - \frac{T}{2}\log(\sigma^2) - \sum_{t=1}^{T} \frac{\varepsilon_t^2}{2\sigma^2}. \tag{4.2.74}$$

Box and Jenkins (1976) recommended setting the ε's to zero but the y's equal to their actual

values. Thus iteration on (4.2.73) is started at date $t = p+1$ with $y_1, y_2, \dots, y_p$ set to the observed values and $\varepsilon_p = \varepsilon_{p-1} = \dots = \varepsilon_{p-q+1} = 0$.

Then the conditional log likelihood of $y_T, \dots, y_{p+1}$ is:

$$\ell(\underline{\theta}) = \log f(y_T, \dots, y_{p+1} \mid y_p, \dots, y_1, \varepsilon_p = 0, \dots, \varepsilon_{p-q+1} = 0)$$

$$= -\frac{T-p}{2}\log(2\pi) - \frac{T-p}{2}\log(\sigma^2) - \sum_{t=p+1}^{T} \frac{\varepsilon_t^2}{2\sigma^2}.$$

The above conditional log likelihood is a fairly complicated nonlinear function of $\underline{\theta} = (\theta_1, \dots, \theta_q)$, so that an analytical expression for the maximum likelihood estimates of $\underline{\theta}$ is not readily calculated. We thus require a numerical optimization method to find the value of $\hat{\underline{\theta}}$ that maximizes $\ell(\underline{\theta})$.

4.3 Forecasting

Forecasting the future values of an observed time series is an important problem in many areas, including economics, production planning, sales forecasting and stock control.

(A) *AR*(1) Case

We first illustrate many of the ideas pertaining to forecasting with an AR(1) process with nonzero mean satisfies

$$Y_t - \mu = \phi(Y_{t-1} - \mu) + \varepsilon_t . \tag{4.3.1}$$

Consider the problem of forecasting 1 time unit into the future.

Replacing t by $t+1$ in (4.3.1), we have

$$Y_{t+1} - \mu = \phi(Y_t - \mu) + \varepsilon_{t+1} \tag{4.3.2}$$

Given $Y_t, Y_{t-1}, \ldots, Y_1$, we take the conditional expectation of both sides of (4.5.2)

and obtain $\hat{Y}_{t+1}(t+1) - \mu = \phi[E(Y_t \mid Y_t, Y_{t-1}, \ldots, Y_1) - \mu] + E(\varepsilon_{t+1} \mid Y_t, Y_{t-1}, \ldots, Y_1)$ (4.3.3)

In general, the term $\hat{Y}_{t+\delta}(t+\delta)$ indicates a forecasted value δ time units into the future from time t.

From a property of conditional expectation, we have that

$$E(Y_t \mid Y_t, Y_{t-1}, \ldots, Y_1) = Y_t . \tag{4.3.4}$$

Since ε_{t+1} is independent of $Y_t, Y_{t-1}, \ldots, Y_1$, we obtain

$$E(\varepsilon_{t+1} \mid Y_t, Y_{t-1}, \ldots, Y_1) = E(\varepsilon_{t+1}) = 0 \tag{4.3.5}$$

Thus (4.3.3) can be written as

$$\hat{Y}_{t+1}(t+1) = \mu + \phi(Y_t - \mu) . \tag{4.3.6}$$

Consider a general δ time unit into the future from time t, Replacing t by $t+\delta$ in (4.3.1) and taking conditional expectations of both sides,

$$\hat{Y}_{t+\delta}(t+\delta) = \mu + \phi[\hat{Y}_{t+(\delta-1)}(t+\delta-1) - \mu] \qquad \text{for } \delta \geq 1. \tag{4.3.7}$$

Iterating backwards on δ in (4.3.7), we have

$$\hat{Y}_{t+\delta}(t+\delta) = \mu + \phi^{\delta}(Y_t - \mu) \tag{4.3.8}$$

Consider the forecast error:

$$e_{t+1}(t+1) = Y_{t+1}(t+1) - \hat{Y}_{t+1}(t+1) = \phi(Y_t - \mu) + \mu + \varepsilon_{t+1} - [\phi(Y_t - \mu) + \mu] = \varepsilon_{t+1}. \tag{4.3.9}$$

The forecast error variance is given by:

$$Var[e_{t+1}(t+1)] = \sigma^2. \tag{4.3.10}$$

The forecast error is given by:

$$\begin{aligned} e_{t+\delta}(t+\delta) &= Y_{t+\delta}(t+\delta) - \hat{Y}_{t+\delta}(t+\delta) \\ &= Y_{t+\delta}(t+\delta) - \mu - \phi^{\delta}(Y_t - \mu) \\ &= \varepsilon_{t+\delta} + \phi\varepsilon_{t+\delta-1} + \ldots + \phi^{\delta-1}\varepsilon_{t+1} + \phi^{\delta}\varepsilon_t + \ldots - \phi^{\delta}(\varepsilon_t + \phi\varepsilon_{t-1} + \ldots) \\ &= \varepsilon_{t+\delta} + \phi\varepsilon_{t+\delta-1} + \phi^2\varepsilon_{t+\delta-2} + \ldots + \phi^{\delta-1}\varepsilon_{t+1}. \end{aligned} \tag{4.3.11}$$

Note that $E[e_{t+\delta}(t+\delta)] = 0$; thus the forecasts are unbiased.

From (4.3.11), we have

$$Var[e_{t+\delta}(t+\delta)] = \sigma^2 \sum_{j=0}^{\delta-1} \psi_j^2 \quad \text{where } \psi_j = \phi^j. \tag{4.3.12}$$

In particular, for an AR(1) process:

$$Var[e_{t+\delta}(t+\delta)] = \sigma^2 \frac{1-\phi^{2\delta}}{1-\phi^2}. \tag{4.3.13}$$

(B) *MA*(1) Case

Consider the *MA*(1) case with a nonzero mean:

$$Y_t = \mu + \varepsilon_t - \theta\varepsilon_{t-1}. \tag{4.3.14}$$

Replacing t by $t+1$ and taking conditional expectations of both sides, we have

$$\hat{Y}_{t+1}(t+1) = \mu - \theta E(\varepsilon_t \mid Y_t, Y_{t-1}, \ldots, Y_1), \tag{4.3.15}$$

where $E(\varepsilon_t \mid Y_t, Y_{t-1}, \ldots, Y_1) = \varepsilon_t$. (4.3.16)

From (4.3.15) and (4.3.16), we thus have the one-step-ahead forecast for the *MA*(1) model:

$$\hat{Y}_{t+1}(t+1) = \mu - \theta\varepsilon_t. \tag{4.3.17}$$

For longer lead times we have

$$\hat{Y}_{t+\delta}(t+\delta) = \mu + E(\varepsilon_{t+\delta} \mid Y_t, Y_{t-1}, \ldots, Y_1) - \theta E(\varepsilon_{t+\delta-1} \mid Y_t, Y_{t-1}, \ldots, Y_1). \tag{4.3.18}$$

Both $\varepsilon_{t+\delta}$ and $\varepsilon_{t+\delta-1}$ are independent of $Y_t, Y_{t-1}, \ldots, Y_1$ for $\delta > 1$. Consequently, these conditional expectations are zero, and we have

$$\hat{Y}_{t+\delta}(t+\delta) = \mu \qquad \text{for } \delta > 1. \tag{4.3.19}$$

(C) General ARIMA (p,0,q) Model

For the general stationary ARIMA$(p,0,q)$ model, the formula for computing forecasts is given by:

$$\hat{Y}_{t+\delta}(t+\delta) = \phi_1 \hat{Y}_{t+(\delta-1)}(\delta-1) + \phi_2 \hat{Y}_{t+(\delta-2)}(\delta-2) + \ldots + \phi_p \hat{Y}_{t+(\delta-p)}(\delta-p) + \theta_0$$

$$-\theta_1 E(\varepsilon_{t+\delta-1} \mid Y_t, Y_{t-1}, ... Y_1) - \theta_2 E(\varepsilon_{t+\delta-2} \mid Y_t, Y_{t-1}, ..., Y_1)$$

$$- ... - \theta_q E(\varepsilon_{t+\delta-q} \mid Y_t, Y_{t-1}, ..., Y_1) \qquad (4.3.20)$$

where $E(\varepsilon_{t+j} \mid Y_t, Y_{t-1}, ..., Y_1) = \begin{cases} 0 & \text{for } j \geq 1 \\ \varepsilon_{t+j} & \text{for } j \leq 0 \end{cases}$.

As an example, consider an ARIMA(1,0,1) model. We have

$$\hat{Y}_{t+1}(t+1) = \phi Y_t + \theta_0 - \theta\varepsilon_t \qquad (4.3.21)$$

with $\hat{Y}_{t+2}(t+2) = \phi \hat{Y}_{t+1}(t+1) + \theta_0$.

More generally,

$$\hat{Y}_{t+\delta}(t+\delta) = \phi \hat{Y}_{t+(\delta-1)}(\delta - 1) + \theta_0 \qquad \text{for } \delta \geq 2. \qquad (4.3.22)$$

(4.3.21) and (4.3.22) can also be solved by normal iteration to get the alternative explicit expression:

$$\hat{Y}_{t+\delta}(t+\delta) = \mu + \phi^{\delta}(Y_t - \mu) - \phi^{\delta-1}\theta\varepsilon_t \qquad \text{for } \delta \geq 1. \qquad (4.3.23)$$

(D) Non-stationary ARIMA Models

A time series $\{Y_t\}$ is said to follow an integrated autoregressive moving average model if the d th difference $\nabla^d Y_t$ is a stationary ARMA process. If $\nabla^1 Y_t$ is ARMA(p,q), we say that Y_t is ARIMA$(p,1,q)$.

Consider an ARIMA$(p,1,q)$ process:

$$Y_t - Y_{t-1} = \phi_1(Y_{t-1} - Y_{t-2}) + \phi_2(Y_{t-2} - Y_{t-3}) + ...$$

$$+ \phi_p(Y_{t-p} - Y_{t-p-1}) + \varepsilon_t - \theta_1\varepsilon_{t-1} - \theta_2\varepsilon_{t-2} - ... - \theta_q\varepsilon_{t-q}. \qquad (4.3.24)$$

We can rewrite this as:

$$Y_t = (1+\phi_1)Y_{t-1} + (\phi_2 - \phi_1)Y_{t-2} + (\phi_3 - \phi_2)Y_{t-3} + \ldots$$

$$+ (\phi_p - \phi_{p-1})Y_{t-p} - \phi_p Y_{t-p-1} + \varepsilon_t - \theta_1\varepsilon_{t-1} - \theta_2\varepsilon_{t-2} - \ldots - \theta_q\varepsilon_{t-q}\,. \qquad (4.3.25)$$

Forecasting for non-stationary ARIMA models is quite similar to forecasting for stationary ARMA models. The ARIMA $(p,1,q)$ model can be written as a non-stationary ARMA $(p+1,q)$ model. In other words, we can write (4.3.25) as:

$$Y_t = \varphi_1 Y_{t-1} + \varphi_2 Y_{t-2} + \ldots + \varphi_{p+1} Y_{t-p-1} + \theta_0 + \varepsilon_t$$

$$- \theta_1\varepsilon_{t-1} - \theta_2\varepsilon_{t-2} - \ldots - \theta_q\varepsilon_{t-q} \qquad \text{for } t > -m \qquad (4.3.26)$$

where $\varphi_1 = 1 + \phi_1$, $\varphi_j = \phi_j - \phi_{j-1}$, for $j = 2,3,\ldots,p$,

and $\varphi_{p+1} = -\varphi_p$.

To illustrate forecasting with an ARIMA(p,1,q) model, consider the ARIMA(1,1,1) model:

$$Y_t - Y_{t-1} = \phi(Y_{t-1} - Y_{t-2}) + \theta_0 + \varepsilon_t - \theta\varepsilon_{t-1},$$

so that $Y_t = (1+\phi)Y_{t-1} - \phi Y_{t-2} + \theta_0 + \varepsilon_t - \theta\varepsilon_{t-1}$. (4.3.27)

Thus $\hat{Y}_{t+1}(t+1) = (1+\phi)Y_t - \phi Y_{t-1} + \theta_0 - \theta\varepsilon_t$,

$$\hat{Y}_{t+2}(t+2) = (1+\phi)\hat{Y}_{t+1}(t+1) - \phi Y_t + \theta_0,$$

and $\hat{Y}_{t+\delta}(t+\delta) = (1+\phi)\hat{Y}_{t+(\delta-1)}(t+\delta-1) - \phi\hat{Y}_{t+(\delta-2)}(t+\delta-2) + \theta_0$, for $\delta > 2$. (4.3.28)

4.4 Estimation and Forecasting using the Greatest Deviation Correlation Coefficient r_{gd}

This section continues to show how estimation and forecasting on time series models can be performed using any nonparametric correlation coefficient. In particular, the method given is illustrated using the Greatest Deviation correlation coefficient, r_{gd}. The reason to use the Greatest Deviation correlation coefficient, r_{gd}, is that the r_{gd} approach is more robust and resistant to outliers than the least squares method.

Consider the AR(1) process with a nonzero mean:

$$Y_t - \mu = \phi(Y_{t-1} - \mu) + \varepsilon_t . \tag{4.4.1}$$

To minimize the sum of squares of the differences

$$Y_t - \mu - \phi(Y_{t-1} - \mu),$$

compute $$S^*(\phi,\mu) = \sum_{t=2}^{n}[(Y_t - \mu) - \phi(Y_{t-1} - \mu)]^2 . \tag{4.4.2}$$

We first take the derivative of $S^*(\phi,\mu)$ with respect to μ:

$$\frac{\partial S^*}{\partial \mu} = \sum_{t=2}^{n} 2[(Y_t - \mu) - \phi(Y_{t-1} - \mu)](-1+\phi)) = 0 . \tag{4.4.3}$$

Solving the above equation (4.4.3) gives:

$$\hat{\mu} = \frac{\sum_{t=2}^{n} Y_t - \phi \sum_{t=2}^{n} Y_{t-1}}{(n-1)(1-\phi)}$$

$$= \frac{\sum_{t=2}^{n} Y_t}{(n-1)(1-\phi)} - \frac{\phi \sum_{t=2}^{n} Y_{t-1}}{(n-1)(1-\phi)} . \tag{4.4.4}$$

For large n, (4.4.4) becomes approximately:

$$\hat{\mu} \approx \frac{\bar{Y}}{1-\phi} - \frac{\phi \sum_{t=2}^{n} Y_{t-1}}{(n-1)(1-\phi)} = \frac{\bar{Y}}{1-\phi} - \frac{\phi \bar{Y}}{1-\phi} = \bar{Y} \tag{4.4.5}$$

Then, taking the derivative of $S^{*}(\phi, \mu)$ with respect to ϕ, and substituting $\hat{\mu}$ for μ:

$$\frac{\partial S^{*}}{\partial \phi} = -\sum_{t=2}^{n} 2[(Y_t - \bar{Y}) - \phi(Y_{t-1} - \bar{Y})](Y_{t-1} - \bar{Y}) = 0\,. \tag{4.4.6}$$

Solving (4.4.6) for ϕ gives:

$$\hat{\phi} = \frac{\sum_{t=2}^{n}(Y_t - \bar{Y})(Y_{t-1} - \bar{Y})}{\sum_{t=2}^{n}(Y_{t-1} - \bar{Y})^2}.$$

Let $\quad Y_t^c = Y_t - \bar{Y}$, $Y_{t-1}^c = Y_{t-1} - \bar{Y}$ be the centered data.

Substituting this into (4.4.6):

$$\frac{\partial S^{*}}{\partial \phi} = -\sum_{t=2}^{n} 2[(Y_t^c - \phi Y_{t-1}^c)]Y_{t-1}^c = 0\,,$$

i.e., $\quad \sum_{t=2}^{n}(Y_t^c - \phi Y_{t-1}^c)Y_{t-1}^c = 0 \qquad (4.4.7)$

$\Rightarrow \quad (Y_t^c - \phi Y_{t-1}^c) \perp Y_{t-1}^c \qquad (4.4.8)$

$\Rightarrow \quad r(Y_{t-1}^c, Y_t^c - \phi Y_{t-1}^c) = 0. \qquad (4.4.9)$

Replacing r with the Greatest Deviation correlation coefficient r_{gd} gives:

$$r_{gd}(Y_{t-1}^c, Y_t^c - \phi Y_{t-1}^c) = 0 \qquad \text{for } t = 2,\ldots,n\ . \tag{4.4.10}$$

We can then use the r_{gd} regression routine to estimate the parameter ϕ in (4.4.10).

Because location shifts do not affect the correlation coefficient equation, then to estimate the parameters σ and μ, we again use the r_{gd} regression method ([16] and [17]) by solving:

$$r_{gd}(\underline{q}, (Y_t^c - \phi Y_{t-1}^c)^o - s\underline{q}) = 0 \qquad (4.4.11)$$

for s as an estimate of σ. $\underline{q}$ here is the vector of $N(0,1)$ quantiles. If Φ is the N(0,1) cumulative distribution function (CDF), then $\Phi^{-1}(\frac{i}{n+1}) = q_i$, $i = 1,2,...,n$.

An estimate of μ may be found as:

$$\hat{\mu} = median\,[(Y_t - \hat{\phi} Y_{t-1})/(1-\hat{\phi})]. \qquad (4.4.12)$$

4.5 Residual Analysis

When a model has been fit to a time series, it is advisable to check that the model really does provide an adequate description of the data. As with most statistical models, this is done by examining the residuals, which are defined by

residuals = observation - fitted value.

Consider for example an AR(2) model with a constant term:

$$Y_t = \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \theta_0 + \varepsilon_t.$$

Having estimated ϕ_1, ϕ_2, and θ_0, the residuals are defined as

$$\hat{\varepsilon}_t = Y_t - \hat{\phi}_1 Y_{t-1} - \hat{\phi}_2 Y_{t-2} - \hat{\theta}_0 \quad \text{for } t = 3,4,..,n.$$

If AR(2) model is correct and if the parameter estimates are close to the true values, then the residuals should have nearly the properties of independent, identically distributed, normal random variables with zero means and standard deviation σ_ε.

The following assessments on the residuals are essential:

(1) The autocorrelation function (ACF) and partial autocorrelation function (PACF) of the residuals should not be significantly different from 0.

(2) The residuals should be without pattern, i.e., they should be white noise.

4.6 Illustrative Examples

Example 1. In this example we follow the monthly inflation rate from January 1970 through December 1985 (see Figure 4.2). The monthly inflation rate in August 1973 was 1.8%, which if continued would have produced an annual rate of over 23%. This was higher than any other monthly rate. Such an observation is somewhat unusual and can have an effect in conducting a time series analysis.

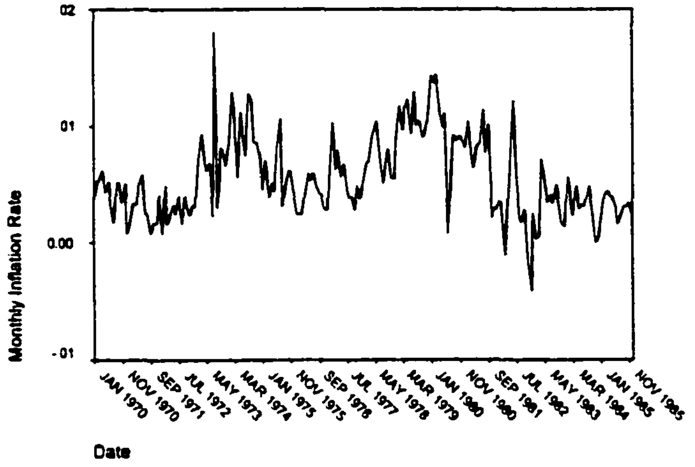


Figure 4.2 Monthly Inflation Rate from January 1970 through December 1985

ARIMA estimation (with the mild outlier):

We will try to develop an ARIMA model for this inflation series.

(1) Identifying the Model

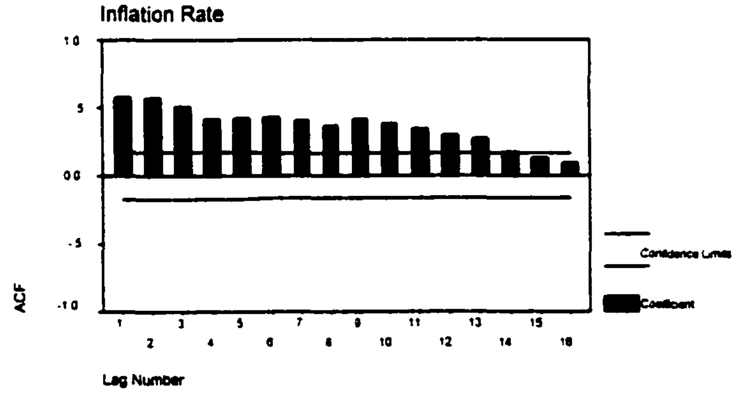
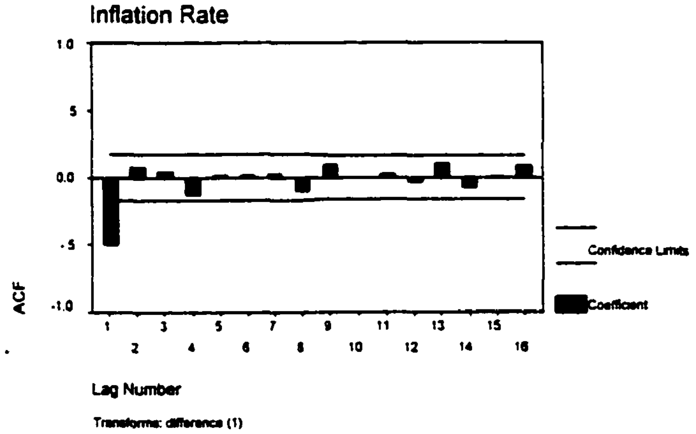


Figure 4.3 ACF of Inflation Series

This ACF plot starts out with large positive values, which die out very slowly at increasing lags. This pattern confirms that the series is not stationary, and that we should take differences to attempt to remove this nonstationarity.

Let $\nabla^1 Y_t = Y_t - Y_{t-1}$,

where Y_t = time series observation at time t,



Y_{t-1} = time series observation at time period $t-1$.

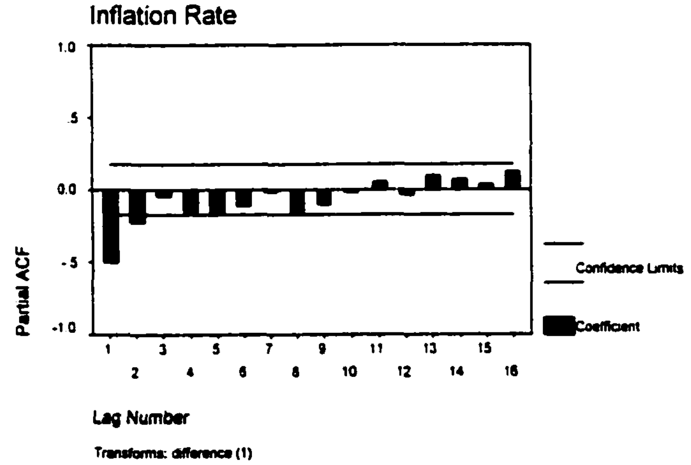


Figure 4.4 ACF and PACF for Differenced Series

The ACF of the differenced series shows a spike at lag 1, while the PACF shows rapid attenuation from its initial value. These patterns suggest an MA(1) process (see Appendix 3, "Guide to ACF/PACF Plots", SPSS Trends). Since we differenced the original series to obtain the MA(1) patterns, our ARIMA identification includes one degree of differencing and a first-order moving average, i.e., an ARIMA(0,1,1) model.

Table 4.1 ARIMA(0,1,1) output for the inflation series

```
Termination criteria:
Parameter epsilon: .001
SSQ Percentage: .001
Maximum number of iterations: 10

Initial values:

MA1        .65199

Sum of squares = .00089164

              Iteration History:

  Iteration       Sum of Squares

          1            .00089041
          2            .00089035


Conclusion of estimation phase.
Estimation terminated at iteration number 3 because:
   Sum of squares decreased by less than .001 percent.

FINAL PARAMETERS:

Number of residuals  131
Standard error       .00261071


            Analysis of Variance:

              DF        Sum of Squares    Residual Variance

Residuals     130          .00089035            .00000682

           Variables in the Model:

              B          SEB     T-RATIO   APPROX. PROB.

MA1    .68464518   .06432286   10.643886        .0000000
```

The output for an ARIMA(0,1,1) least square model fit appear in Table 4.1. The first differences in monthly inflation rates followed an ARIMA (0,1,1) process with θ=0.685. Verifying that the resulting ARIMA residuals are white noise, consider the ACF and PACF shown for the residuals in Figure 4.5.

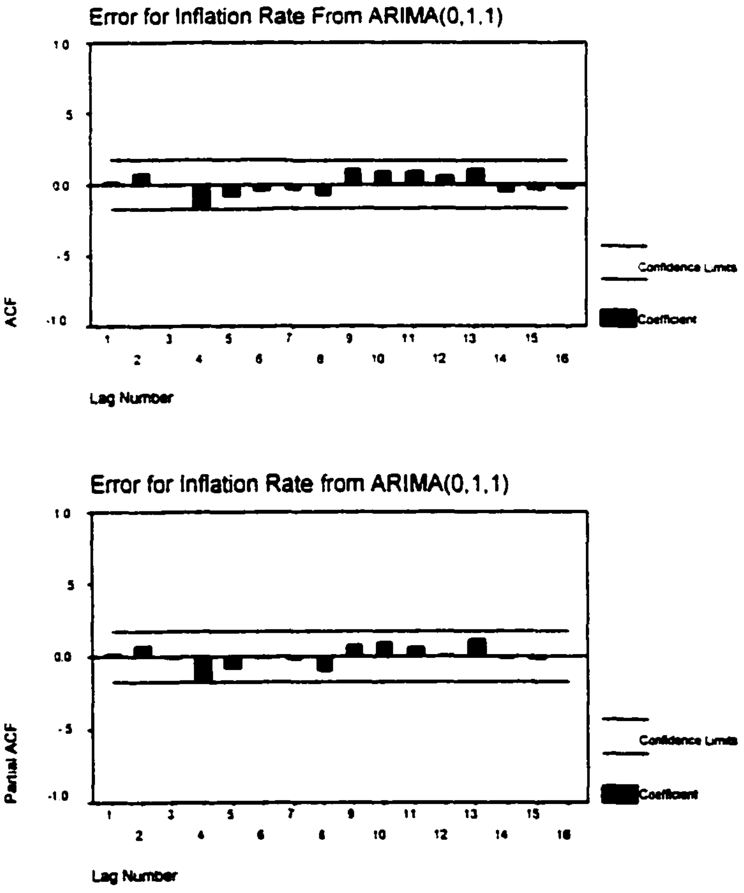


Figure 4.5 ACF and PACF for Residuals

None of the residual autocorrelations exceeds the confidence limits around 0. Since they are not statistically significant at any lag, we have no evidence that the residuals are not a white noise process.

Figure 4.6 shows a sequence chart of the ARIMA(0,1,1) residuals. In general the residuals show no pattern, although the outlier of August 1973 is still present.

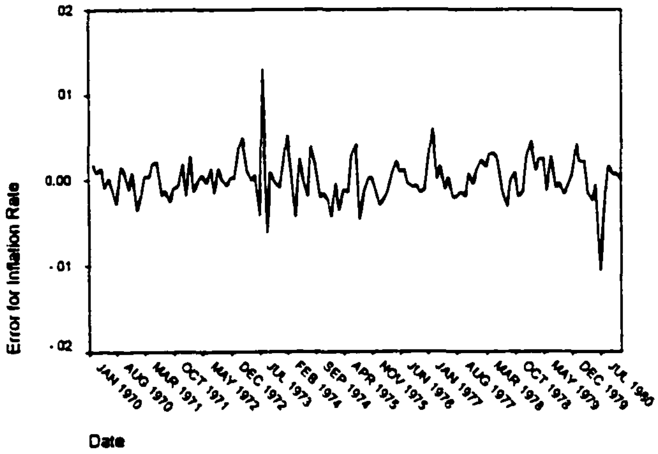


Figure 4.6 Residuals from ARIMA(0,1,1) including outlier

ARIMA estimation (with the mild outlier) using the r_{gd} method

First we need to identify the appropriate ARIMA model for use with r_{gd}.

Using the Greatest Deviation correlation coefficient r_{gd}, $ACF(1) = r_1 = r_{gd}(Y_t, Y_{t-1}), \ldots,$

$ACF(k) = r_k = r_{gd}(Y_t, Y_{t-k})$, $PACF(k) = r_{kk} = r_{gd}(Y_t, Y_{t-k} \mid Y_{t-1}, Y_{t-2}, \ldots, Y_{t-k+1})$, $k = 1, 2, \ldots.$

From (4.1.10), replacing ρ by r_{gd}, the PACF at lag k is:

$$r_{kk} = \frac{r_k - \sum_{j=1}^{k-1} r_{k-1,j} r_{k-j}}{1 - \sum_{j=1}^{k-1} r_{k-1,j} r_j}$$

where $r_{kj} = r_{k-1,j} - r_{kk} r_{k-1,k-j}$ for $j = 1, 2, \ldots, k-1$.

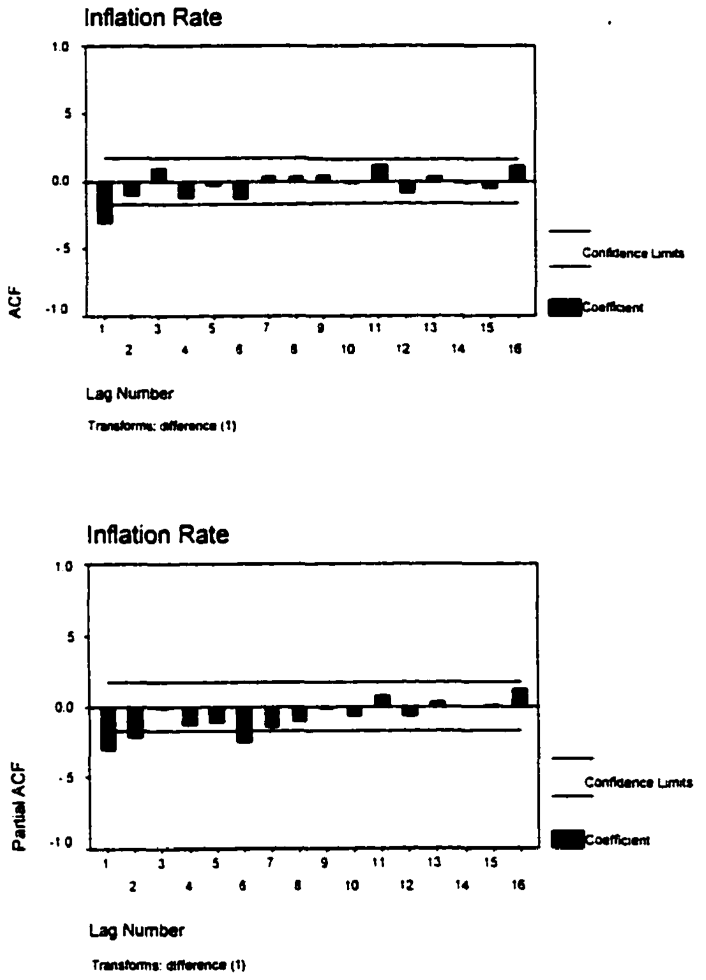


Figure 4.7 ACF and PACF using r_{gd} method

Figure 4.7 shows the ACF and PACF plots using the r_{gd} method. Comparing these plots with Figure 4.4, we see that using r_{gd} method has reduced the size of the negative ACF at lag 1. Both the ACF and PACF show declines from their initial value at lag 1, rather than spikes.

This suggests a model with both autoregressive and moving average components,; hence, an ARIMA(1,1,1) model (see Appendix 3).

The output for the ARIMA(1,1,1) model fit for the inflation series using the r_{gd} method appears in Table 4.2.

<u>Table 4.2</u> ARIMA(1,1,1) using r_{gd} method:

```
Termination criteria:
Parameter epsilon: .001
SSQ Percentage: .001
Maximum number of iterations: 10

Initial values:

AR1         .33674
MA1         .72357

Sum of squares = .00067359

              Iteration History:

  Iteration        Sum of Squares

          1            .00066685
          2            .00066584
          3            .00066577

Conclusion of estimation phase.
Estimation terminated at iteration number 4 because:
   Sum of squares decreased by less than .001 percent.

FINAL PARAMETERS:

Number of residuals  131
Standard error       .0022672


            Analysis of Variance:

              DF        Sum of Squares    Residual Variance

Residuals     129           .00066577            .00000514

           Variables in the Model:

              B          SEB     T-RATIO   APPROX. PROB.

AR1    .40987963   .12585801    3.256683       .00144041
MA1    .83249607   .07776481   10.705305       .00000000
```

The monthly inflation rates followed an ARIMA (1,1,1) process with θ=0.8325 and ϕ=0.4098 using the r_{gd} method.

We compared these results with Table 4.1. The standard error of the residuals is smaller. The model estimated using the r_{gd} method seems to be slightly better on these statistical grounds. That is not surprising, since r_{gd} is resistant to outliers.

Verifying that the resulting ARIMA residuals are white noise process, consider the ACF and PACF shown for the residuals in Figure 4.8.

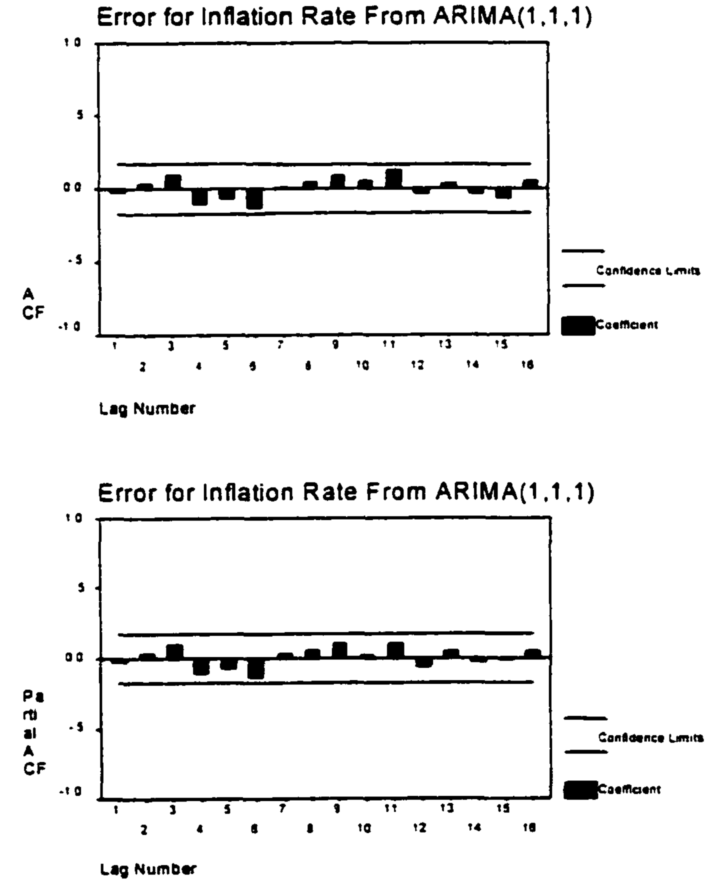


Figure 4.8 ACF and PACF for Residuals using the r_{gd} method

Figure 4.8 shows that the residual ACF and PACF for this last model is acceptable. The robust r_{gd} method might protect against outliers causing misspecfication of a time series model.

Example 2: Modeling and forecasting the healthcare cost and utilization for the years 1997 to 2001.

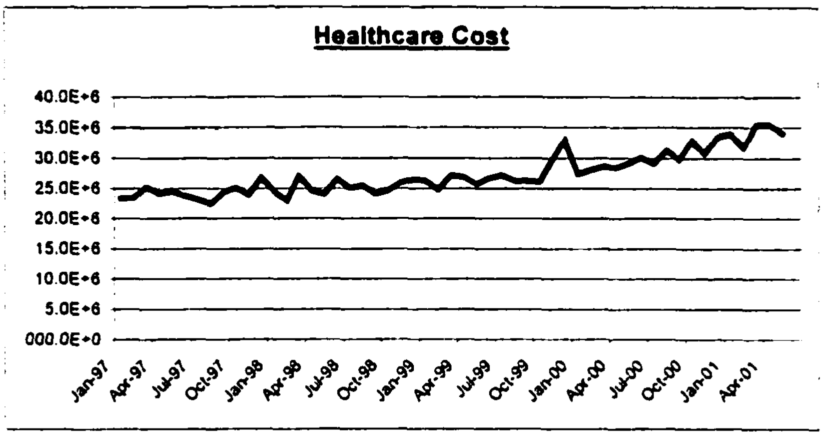


Figure 4.9 Monthly Healthcare Cost from Years 1997 to 2001

Figure 4.9 shows the Blue Cross Blue Shield monthly medical utilization for the years 1997 to 2001. To identify an appropriate model for these healthcare cost data, an ACF plot was created in Figure 4.10.

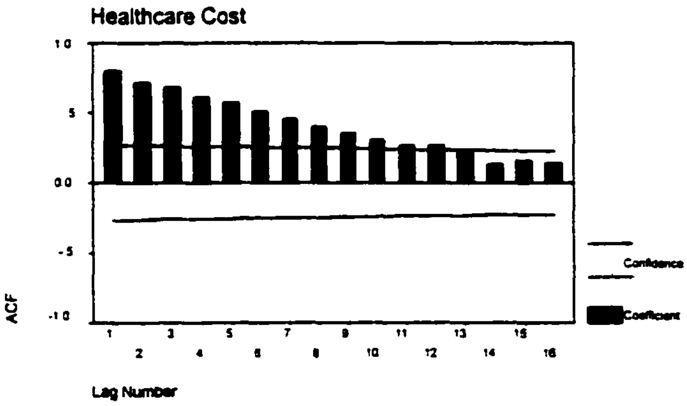


Figure 4-10 ACF of Healthcare Cost

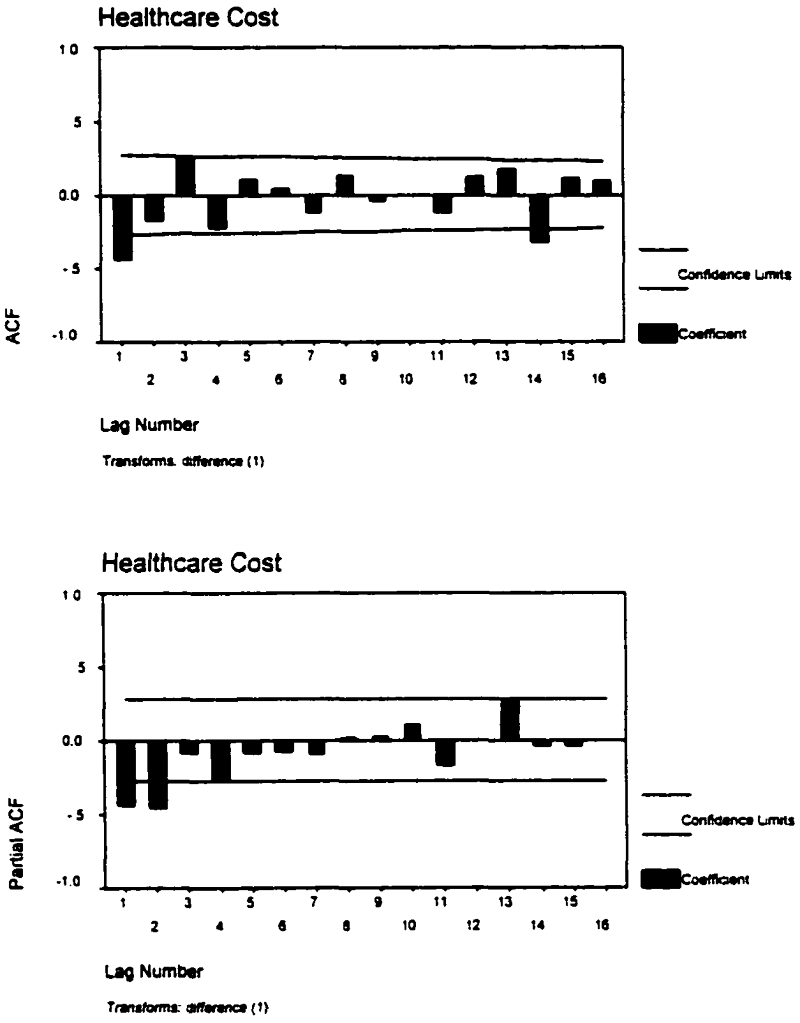


Figure 4.11 ACF and PACF of the differenced Healthcare Cost

The time sequence chart (Figure 4.9) of healthcare costs suggested that the series was not stationary. The autocorrelation plot (Figure 4.10) starts out with large positive values, which die out very slowly at increasing lags. This pattern confirms that the series is not stationary, and that differences should be taken when analyzing the data.

In viewing Figure 4.11, the PACF of the differenced series shows one spike at lag 1, while the ACF shows rapid attenuation from its initial value. These patterns suggest an ARIMA(1,1,0) process (see Appendix 3).

Both the least squares and r_{gd} methods were used with S-plus to estimate the parameters of the ARIMA(1,1,0) model. The OLS output is given in Table 4.3.

Table 4.3 Ordinary Least Squares Estimation

```
Termination criteria:
Parameter epsilon: .001
SSQ Percentage: .001
Maximum number of iterations: 10

Initial values:

AR1       -.41659


               Iteration History:

  Iteration       Sum of Squares

          1         1.5234821E+14


Conclusion of estimation phase.
Estimation terminated at iteration number 2 because:
   Sum of squares decreased by less than .001 percent.

FINAL PARAMETERS:

Number of residuals  52
Standard error       1725246.6

             Analysis of Variance:

               DF  Adj. Sum of Squares    Residual Variance

Residuals      51         1.5234821E+14        2976475795132

            Variables in the Model:

                  B          SEB      T-RATIO   APPROX. PROB.

AR1     -.41335004   .12764725   -3.2382213        .00211764
```

Table 4.4 Comparison of the r_{gd} method and least squares with original data (one outlier)

	$\hat{\phi}$ (B)	t	p-value
least squares	-0.41335004	-3.2382213	0.0021
r_{gd}	-0.4060088	-3.7371	0.0000

Table 4.4 gives the parameter estimates for the least squares method and r_{gd} method. There are no outliers in this example; hence, the r_{gd} method and least squares method give similar results.

Suppose that there exists another outlier in this time series. We recalculate the coefficients.

Table 4.5 Comparisons of r_{gd} method and least squares (with another outlier)

	$\hat{\phi}$ (B)	t	p-value
least squares	-0.4508505	-3.213653	0.002
r_{gd}	-0.4060088	-3.737100	0.0000

Table 4.5 gives the comparison between the least squares method and r_{gd} method (for calculations, see Appendix 4). The r_{gd} method is clearly more robust to the effects of the outlier than least squares.

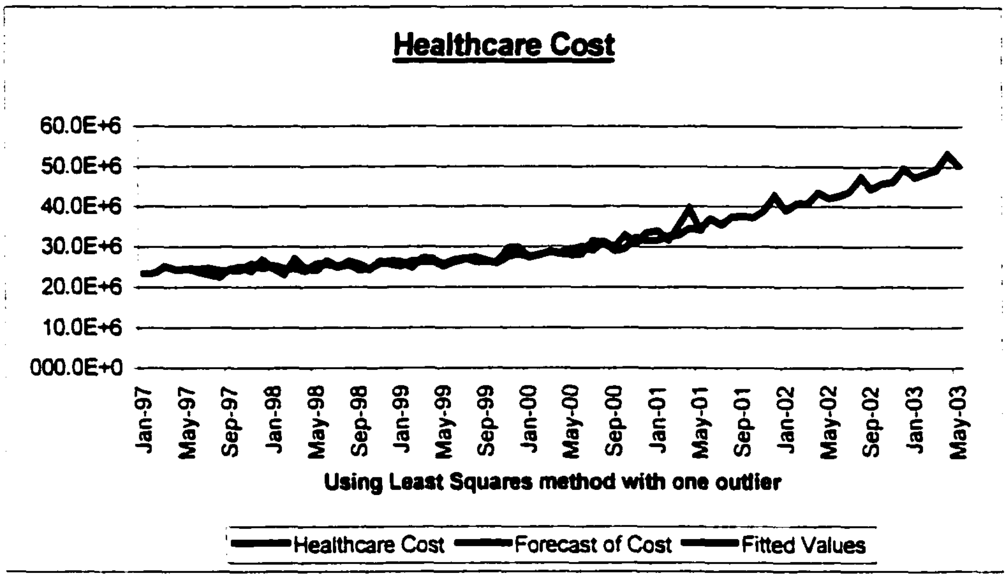


Figure 4.12 Forecast using least squares method

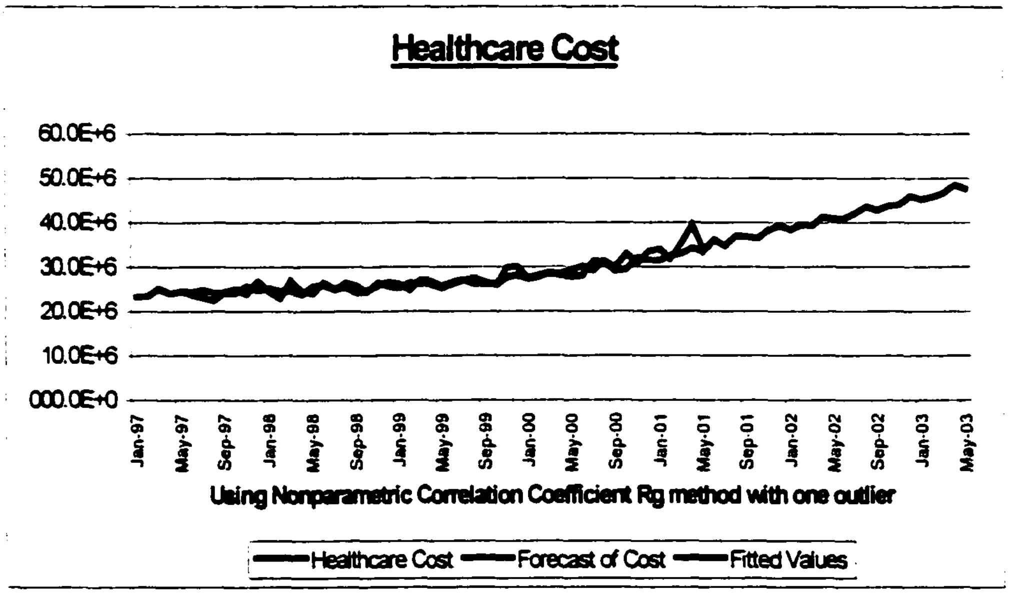


Figure 4.13 Forecast using Greatest Deviation correlation coefficient r_{gd} method

We use the formulas in Section 4.3 and Section 4.4 to calculate forecasted values for both the least squares and r_{gd} methods for comparison (Figures 4.12, 4.13).

Forecasting three months into the future from May of 2001 for both the least squares and r_{gd} methods, the results are summarized in Table 4.6.

Table 4.6 Forecast comparisons of r_{gd} method and least squares

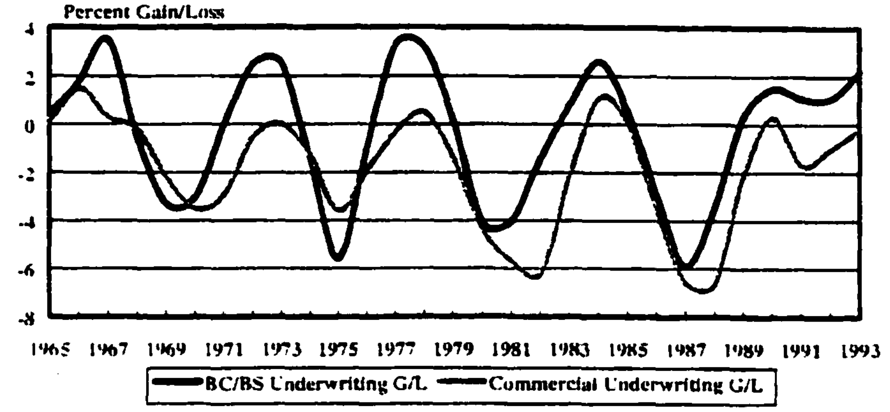
Date	r_{gd} method	Least Squares	Actual	Prediction error (r_{gd})	Prediction error (LS)
June 2001	36,135,374	36,992,763	35,975,884	0.4%	2.8%
July 2001	34,616,659	35,405,008	34,837,656	-0.6%	1.6%
August 2001	36,942,358	37,446,625	36,745,892	0.05%	1.9%

In viewing this table, the prediction error is much smaller when we choose the r_{gd} method as compared to the least squares method. For this example, the r_{gd} method is better than the least squares or maximum likelihood methods. It performs robustly when the data have some suspect values.

Example 3. Trend Analysis and Forecasting Health Insurer Profitability

We use a statistical ARMA model fit with the nonparametric correlation coefficient r_{gd} and utilize forecasted values of the healthcare cost to project underwriting results. The Blue Cross/Blue Shield system reported underwriting losses between 1995 and 1998. The prolonged losses were attributable to the low increases in premiums as companies tried to gain or maintain market share.

BLUE CROSS/BLUE SHIELD AND COMMERCIAL UNDERWRITING GAIN/LOSS



Commercial data is not available after 1993

Figure 4.14 Blue Cross/Blue Shield and Commercial Underwriting Gain/Loss

Figure 4-14 shows a consistent pattern of three consecutive years of gain followed by three consecutive years of loss. Breaking the string of four consecutive years of underwriting loss ending in 1998, which followed six years of underwriting gain. 1999 showed a marginal gain, well within the range of statistical fluctuation of another loss year. Most business cycles, by definition, tend to be recurrent, but do not exhibit the level of regular periodicity seen in the Blue Cross/Blue Shield underwriting results, at least up to 1992.

Figure 4.14 illustrates the health insurance gains/losses for commercial carriers compared to Blue Cross/Blue Shield plans. These results are not completely comparable because of differences in commercial reporting. However, they do exhibit a great deal of consistency in the cyclical patterns.

The Underwriting Cycle:

Underwriting gains and losses are results of the difference between revenues and expenses. The former is represented by the amount of premiums earned, and the latter is

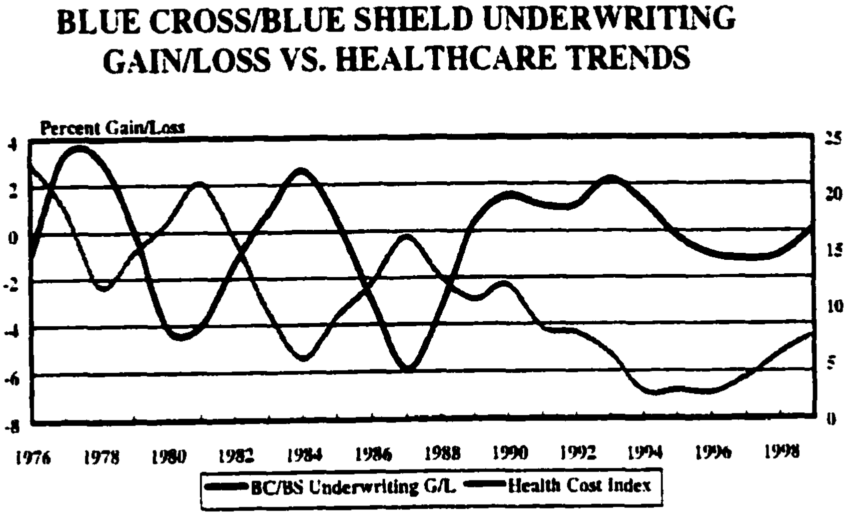


Figure 4.15 Blue Cross/Blue Shield Underwriting Gain/Loss vs. Healthcare Trends

measured by the amount of incurred claims and other operating expenses. If revenues are rising faster than costs, then a gain is likely to occur. Conversely, if the insurer's claims

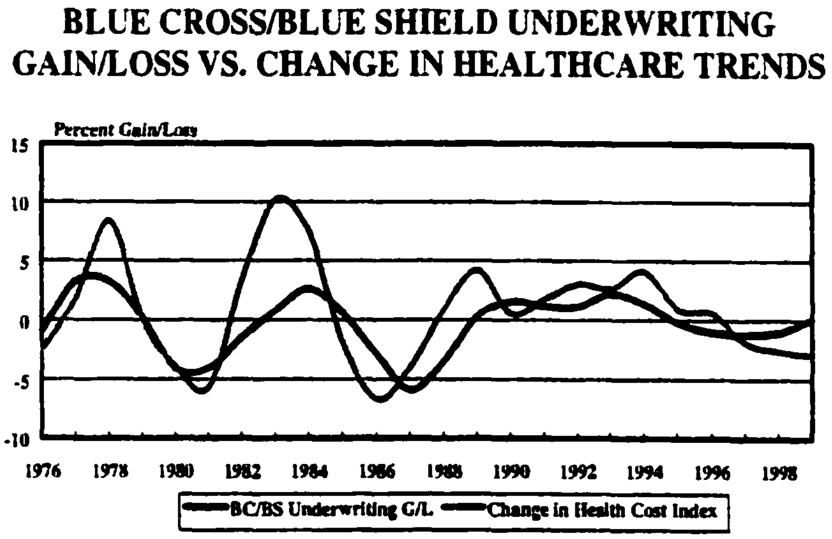


Figure 4.16 Blue Cross/Blue Shield Underwriting Gain/Loss vs.

Change in Healthcare Trends

and expenses rises faster than the premiums charged, a loss will result. Typically, some profit margin is built into target premiums. As a result, underwriting gains should occur unless expenses and claims rise at a faster rate than revenues plus the margin percentage.

Figure 4.15 illustrates the Blue Cross/Blue Shield underwriting results compared to healthcare cost trends as represented by the Health Cost Index (HCI). It is apparent that underwriting results and healthcare trends as measured by the HCI are inversely related. This pattern seems to diverge somewhat near the end of the period.

Figure 4.16 portrays a better visualization of this relationship by reflecting the change in HCI trends 18 months apart and by reversing the scale (changing positive numbers to negative numbers and vice versa) of the HCI trend graph. This 18 month lag follows the premise that cost trends for providers lead health insurance premiums by about 18 months. This lag is due to the time needed to collect and analyze historical claims data and to implement changes in premiums.

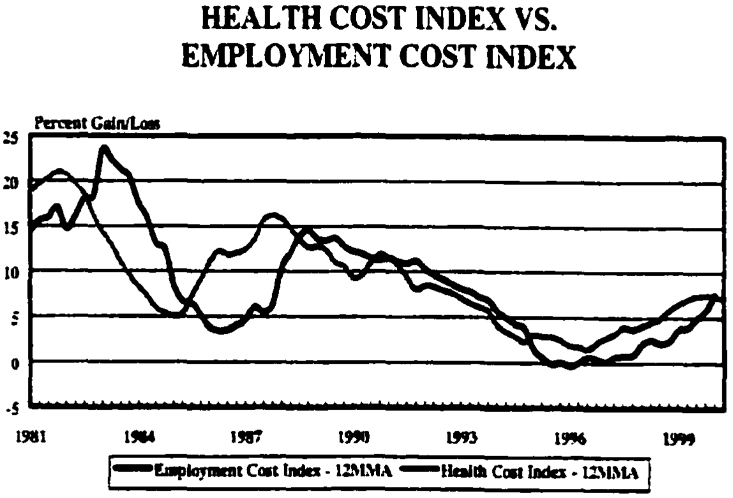
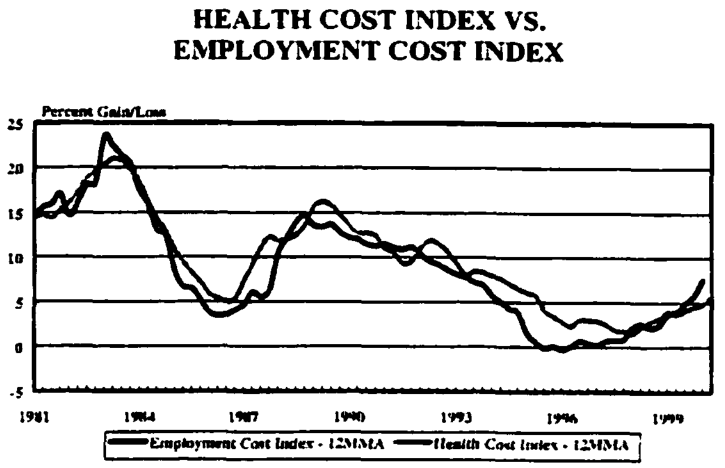


Figure 4.17 Health Cost Index vs. Employment Cost Index

Figures 4.17 and 4.18 illustrate the twelve month moving average of Employment Cost Index versus the HCI. Figure 4.17 shows them on the actual time scale, and Figure 4.18 shows the HCI trends delayed 18 months to correspond more closely with the Employment Cost Index.



Health Cost Index is shifted forward (lagged) 18 months.

Figure 4.18 Health Cost Index vs. Employment Cost Index

The close correspondence between these two graphs is indicative of the delay that exists

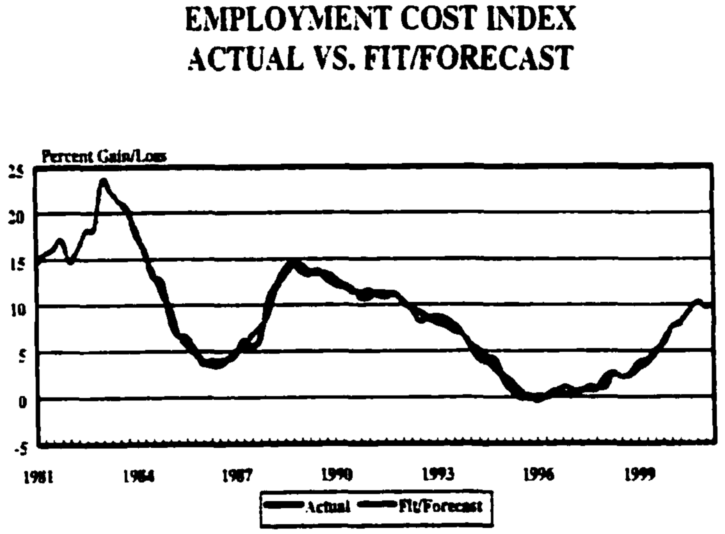


Figure 4.19 Employment Cost Index Actual vs. Forecast

between changes in claim cost trends and the insurers' recognition of these trend changes in premium rates. This close relationship has permitted us to build a statistical model for future forecasting. We use the nonparametric correlation coefficient r_{gd} method to forecast the Employment Cost Index shown in Figure 4.19.

Modeling and Forecasting Profitability:

Using the nonparametric correlation coefficient r_{gd} method, the next step is to formulate a statistical time series model that enables the forecasting of underwriting profitability. We use the formulas in Sections 4.3 and 4.4 to calculate the projected trends. Figure 4.20 shows the health insurance's billed charge trend, allowed charge trend, paid claim trend, and forecast using the nonparametric correlation coefficient r_{gd} method.

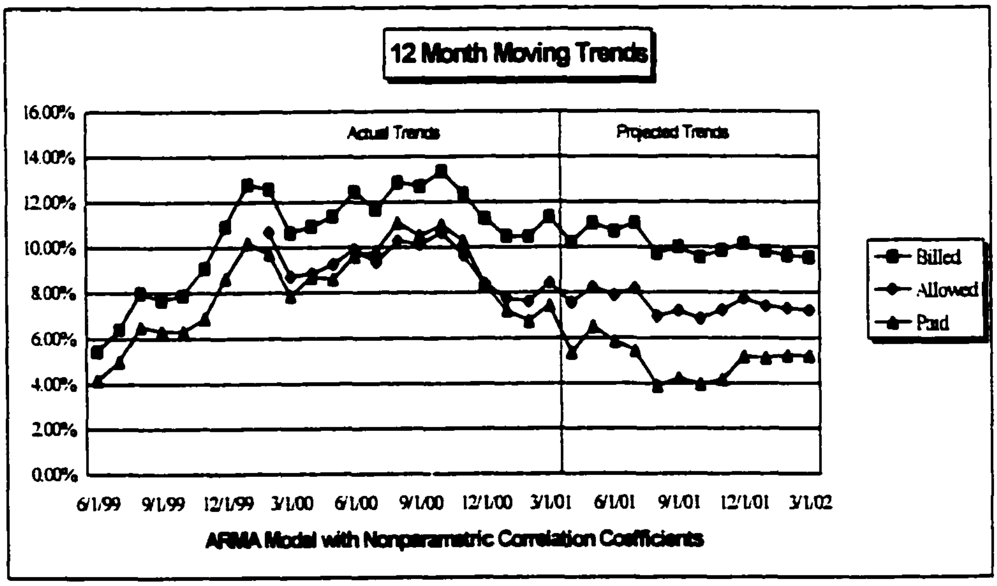


Figure 4.20 ARIMA model forecast using r_{gd} method

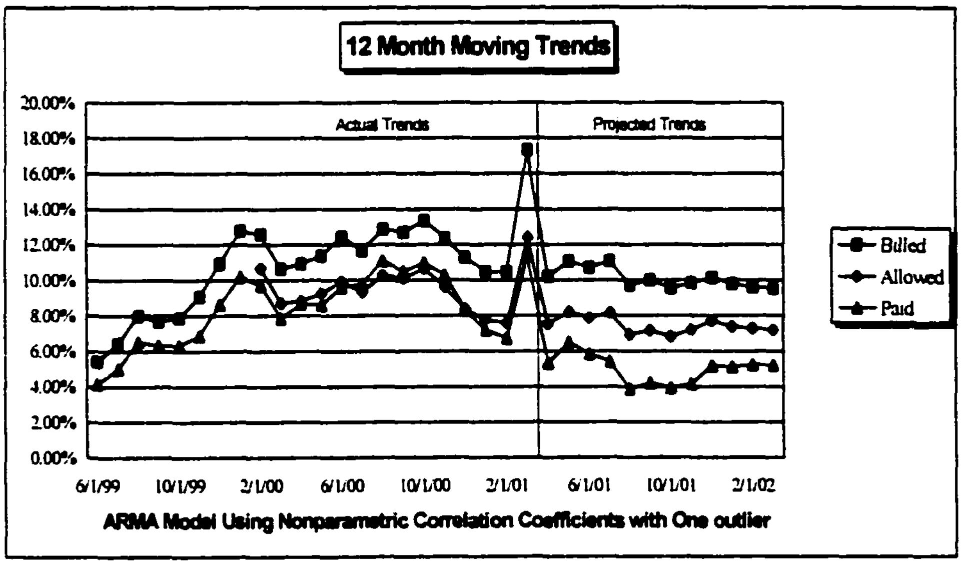


Figure 4.21 ARIMA model forecast using r_{gj} method (one outlier)

Supposed now an extreme outlier is added to the time series in March of 2001. Figure 4.20 shows the billed charge trend, allowed charge trend, paid claim trend, and forecast with the outlier using the nonparametric correlation coefficient r_{gj} method.

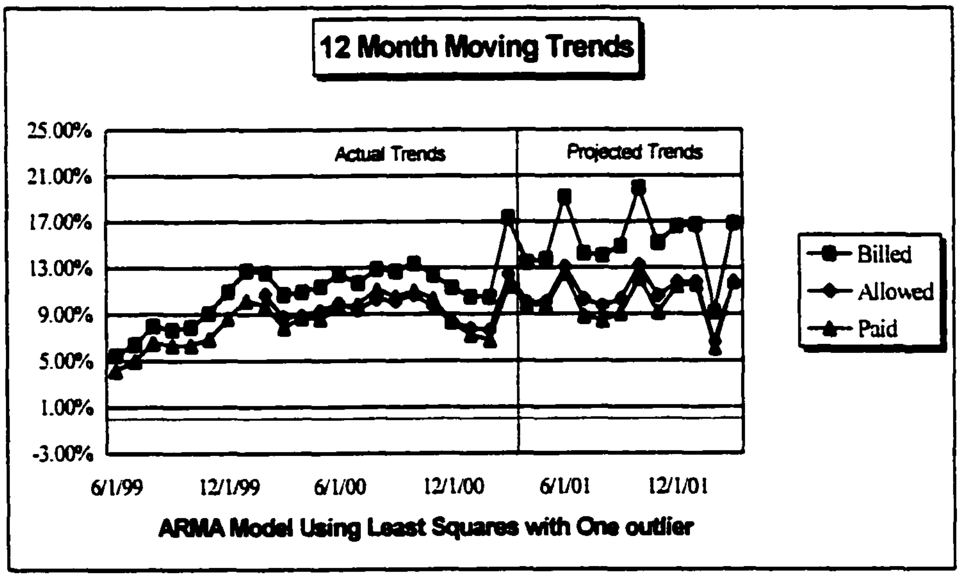


Figure 4.22 ARIMA model forecast using least squares (one outlier)

Figure 4.22 shows the billed charge trend, allowed charge trend, paid claim trend, and forecast with the outlier using the least squares method. Figures 4.21 and 4.22 show that the r_{gd} method provides much more stable projections and appears to be unaffected by the outlier, unlike least squares.

Table 4.7 Comparisons of least squares and r_{gd} methods (with an outlier)

Date	Least Squares Method	r_{gd} Method	Actual Trend
April 2001	13.44%	10.21%	10.56%
May 2001	13.74%	11.07%	11.01%
June 2001	19.14%	10.71%	10.86%
July 2001	14.22%	11.08%	11.03%

Forecasting four months into the future from March of 2001 for both the least squares and r_{gd} methods (with an outlier), the results are summarized in Table 4.7. In viewing this table, the prediction error is much smaller when we choose the r_{gd} method as compared to the least squares method. For this example, the r_{gd} method is better than the least squares . It performs robustly when the data have some suspect values.

Chapter 5

Main Results and Future Research

5.1 Main Results

We now summarize the main results of this dissertation.

In Chapter 1 we discussed linear regression models, some properties of the Greatest Deviation correlation coefficient r_{gd}, and the application of r_{gd} to the estimation of linear model parameters.

In Chapter 2 and Chapter 3, we studied generalized linear models and nonlinear models fit with nonparametric correlation coefficients. Specifically, we investigated the robustness of parameter estimates to outliers using the nonparametric correlation coefficients method of model fitting. We illustrated that estimation is more robust to outliers if we choose the Greatest Deviation correlation coefficient r_{gd} method, as opposed to least squares.

In Chapter 4, we reviewed the time series models and estimation. We developed estimation methods for the class of ARIMA time series models using the Greatest Deviation correlation coefficient r_{gd} methodology. Parameter estimates obtained for several data sets show that the nonparametric correlation coefficient r_{gd} methodology is comparable to least squares and maximum likelihood estimation methods, when the data is well-behaved, but performs robustly in the presence of suspect data.

5.2 Further Research

In this section, we discuss some further problems for study that follow from this dissertation. Nonparametric approaches have become an area with an abundance of new methodological developments in recent years.

Future efforts pertaining to the subject matter in this dissertation will fall into three categories: theory, application, and performance. Theoretical research will include extensions of the statistical inferences using the Greatest Deviation correlation coefficient and the exploration of the use of any type of correlation coefficient into all areas of statistics. There are many research applications for the Greatest Deviation correlation coefficient and others, including financial event prediction, healthcare quality improvement research, etc. As the data sets grow larger, the computational effort required to implement the Greatest Deviation correlation coefficient methodology is great and warrants further study.

There are some other interesting and potential research areas:

1. Comparison of the r_{gd} methodology to methods using other robust nonparametric correlation coefficients.
2. Simulation studies (as opposed to using real data) for comparing least squares and nonparametric correlation coefficient.
3. Development of Inference procedures (confidence intervals and tests) using nonparametric correlation coefficients for generalized linear models, nonlinear models and time series models.

Bibliography

[1] Abraham, Bovas and Ledolter, Johannes "Statistical Methods for Forecasting", Wiley Series in Probability and Mathematical Statistics, 1983.

[2] Brockwell, Peter J. and Davis, Richard A. "Time Series: Theory and Methods", 1987.

[3] Becker, Richard A, Chambers, John M. and Wilks, Allan R. "The New S Language", 1988.

[4] Bosq, D. "Nonparametric Statistics for Stochastic Process" Second Edition, 1998.

[5] Berry, Donald A. and Lindgren, Bernard W. "Statistics, Theory and Methods", 1990.

[6] Chatfield, C. "The Analysis of Time Series", Fourth Edition, 1988.

[7] Cox, D.R. and Hinkley, D.V. "Theoretical Statistics", Chapman & Hall, 1994.

[8] Dahlquist, Germund "Numerical Methods", Prentice-Hall Series in Automatic Computation, 1974.

[9] Enders, Walter "Applied Economic Time Series", John Wiley & Sons, Inc. 1995.

[10] Franses, Philip Hans "Time Series Models for Business and Economic Forecasting", Cambridge University Press, 1998.

[11] Fox, John "Linear Statistical Models and Related Methods" Wiley Series in Probability and Mathematical Statistics, 1984.

[12] Dobson, Annette J., "An Introduction to Generalized Linear Models", Chapman and Hall, 1990.

[13] Draper, N. R. and Smith, H., "Applied Regression Analysis", John Wiley & Sons, Inc., 1966.

[14] Fuller, Wayne A., "Introduction to Statistical Time Series", Second Edition, 1996.

[15] Fokianos, Konstantinos "Power Divergence Family of Tests for Categorical Time Series Models", 2000.

[16] Gideon, Rudy A., et al, "Robust Linear Regression with Nonparametric Correlation Coefficient", submitted to Statistics, June 1993.

[17] Gideon, Rudy A., et al, "Nonparametric Correlation and Regression Methods", unpublished paper, 1994.

[18] Gideon, Rudy A., et al, "Multiple Linear Regression Using Nonparametric Correlation Coefficient Rg", unpublished paper, 1993.

[19] Gideon, Rudy and Rummel, Steven E. and Li, Hongzhe "Robust Simple Linear Regression with Nonparametric Correlation Coefficients", 1994.

[20] Gideon, Rudy and Hollister, Robert "A Rank Correlation Coefficient Resistant to Outliers", 1987.

[21] Gideon, Rudy, Bruder, John, Lee, Li-Chio and Thiel, Mike "Location Estimation with Nonparametric Correlation Coefficients", 1992.

[22] Gideon, Rudy "Population Value of Spearman's Modified Footrule Correlation Coefficient",

[23] Grunwald, Gary K. and Hyndman, Rob J., etc "Non-Gaussian Conditional Linear AR(1) Models", 1999.

[24] Green, P.J. and Silverman, B.W., "Nonparametric Regression and Generalized Linear Models", 1994.

[25] Hosmer, David W. and Lemeshow, Stanley "Applied Logistic Regression", John Wiley & Sons, Inc, 1989.

[26] Harvey, Andrew C., "Time Series Models", Second Edition, The MIT Press Cambridge, Massachusetts, 1994.

[27] Hardle, Wolfgan and Chen, Ron "Nonparametric Time Series Analysis, a selective review with examples", 1996.

[28] Heiler, Siegfried "A Survey on Nonparametric Time Series Analysis", 1998.

[29] Hollander, Myles and Wolfe, Douglas "Nonparametric Statistical Methods", John Wiley & Sons, 1973.

[30] Kelly, Al and Pohl, Ira, "C by Dissection" (Second Edition), The Essentials of C Programming, 1992.

[31] Kahanger, David, Moler, Cleve and Nash, Stephen, "Numerical Methods and Software", Prentice-Hall, Inc., 1989.

[32] Kantz, Holger and Schreiber, Thomas "Nonlinear Time Series Analysis", Cambridge Nonlinear Science Series 7, 1997.

[33 Lange, Kenneth and Sinsheimer, Janet S., "Normal/Independent Distribution and Their Application in Robust Regression", Journal of Computational and Graphical Statistics, Vol. 2, June 1993.

[34] Morrison, Donald F., " Applied Statistical Methods", Prentice-Hall, Inc., 1983.

[35] McCullaph, P. and Nelder, J. A., "Generalized Linear Models" (Second Edition), Chapman and Hall, 1989.

[36] Mathsoft "S-PLUS 2000", Modem Statistics and Advanced Graphics, 2000.

[37] Noether, Gottfried E., "Introduction to Statistics, The Nonparametric Way", Springer-Verlag, 1991.

[38] Puri, Madan Lal and Sen, Pranab Kumar, "Non-parametric Methods in General Linear Models", 1985.

[39] Pandit, S. M. and Wu, S.M., "Time Series and System Analysis with Applications", 1983.

[40] Rummel, S. E, "A Procedure for obtaining a Robust Regression Employing the Greatest Deviation Correlation Coefficient" Unpublished Ph.D. Dissertation, Department of Mathematical Science, University of Montana, 1991.

[41] Rousseeuw, Peter J. and Leroy, Annick M., "Robust Regression and Outlier Detection", Wiley Series in Probability and Mathematical Statistics, 1987.

[42] Searle, S. R., "Linear Models", John Wiley & Sons, 1971.

[43] Staudte, Robert G. and Sheather, Simon J., "Robert Estimation and Testing", Wiley Series in Probability and Mathematical Statistics, 1990.

[44] Wilson, J. Holton and Keating, Barry "Business Forecasting", Irwin/McGraw-Hill, 1998.

[45] Young, N., "An Introduction to Hilbert Space", Cambridge Mathematical Textbooks, 1988.

Appendix 1. C Programs for Estimation in Generalized Linear Models and Non-linear Models

```
#include<stdio.h>
#include"Glim.h"
#include<math.h>
double exp(double x);

void main()
{
  int choice;
  header();
  do
  {
   do
   {
    printf("\n\nChoose an operation by number.\n\n");
    printf("\t**********1. Glimrg(x,y)---Poisson Distribution************\n");
    printf("\t**********2. Logirg(x,y)---Logistic Regression*************\n");
    printf("\t**********3. Nonlrg(x,y)---Nonlinear Regression************\n");
    printf("\t**********4. Linerg(x,y)---Linear Regression***************\n");
    printf("\t**********5. Multrg(x,y)---Multiple Regression*************\n");

    printf("\t**********0. Quit             **************************\n\n");
    scanf("%d", &choice);
   }
   while ( (choice < 0) || (choice > 5) );
   switch (choice)
   {
       case1:
       glim_calculation();
       break;
       case 2:
       log_calculation();
          break;
       case 3:
       nonlin_calculation();
       break;
       case 4:
          Linerg_calculation();
          break;
       case 5:
          Multrg_calculation();
          break;
```

```
        default:
        choice = 0;
        break;
    }
  }
  while (choice != 0);
}

void header(void)
{
printf("\n*********************************************************************
*\n");
  printf("* This C program uses rg subroutines.                    \n");
  printf("* The C program is used for Generalized Linear Model, Logistic  \n");
  printf("*  Regression  and  Nonlinear Model with rg by iteration.       \n");
  printf("* First choose which calculation you want to execute, then input \n");
  printf("* the data from a data file.                        \n");
  printf("* Department of Mathematical Sciences                          \n");
  printf("* University of Montana                             \n");
  printf("* Missoula, MT 59812                                \n");

printf("**********************************************************************
\n");
}

double* logis(double** x,double *y, double* b,double* bb, int* n)
{
int i, ii, j,k,m=0,cnt=1;
double **res,a,c,d,e,aa,aa1,aa2,rr,*ress,*b1,*b0,*bi;
double **xstar,*zstar,ak,bk,bk1,ck,sum,**res1;
double exp(double x);
res=(double **)malloc((n[0])*sizeof(double*));
res1=(double **)malloc((n[0])*sizeof(double*));
zstar=(double *)malloc((n[0])*sizeof(double));
xstar=(double **)malloc((n[0])*sizeof(double*));
ress=(double *)malloc((n[0])*sizeof(double));
b0=(double *)malloc(n[1]*sizeof(double));
b1=(double *)malloc(n[1]*sizeof(double));
bi=(double *)malloc(n[1]*sizeof(double));
for(i=0;i<n[0];++i){
res[i]=(double*)malloc(2*sizeof(double));
res1[i]=(double*)malloc(2*sizeof(double));
}
for(i=0;i<n[1];++i){
b1[i]=b[i];
```

```
}
while(cnt && (m<2000)) {
ii=0;
for(i=0;i<n[1];++i){
b[i]=b1[i];
}
for(i=0;i<n[0];++i){
xstar[i]=(double*)malloc((n[1])*sizeof(double));
res[i]=(double*)malloc(2*sizeof(double));
}
for(i=0;i<n[0];++i){
xstar[i][0]=(exp(-b[0]-b[1]*x[i][0]))/pow(1.+exp(-b[0]-b[1]*x[i][0]),2.0);
xstar[i][1]=(x[i][0]*exp(-b[0]-b[1]*x[i][0]))/pow(1+exp(-b[0]-b[1]*x[i][0]),2.0);
zstar[i]=y[i]-1./(1.+exp(-b[0]-b[1]*x[i][0]));
}
for(i=0;i<n[1];++i){
b0[i]=b1[i];
}
for(i=0;i<n[1];++i){
bi[i]=bb[i];
}
for(i=0;i<n[1];++i){
b1[i]=bi[i]+b0[i];
}
printf("\nslope estimation: \n");
for(i=0;i<n[1];++i){
printf("\tb1[%d]=%lf\n",i,b1[i]);
}
for(i=0;i<n[0];++i){
res[i][1]=y[i]*x[i][1];
}
for(i=0;i<n[0];++i){
res[i][1]=res[i][1]-x[i][1]/(1.+exp(-b1[0]-b1[1]*x[i][0]));
res[i][0]=x[i][0];
}

ak=rgave(res,n[0]);
printf("\nrgave(x,res)=%lf",ak);
for(i=0;i<n[0];++i){
ress[i]=res[i][1];
}
sum=0.;
for(i=0;i<n[0];i++){
sum = sum+ res[i][1];
}
```

```
/*printf("\nsum=%lf\n",sum);*/

Qsort(ress,&n[0]);
for(i=0;i<n[0];++i){
res1[i][0]=i+1;
if(ress[i]<0)  res1[i][1]=-ress[i];
if(ress[i]>=0) res1[i][1]=ress[i];
}
bk=rgave(res1,n[0]);
printf("\nrgave(e,|sort(res)|)=%lf\n",bk);
ii=0;
k=0;
cnt=0;
aa=b1[0]-b0[0];
if(aa<0) aa=-aa;
while(k<n[1] ) {
aa1 = b1[k]-b0[k];
aa2 = b1[k]-b0[k];
if(aa2<0.0) aa2=-aa2;
if(aa1<0.0)  aa1=-aa1;
if(aa1>aa) aa=aa1;
if(ak<0) ak=-ak;
ck=bk1*bk;
if(ak>0.001 || ck>0. )
cnt=1;
k++;
}
bk1=bk;
m++;
printf("\n\nthe %d step of iteration:\n",m);
}
free(res);
return b1;
}

void  log_calculation()
{
int i,j, n[2],cc,j1,j2;
double **x, **xi,*res,*ress,*b,*bb;
FILE *xfp;
char filename[20];
printf("Enter the data file name for matrix x and vecter y:\n\n");
scanf("%s",filename);
xfp= fopen(filename,"r");
if(xfp == NULL){
```

```
printf("Error in opening: %s\n",filename);
exit(1);
}
fscanf(xfp,"%d",&n[0]);
fscanf(xfp,"%d",&n[1]);
b=(double*)malloc(n[1]*sizeof(double));
bb=(double*)malloc(n[1]*sizeof(double));
for(i=0;i<n[1];++i)
fscanf(xfp,"%lf",&b[i]);
for(i=0;i<n[1];++i)
fscanf(xfp,"%lf",&bb[i]);
x=(double**)malloc(n[0]*sizeof(double*));
xi=(double**)malloc(n[0]*sizeof(double*));
res=(double*)malloc(n[0]*sizeof(double));
ress=(double*)malloc(n[0]*sizeof(double));
for(i=0;i<n[0];++i){
x[i]=(double*)malloc(n[1]*sizeof(double));
for(j=0;j<n[1];++j)
fscanf(xfp,"%lf",&x[i][j]);
fscanf(xfp,"%lf",&ress[i]);
res[i]=ress[i]/x[i][n[1]-1];
}
printf("filename is :  %s\n\n",filename);
b=logis(x,res,b,bb,n);
printf("\nslope estimation:\n");
for(i=0;i<n[1];++i)
printf("\t b[%d] = %lf\n",i,b[i]);
fclose(xfp);
free(x);
free(res);
free(b);
}

double* nonlin(double** x, double *y, double *b,double* bb, int* n)
{
int i, ii,cc,j,k,m=0,cnt=1;
double **res,a,c,d,e,aa,aa1,aa2,rr,*ress,*b1,*b0,*bi;
double **xstar,*zstar,ak,ak1,bk,bk1,ck,sum,**res1;
double exp(double x);
res=(double **)malloc((n[0])*sizeof(double*));
res1=(double **)malloc((n[0])*sizeof(double*));
zstar=(double *)malloc((n[0])*sizeof(double));
xstar=(double **)malloc((n[0])*sizeof(double*));
ress=(double *)malloc((n[0])*sizeof(double));
b0=(double *)malloc((n[1])*sizeof(double));
```

```
b1=(double *)malloc((n[1])*sizeof(double));
bi=(double *)malloc((n[1])*sizeof(double));
for(i=0;i<n[0];++i){
res[i]=(double*)malloc(2*sizeof(double));
res1[i]=(double*)malloc(2*sizeof(double));
}
for(i=0;i<n[1];++i){
b1[i]=b[i];
}
for(i=0;i<n[0];++i){
xstar[i]=(double*)malloc((n[1])*sizeof(double));
res[i]=(double*)malloc(2*sizeof(double));
                         }
printf("\nWhich calculation you want to choose?(enter 1,2)\n");
printf("\t********************* 1. example 1  ****************\n");
printf("\t********************* 2. example 2  ****************\n");
scanf("%d",&cc);
while(cnt && (m<2000)) {
ii=0;
for(i=0;i<n[1];++i){
b[i]=b1[i];
}
if(cc==1){
for(i=0;i<n[0];++i){
xstar[i][0]=1.-exp(-b[1]*(x[i][0]-8.));
xstar[i][1]=-(0.49-b[0])*(x[i][0]-8.)*exp(-b[1]*(x[i][0]-8.));
zstar[i]=y[i]-b[0]-(0.49-b[0])*exp(-b[1]*(x[i][0]-8.));
}
}

 if(cc==2){
for(i=0;i<n[0];++i){
xstar[i][0]=-x[i][0]*exp(-b[1]/x[i][0]);
xstar[i][1]=b[0]*exp(-b[1]/x[i][0]);
zstar[i]=y[i]-x[i][0]*(1.-b[0]*exp(-b[1]/x[i][0]));
}
}

for(i=0;i<n[1];++i){
b0[i]=b1[i];
}
for(i=0;i<n[1];++i){
bi[i]=bb[i];
printf("\nbi[%d]=%lf\n",i,bi[i]);
}
/*bi = gsrg(xstar,zstar,bb,n);*/

for(i=0;i<n[1];++i){
b1[i]=bi[i]+b0[i];
}
for(i=0;i<n[1];++i){
printf("\tb[%d]=%lf\n",i,b1[i]);
}
for(i=0;i<n[0];++i){
res[i][0]=x[i][0];
}
for(i=0;i<n[0];++i){
for(j=0;j<n[1];++j){
res[i][1]=zstar[i]-xstar[i][j]*bb[j];
}
}
sum=0.;
for(i=0;i<n[0];++i){
sum+=res[i][1];
}
printf("\nsum of residuals=%lf\n",sum);
ii=0;
k=0;
cnt=0;
while(k<n[1]){
for(j=0;j<n[0];++j){
res[j][0]=xstar[j][k];
}
ak=rgave(res,n[0]);
ck=ak*ak1;
printf("\nrgave(xi,y-f-z*theta)=%lf",ak);
if(ak<0) ak=-ak;
if(ak>0.001)
cnt=1;
b1[i]=bi[i]+b0[i];
k++;
}
ak1=ak;
m++;
printf("\n\nthe %d step of iteration: \n",m);
}
free(res);
return b1;
}
```

```
void nonlin_calculation()
 {
int i,j,n[2],cc,j1,j2;
double **x,**xi,*res,*ress,*b,*bb;
FILE *xfp;
char filename[20];
printf("Enter the data file name for matrix x and vector y:\n\n");
scanf("%s",filename);
xfp=fopen(filename,"r");
if(xfp== NULL){
printf("Error in opening:%s\n",filename);
exit(1);
}
fscanf(xfp,"%d",&n[0]);
fscanf(xfp,"%d",&n[1]);
b=(double*)malloc((n[1])*sizeof(double));
bb=(double*)malloc((n[1])*sizeof(double));
for(i=0;i<n[1];++i)
fscanf(xfp,"%lf",&b[i]);
for(i=0;i<n[1];++i)
fscanf(xfp,"%lf",&bb[i]);
x=(double**)malloc(n[0]*sizeof(double*));
xi=(double**)malloc(n[0]*sizeof(double*));
res=(double*)malloc(n[0]*sizeof(double));
ress=(double*)malloc(n[0]*sizeof(double));
for(i=0;i<n[0];++i){
x[i]=(double*)malloc((n[1]-1)*sizeof(double));
for(j=0;j<n[1]-1;++j)
fscanf(xfp,"%lf",&x[i][j]);
fscanf(xfp,"%lf",&ress[i]);
res[i]=ress[i];
}
printf("filename is : %s\n\n",filename);
b=nonlin(x,res,b,bb,n);
printf("\nslope estimation:\n");
for(i=0;i<n[1];++i)
printf("\t b[%d]=%lf\n",i,b[i]);
fclose(xfp);
free(x);
free(res);
free(b);
}
```

```
double rg(int* ranks,int N)
{
     register I=0;
     int *M, *w, k= N-1;
     int num1l=0, num2l=0, mnum1l=0, mnum2l=0;
     M=(int*)malloc(N*sizeof(int));
     w=(int*)malloc(N*sizeof(int));
     for ( ;I< N; ++I) {
          M[ranks[I]-1] = I;
          w[I] = 0;
        }
     for(I=0; I<=k ; ++I)
     for ( I =0; I< k; ++I) {
     w[M[I]] =1;
     num1l -= w[I];
     num2l -= w[k-I];
     if(M[I] >= I)
          num1l +=1;
     if( k - M[I] >= I)
          num2l +=1;
     mnum1l = (mnum1l > num1l) ? mnum1l:num1l;
     mnum2l = (mnum2l > num2l) ? mnum2l:num2l;
    }
  /*free(M);
  free(w);*/
  return (mnum2l - mnum1l)/((double)(N/2));
}

/* Most positive rg correlation with possible tied values. */
double rgpos(double** x,int n)
{ /* n is sample size, x is a matrix and its first column is the x vector,
    the second is the y vector.   */

  int i,j,*ypos;
  Data *Apos;

  ypos = (int*)malloc(n*sizeof(int));
  Apos = (Data*)malloc(n*sizeof(struct data));

  for(i = 0; i < n; ++i){   /* assign values to data structure */
    Apos[i].L = x[i][0];
    Apos[i].R = x[i][1];
    Apos[i].n = i+1;
   }
   Qsortpos(Apos,&n,1);      /*sort on first element, if tied,look at
second */
   for(i = 0; i < n; ++i){
     Apos[i].n = i+1;         /*initial all third elements as 1:n */
   }

   Qsortpos(Apos,&n,2);      /*sort on second element, if tied.look at
first */

  for(i = 0; i < n; ++i) {    /*initial all second elements as 1:n */
     Apos[i].R = i+1;
   }
  Qsortpos(Apos,&n,3);      /*sort on third element, if tied. look at
first.*/

  for(i = 0; i < n ; ++i){
   ypos[i] = Apos[i].R;
  }
  return rg(ypos, n);
 }

/* Most negative rg correlation with possible tied values. */
double rgneg(double** x, int n)
{ /* n is sample size, x is a matrix and its first column is the x vector,
    the second is the y vector.   */

   int i,j,*yneg;
   double neg;
   Data *Aneg;

   yneg = (int*)malloc(n*sizeof(int));
   Aneg = (Data*)malloc(n*sizeof(struct data));

   for(i = 0; i < n; ++i){    /* assign values to data structure */
     Aneg[i].L = x[i][0];
     Aneg[i].R = x[i][1];
     Aneg[i].n = i+1;
   }

    Qsortneg(Aneg,&n,1);      /* sort on first ele.,if tied,look at
 second */
    for(i = 0; i < n; ++i){
      Aneg[i].n = i+1;
    }

   Qsortneg(Aneg,&n,2);      /* sort on second ele. if tied,look at
first. */

  for(i = 0; i < n; ++i) {  /*initial all second elements as 1:n */
     Aneg[i].R = i+1;
   }
  Qsortneg(Aneg,&n,3);       /* sort on third ele. if tied,look at
second.*/
  for(i = 0; i < n ; ++i){
   yneg[i] = Aneg[i].R;
  }
  return rg(yneg,n);
 }

/* rg correlation with possible tied values. */
double rgave(double** x, int n)
{ /* n is sample size, x is a matrix and its first column is the x vector,
    the second is the y vector.  */
  double pos,neg;
  pos = rgpos(x,n);
  neg = rgneg(x,n);
  return (pos+neg)/2;
}


/* do rg simple regression,estimation of rg slope and intercept. */
double  rgrg(double ** x, int n)
{ /* n is sample size, x is a matrix and its first column is the x vector,
    the second is the y vector.  */

  int i,j;
  int k,m,M, I,R,cnt =0;
  double  a,b,**res, *z;
  res = (double **)malloc(n*sizeof(double*));
  for(i = 0; i< n; ++i)
    res[i] = (double*)malloc(2*sizeof(double));
  z = (double* )malloc((n*(n-1)/2)*sizeof(double));

  for( i = 0; i < n - 1; ++i) {
     for(j = i+1; j < n; ++j) {
        if(x[i][0] == x[j][0])
           cnt++;
        else {
```

```
        k =  n*i - ((i+1)*i)/2 + (j - i) - cnt - 1;          z[k] = (x[i][1] - x[j][1])/(x[i][0] -
x[j][0]);
          }
        }
   }
  k++;
  Qsort(z,&k);
  m = k;
  k = 0;
  for( i = 1; i < m; ++i){    /* delete the tied values  */
     if(z[i] != z[k]){
        z[++k] = z[i];
          }
   }
  i = (n-1)/4;          /* skip impossible left 0 solution point. */
  m = k - (n-1)/4;       /* skip impossible right 0 solution point. */
  /* calculate the first 0 solution point. */
  while(i < m && a != 0)
   {
     b = (z[i]+z[m])/2;
     for(j = 0; j < n; ++j) {
      res[j][0] = x[j][0];
      res[j][1] = x[j][1] - b*x[j][0];
     }
     a = rgave(res,n);
     if(a<0)
        while(z[m] > b) m--;
     if(a >0)
        while(z[i] < b) i++;
   }

   R = m; /* known closest nonzero solution right point. */
   while( i < m) {         /*bisection method to get the left solution point */
    b = (z[i]+z[m])/2;
      for(j = 0; j< n; ++j){
       res[j][0] = x[j][0];
       res[j][1] = x[j][1] - b*x[j][0];
      }
       a = rgave(res,n);

      if(a <= 0) {
           while(z[m]  > b)
              m--;
         }
      else  while(z[i] < b)
```

```
          i++;
    }

      I = m;            /* m is the left solution point. */
      M = min(m+2*n,R);  /* M is the possible closest right nonzero solution
                          point,we assume that the width of solution
                          interval is less than 2*n here.     */
    /* bisection method to get the right solution point. */
      while(I < M) {
       b = (z[I]+ z[M])/2;
       for(j = 0; j < n; ++j)
         res[j][1] = x[j][1] - x[j][0] * b;
      a = rgave(res,n);
        if(a >= 0) {
            while(z[I] < b)
              I++;
             }
        else while(z[M] > b)
              M--;
      }
      free(z);
      free(res);
      return (z[m]+z[M])/2;
    }

    double rgmean(double* x, int n)
    {
     int i,m, k[4];
     double  a = 0;
     m = n;
     Qsort(x,&m);
     k[0] = (n+1)/3; k[1] = (n+3)/3; k[2] = (2*n+2)/3; k[3] = (2*n+4)/3;
     for(i = 0; i < 4; ++i)
     a = a + (x[--k[i]]/4);
     return a;
     }


    /* do rg generalized linear regression. */
    double* glim(double** x,double *y,double* b,int* n)
    {
     int  i,ii,j,k,m=0, cnt = 1;
     double **res,a,aa,rr,*ress,*b1,*b0;
     double **w,*sw,**sqw,*z,**xstar,*zstar;
    res = (double **)malloc((n[0])*sizeof(double*));
```

```
 sqw = (double **)malloc((n[0])*sizeof(double*));
 w=   (double **)malloc((n[0])*sizeof(double*));
 sw=  (double *)malloc((n[0])*sizeof(double));
 z=  (double *)malloc((n[0])*sizeof(double));
 zstar= (double *)malloc((n[0])*sizeof(double));
 xstar= (double **)malloc((n[0])*sizeof(double*));
 ress=(double *)malloc((n[0])*sizeof(double));
 b1=(double*)malloc(n[1]*sizeof(double));
 b0=(double*)malloc(n[1]*sizeof(double));
for(i=0;i<n[1];++i){
b1[i]=b[i];
}
while( cnt && ( m <1000)) {
ii=0;
sw=(double*)malloc((n[0])*sizeof(double));
for(i=0;i<n[1];++i){
b[i]=b1[i];
}
for(i=0; i<n[0];++i) {

w[i]=(double*)malloc((n[0])*sizeof(double));
sqw[i]=(double*)malloc((n[0])*sizeof(double));

xstar[i]=(double*)malloc((n[1])*sizeof(double));
res[i]=(double*)malloc(2*sizeof(double));
}
for(i=0;i<n[0];++i){

sw[i]=0.0;

}

 for(i = 0; i < n[0]; ++i){
 sw[i]=0.0;
 for(j=0; j < n[1]; ++j) {
 sw[i]=sw[i]+x[i][j]*b[j];
}
w[i][i]=1./sw[i];
sqw[i][i]=sqrt(w[i][i]);
}

for(i=0; i<n[0]; ++i){
z[i]=y[i];
zstar[i]=sqw[i][i]*z[i];
/*res[i][1]=zstar[i];*/
```

```
for(j=0; j<n[1];++j){

xstar[i][j] = sqw[i][i]*x[i][j];        /*at step m-1  */

/*res[i][1] = res[i][1] - xstar[i][j]*b[j];*/
      }
      }
for(i=0;i<n[1];++i){
b0[i]=b[i];
printf("\nb0[%d]=%lf\n",i,b0[i]);
}
b = gsrg(xstar,zstar,b,n);
for(i=0;i<n[1];++i){

b1[i]=b[i];    /* at step m */
}
printf("\tslope estimation:\n");
for(i=0;i<n[1];++i){
printf("\tb1[%d]=%lf\n",i,b1[i]);
}
ii = 0;

/******do iteration*********************************************/

for(i=0;i<n[0];++i){
sw[i]=0.0;
}
for(i=0;i<n[0];++i){
for(j=0;j<n[1];++j){
sw[i] += x[i][j]*b1[j];
}
w[i][i]=1./sw[i];
sqw[i][i]=sqrt(w[i][i]);
}

for(i=0; i<n[0];++i){
z[i]=y[i];
zstar[i]=sqw[i][i]*z[i];
res[i][1]=zstar[i];
for(j=0;j<n[1];++j){
xstar[i][j]=sqw[i][i]*x[i][j];
}

for(j=0;j<n[0];++j){
```

```
res[j][0]=xstar[j][ii];
}

      k = 0;
      cnt = 0;
      while(k < n[1] ) {
      for(j = 0; j < n[0]; ++j){
       res[j][0] = xstar[j][k];
          }
          a=rgave(res,n[0]);

          printf("\nrgave(xstar,res)=%lf\n",a);
        printf("b1[%d]=%lf,b0[%d]=%lf\n",k,b1[k],k,b0[k]);

          aa=b1[k]-b0[k];
          if(aa < 0.0) aa=-aa;
          printf("\nerror = | b(ith step)-b(i-1 th step) |=%lf\n",aa);

       if(aa >=0.0000001)
          cnt=1;

          k++;

}
      m++;
      printf("\n\nthe %d step of iteration:\n\n\n",m);
 }

free(res);
free(w);
free(sw);
free(sqw);
free(z);
return b;
}



double* gsrg(double** x,double *y,double * b,int* n)
{
int i,ii,j,k,m=0,cnt=1;
double **res,a;

res=(double **)malloc((n[0])*sizeof(double*));
for(i=0;i<n[0];++i)
```

```
res[i]=(double*)malloc(2*sizeof(double));
for(i=0;i<n[0];++i){
res[i][1]=y[i];
for(j=0;j<n[1];++j){
res[i][1]=res[i][1]-x[i][j]*b[j];
}
}
while(cnt && (m<50)) {
ii=0;
while(ii<n[1] && cnt){
for(j=0;j<n[1];++j){
res[j][1]=res[j][1] + x[j][ii]*b[ii];
res[j][0]=x[j][ii];
}
b[ii]=rgrg(res,n[0]);
for(j=0;j<n[0];++j){
res[j][1]=res[j][1]-x[j][ii]*b[ii];
}
k=0;
cnt=0;
while(k<n[1] && (cnt== 0) ) {
if(k != ii) {
for(j=0;j<n[0];++j) {
res[j][0]=x[j][k];
}
a=rgave(res,n[0]);
if(a)
cnt++;
}
k++;
}
ii++;
}
m++;
}
free(res);
return b;
}


void clear_screen(void){
   system("clear");
  }

void glim_calculation()
```

```
{
   int i,j, n[2],cc,j1,j2,mi,ma;
   double **x,**xi,*res,*ress,*b;
   /*double **xstar,*zstar;*/
   FILE *xfp;
   char filename[20];
   printf("Enter the data file name for matrix x and vector y:\n\n");
   scanf("%s", filename);
   xfp = fopen(filename,"r");
   if(xfp == NULL){
     printf("Error in opening: %s\n",filename);
       exit(1);
    }
   fscanf(xfp."%d",&n[0]);
   fscanf(xfp,"%d",&n[1]);
   b = (double*)malloc(n[1]*sizeof(double));
   for(i =0; i < n[1]; ++i)
     fscanf(xfp,"%lf",&b[i]);


     x = (double**)malloc(n[0]*sizeof(double*));

     xi = (double**)malloc(n[0]*sizeof(double*));
     res = (double*)malloc(n[0]*sizeof(double));
     ress = (double*)malloc(n[0]*sizeof(double));
          for(i = 0; i < n[0]; ++i){
          x[i] = (double*)malloc(n[1]*sizeof(double));
          for(j = 0; j < n[1]; ++j)
          fscanf(xfp ,"%lf",&x[i][j]);
          fscanf(xfp, "%lf",&ress[i]);
          res[i] = ress[i];
         }

     printf ("filename is:    %s\n\n",filename);


     b = glim(x,ress,b,n);
     printf ("\tslope estimation:\n");
     for(i = 0; i < n[1]; ++i)
     printf("\t b[%d] = %lf\n",i,b[i]);
     fclose(xfp);
     free(x);
     free(res);
     free(b);
   }
```

```
Qsort(double *x,int* n)
{
 char done;
 int ip,lv[16],iv[16],iup,lp;
 register double y;

 lv[0] = 0;
 iv[0] = *n - 1;
 ip = 0;

 while(ip >= 0)
   {if((iv[ip] - lv[ip]) < 1)
     {ip--;
      continue; }

   lp = lv[ip] - 1;
   iup = iv[ip];
   y = x[iup];

   for(;;)
     {if((iup - lp) < 2)break;
      if(x[++lp] < y)continue;
      x[iup] = x[lp];

      for(;;)
        {if((iup-- - lp) < 2)break;
         if(x[iup] >= y)continue;
         x[lp] = x[iup];
         break; }
     }

    x[iup] = y;

    if((iup - lv[ip]) < (iv[ip] - iup))
      {lv[ip + 1] = lv[ip];
       iv[ip + 1] = iup - 1;
       lv[ip]     = iup + 1; }
    else
      {lv[ip + 1] = iup + 1;
       iv[ip + 1] = iv[ip];
       iv[ip]     = iup - 1; }
   ip++;
  }
}

Qsortpos(Data *x,int* n,int options)
{
 char done;
 int ip,lv[16],iv[16],iup,lp;
 Data y;

 lv[0] = 0;
 iv[0] = *n - 1;
 ip = 0;

 while(ip >= 0)
  {if((iv[ip] - lv[ip]) < 1)
    {ip--;
     continue; }

   lp = lv[ip] - 1;
   iup = iv[ip];
   y = x[iup];

   for(;;)
    {if((iup - lp) < 2)break;

/*using positive criteria for comparing. */
   switch(options)
   {
   case 1:
     if(x[++lp].L < y.L || (x[lp].L == y.L && x[lp].R < y.R)
      ||((x[lp].L == y.L) && (x[lp].R == y.R) && (x[lp].n < y.n)))
       continue;
     x[iup] = x[lp];
     break;
   case 2:
     if(x[++lp].R < y.R || (x[lp].R == y.R && x[lp].L < y.L)
      ||((x[lp].R == y.R) && (x[lp].L == y.L) && (x[lp].n < y.n)))
       continue;
     x[iup] = x[lp];
     break;
   case 3:
     if(x[++lp].n < y.n)continue;
     x[iup] = x[lp];
     break;
    default:
   printf("The choice of options is just 1,2 or 3\n");
     exit(1);
   }
    for(;;)
      {if((iup-- - lp) < 2)break;

/*using positive criteria for comparing. */
      switch(options)
        {
        case 1:
          if(x[iup].L > y.L || (x[iup].L == y.L && x[iup].R > y.R)
          ||((x[iup].L == y.L) && (x[iup].R == y.R) && (x[iup].n >
y.n)))
              continue;
          x[lp] = x[iup];
          break;
        case 2:
         if(x[iup].R > y.R || (x[iup].R == y.R && x[iup].L > y.L)
         ||((x[iup].R == y.R) && (x[iup].L == y.L) && (x[iup].n >
y.n)))
             continue;
          x[lp] = x[iup];
          break;
        case 3:
          if(x[iup].n > y.n)continue;
          x[lp] = x[iup];
          break;
        }
          break; }
    }

   x[iup] = y;

   if((iup - lv[ip]) < (iv[ip] - iup))
     {lv[ip + 1] = lv[ip];
      iv[ip + 1] = iup - 1;
      lv[ip]     = iup + 1; }
   else
     {lv[ip + 1] = iup + 1;
      iv[ip + 1] = iv[ip];
      iv[ip]     = iup - 1; }

    ip++;
  }
```

```
}

Qsortneg(Data *x,int* n,int options)
{
 char done;
 int ip,lv[16],iv[16],iup,lp;
 Data y;

 lv[0] = 0;
 iv[0] = *n - 1;
 ip = 0;

 while(ip >= 0)
  {if((iv[ip] - lv[ip]) < 1)
    {ip--;
     continue; }

   lp = lv[ip] - 1;
   iup = iv[ip];
   y = x[iup];

   for(;;)
    {if((iup - lp) < 2)break;

/*using negative criteria for comparing. */
      switch(options)
      {
      case 1:
       if(x[++lp].L < y.L || (x[lp].L == y.L && x[lp].R > y.R)
          ||((x[lp].L == y.L) && (x[lp].R == y.R) && (x[lp].n > y.n)))
         continue;
       x[iup] = x[lp];
       break;
      case 2:
       if(x[++lp].R < y.R || (x[lp].R == y.R && x[lp].L > y.L)
          ||((x[lp].R == y.R) && (x[lp].L == y.L) && (x[lp].n > y.n)))
         continue;
       x[iup] = x[lp];
       break;
      case 3:
       if(x[++lp].n < y.n)continue;
       x[iup] = x[lp];
       break;
      }
       for(;;)
      {if((iup-- - lp) < 2)break;

/*using negative criteria for comparing. */
      switch(options)
      {
      case 1:
         if(x[iup].L > y.L || (x[iup].L == y.L && x[iup].R < y.R)
           ||((x[iup].L == y.L) && (x[iup].R == y.R) && (x[iup].n <
y.n)))
           continue;
         x[lp] = x[iup];
         break;
      case 2:
         if(x[iup].R > y.R || (x[iup].R == y.R && x[iup].L < y.L)
          ||((x[iup].R == y.R) && (x[iup].L == y.L) && (x[iup].n <
y.n)))
         continue;
         x[lp] = x[iup];
         break;
      case 3:
         if(x[iup].n > y.n)continue;
         x[lp] = x[iup];
         break;
        }
         break; }
      }

    x[iup] = y;

    if((iup - lv[ip]) < (iv[ip] - iup))
      {lv[ip + 1] = lv[ip];
      iv[ip + 1] = iup - 1;
      lv[ip]    = iup + 1; }
    else
      {lv[ip + 1] = iup + 1;
      iv[ip + 1] = iv[ip];
      iv[ip]    = iup - 1; }

    ip++;
   }
 }
```

Header File: Glim.h

```
#include<stdio.h>
#include<math.h>
double exp(double x);
struct data {
    double L;
    double R;
    int    n;
   };
typedef struct data Data;
Qsort(double *x,int* n);
Qsortpos(Data *x,int* n, int i);
Qsortneg(Data *x,int* n,int i);

double* logis(double** x,double* y,double* b,double* bb,int* n);
double* nonlin(double** x,double* y,double* b,double* bb,int* n);
double rgpos(double** x,int n);
double rgpos(double** x,int n);
double rgave(double** x, int n);
double rgrg(double ** x, int n);
double* glim(double** x,double *y,double * b,int* n);
double* gsrg(double** x,double *y,double *b,int* n);
void rgrg_calculation();
void glim_calculation();
void log_calculation();
void nonlin_calculation();

void clear_screen(void);
void header(void);
```

Appendix 2. Computational Results and Iterative Steps using r_{gd}

```
15 2
0.55 0.40          /*initial value */
0.0005 0.0005      /*iterative step length */
-7  10  0
-6   6  1
-5   5  1
-4  12  3
-3  13  5
-2  10  1
-1  20  9
0   18  14
1   15  11
2   13  9
3   13  11
4   11  10
5    9  9
6    5  5
7   15  15
*******************************************************************
* This C program uses r_gd subroutines.
* The C program is used for Generalized Linear Model, Logistic
*  Regression  and  Nonlinear Model with r_gd by iteration.
* First choose which calculation you want to execute, then input
* the data from a data file.
* Department of Mathematical Sciences
* University of Montana
* Missoula, MT 59812
********************************************************************

Choose an operation by number.

        **********1. Glimrg(x,y)---Poisson Distribution************
        **********2. Logirg(x,y)---Logistic Regression*************
        **********3. Nonlrg(x,y)---Nonlinear Regression************
        **********4. Linerg(x,y)---Linear Regression***************
        **********5. Multrg(x,y)---Multiple Regression*************
        **********0. Quit           **************************

2
Enter the data file name for matrix x and vecter y:

log2
```

```
filename is :  log2


slope estimation:
        b1[0]=0.550500
        b1[1]=0.400500

rgave(x,res)=0.285714
rgave(e,|sort(res)|)=0.142857

the 1 step of iteration:

slope estimation:
        b1[0]=0.551000
        b1[1]=0.401000

rgave(x,res)=0.285714
rgave(e,|sort(res)|)=0.142857

the 2 step of iteration:
slope estimation:
        b1[0]=0.551500
        b1[1]=0.401500

rgave(x,res)=0.285714
rgave(e,|sort(res)|)=0.142857

the 3 step of iteration:
slope estimation:
        b1[0]=0.552000
        b1[1]=0.402000

rgave(x,res)=0.285714
rgave(e,|sort(res)|)=0.142857

the 4 step of iteration:
slope estimation:
        b1[0]=0.552500
        b1[1]=0.402500

rgave(x,res)=0.285714
rgave(e,|sort(res)|)=0.142857

the 5 step of iteration:
slope estimation:
        b1[0]=0.553000
        b1[1]=0.403000

rgave(x,res)=0.142857
rgave(e,|sort(res)|)=0.142857

the 6 step of iteration:
slope estimation:
        b1[0]=0.553500
        b1[1]=0.403500

rgave(x,res)=0.142857
rgave(e,|sort(res)|)=0.142857

the 7 step of iteration:
slope estimation:
        b1[0]=0.554000
        b1[1]=0.404000

rgave(x,res)=0.142857
rgave(e,|sort(res)|)=0.142857

the 8 step of iteration:
slope estimation:
        b1[0]=0.554500
        b1[1]=0.404500

rgave(x,res)=0.142857
rgave(e,|sort(res)|)=0.142857

the 9 step of iteration:
slope estimation:
        b1[0]=0.555000
        b1[1]=0.405000

rgave(x,res)=0.142857
rgave(e,|sort(res)|)=0.142857

the 10 step of iteration:
slope estimation:
        b1[0]=0.555500
        b1[1]=0.405500

rgave(x,res)=0.142857
rgave(e,|sort(res)|)=0.142857
```

```
the 35 step of iteration:
slope estimation:
        b1[0]=0.568000
        b1[1]=0.418000

rgave(x,res)=0.000000
rgave(e,|sort(res)|)=0.142857

the 36 step of iteration:
slope estimation:
        b1[0]=0.568500
        b1[1]=0.418500

rgave(x,res)=0.000000
rgave(e,|sort(res)|)=0.142857

the 37 step of iteration:
slope estimation:
        b1[0]=0.569000
        b1[1]=0.419000

rgave(x,res)=0.000000
rgave(e,|sort(res)|)=0.142857

the 38 step of iteration:
slope estimation:
        b1[0]=0.569500
        b1[1]=0.419500

rgave(x,res)=0.000000
rgave(e,|sort(res)|)=0.142857

the 39 step of iteration:
slope estimation:
        b1[0]=0.570000
        b1[1]=0.420000

rgave(x,res)=0.000000
rgave(e,|sort(res)|)=0.142857

the 40 step of iteration:
slope estimation:
        b1[0]=0.570500
        b1[1]=0.420500
```

```
rgave(x,res)=0.000000
rgave(e,|sort(res)|)=0.142857

the 41 step of iteration:
slope estimation:
        b1[0]=0.571000
        b1[1]=0.421000

rgave(x,res)=0.000000
rgave(e,|sort(res)|)=0.142857

the 42 step of iteration:
slope estimation:
        b1[0]=0.571500
        b1[1]=0.421500

rgave(x,res)=0.000000
rgave(e,|sort(res)|)=0.142857

the 43 step of iteration:
slope estimation:
        b1[0]=0.572000
        b1[1]=0.422000

rgave(x,res)=0.000000
rgave(e,|sort(res)|)=0.000000

the 44 step of iteration:
slope estimation:
        b[0] = 0.572000
        b[1] = 0.422000
```

Appendix 3. Guide to ACF/PACF Plots

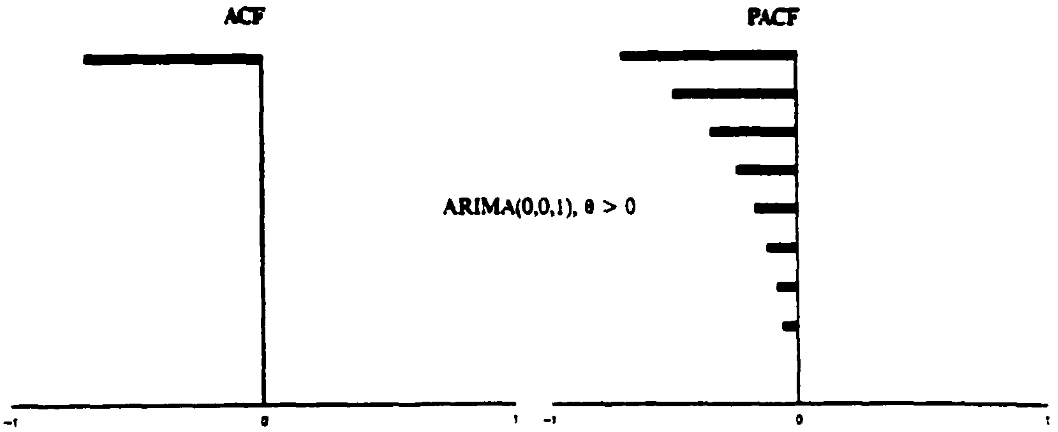
The plots shown here are those of pure or theoretical ARIMA processes. Here are some general guidelines for identifying the process(see SPSS Trends):

(1) Nonstationary time series have an ACF that remains significant for half a dozen or more lags, rather than quickly declining to zero. We must difference such a time series until it is stationary before we can identify the process.

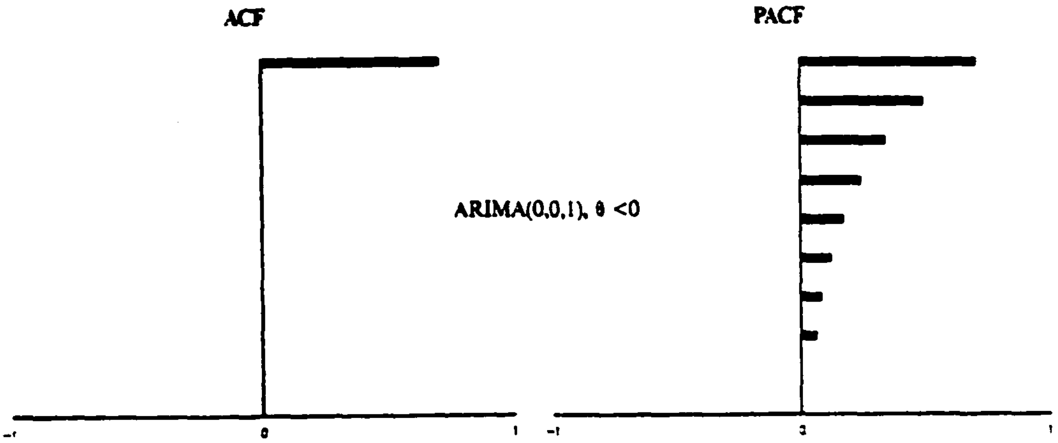
(2) Autoregressive processes have an exponentially declining ACF and spikes in the first one or more lags of the PACF. The number of spikes indicates the order of the autoregression.

(3) Moving average processes have spikes in the first one or more lags of the ACF and an exponentially declining PACF. The number of spikes indicates the order of the moving average.

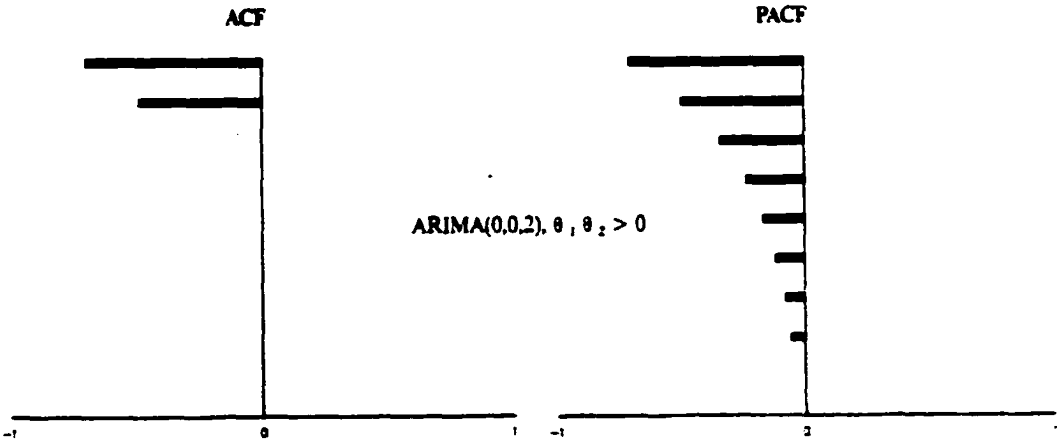
(4) Mixed (ARMA) processes typically show exponentially declines in both the ACF and the PACF.



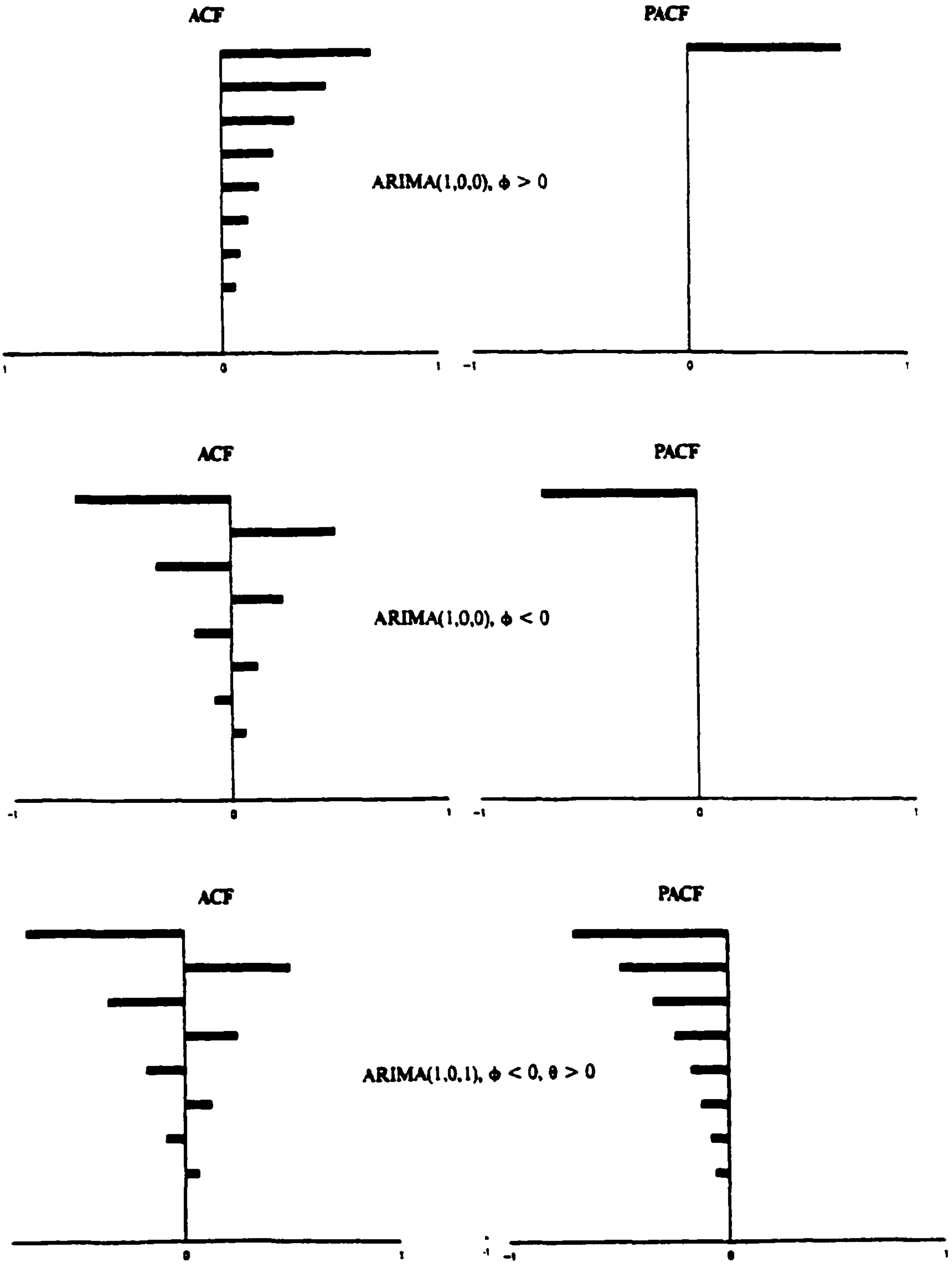
ARIMA(0,0,1), θ > 0

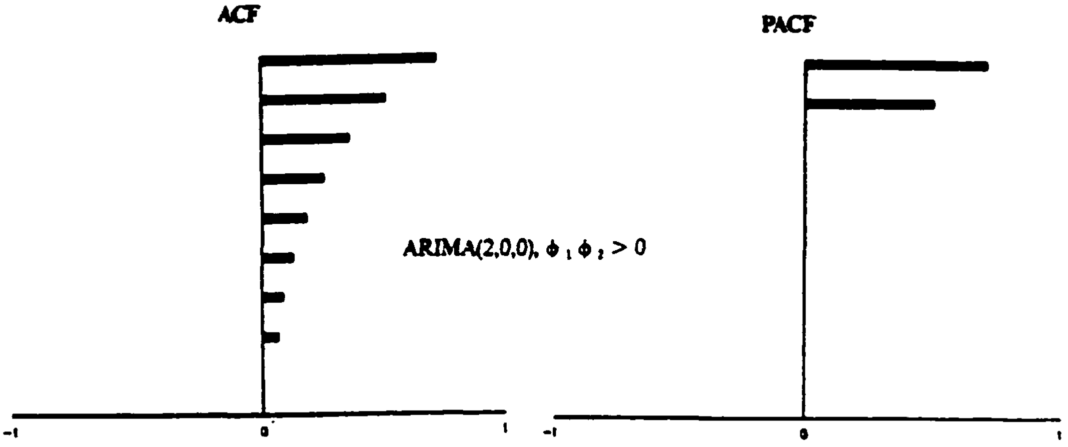


ARIMA(0,0,1), θ <0

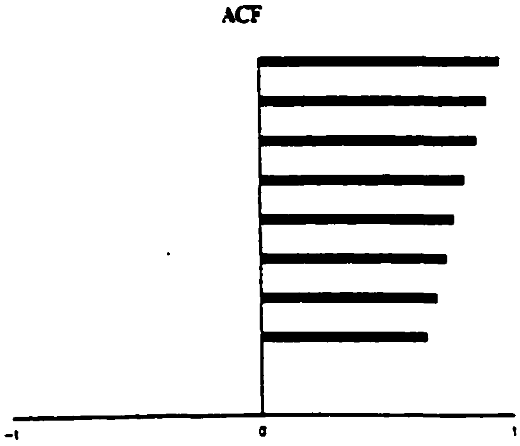


ARIMA(0,0,2), $\theta_1, \theta_2 > 0$


ACF
PACF
ARIMA(1,0,0), ϕ > 0
1
0
1
-1
0
1
ACF
PACF
ARIMA(1,0,0), ϕ < 0
-1
0
1
-1
0
1
ACF
PACF
ARIMA(1,0,1), ϕ < 0, θ > 0
0
1
-1
-1
0
1



ARIMA(2,0,0), $\phi_1, \phi_2 > 0$



ARIMA(0, 1, 0) (integrated series)

Appendix 4. S-plus Output for ARIMA Model Estimation

Computation for the 1st outlier:

```
> Data
   Date    Cost
 1 13515 23351543
 2 13546 23397885
 3 13574 25184900
 4 13605 24075128
 5 13635 24525551
 6 13666 23695725
 7 13696 23065387
 8 13727 22416962
 9 13758 24351172
10 13788 25111104
11 13819 23873198
12 13849 26747925
13 13880 24385326
14 13911 22915578
15 13939 26967128
16 13970 24575543
17 14000 23981235
18 14031 26514473
19 14061 25019793
20 14092 25453219
21 14123 24093069
22 14153 24606648
23 14184 25895048
24 14214 26482519
25 14245 26261222
26 14276 24789612
27 14304 27145030
28 14335 26845326
29 14365 25618232
30 14396 26640631
31 14426 27146923
32 14457 26172580
33 14488 26246558
34 14518 26022770
35 14549 29703957
36 14579 32942675
37 14610 27336754
38 14641 28058397
39 14670 28618759
40 14701 28379100
41 14731 29204547
42 14762 30065538
43 14792 29078126
44 14823 31302699
45 14854 29705168
46 14884 32879520
```

```
47 14915 30744989
48 14945 33428391
49 14976 33922814
50 15007 31705570
51 15035 35421359
52 15066 39493265
53 15096 34053766

> Y<-Data[,2]
> tsmatrix<-tsmatrix(Y,lag(Y),diff(Y))
> diffY<-tsmatrix[,3]
> diffY
> tsmatrix1<-tsmatrix(diffY,lag(diffY))
> YT1<-tsmatrix1[,2]
> YT<-tsmatrix1[,1]
> YT1<-matrix(YT1)

> gsrgc(YT1,YT)
$intercept:
[1] 175216.9

$slopes:
[1] -0.4060088
```

Computation for the 2nd outlier:

```
> Data
   Date    Cost
 1 13515 23351543
 2 13546 23397885
 3 13574 25184900
 4 13605 24075128
 5 13635 24525551
 6 13666 23695725
 7 13696 23065387
 8 13727 22416962
 9 13758 24351172
10 13788 25111104
11 13819 23873198
12 13849 26747925
13 13880 24385326
14 13911 22915578
15 13939 26967128
16 13970 24575543
17 14000 23981235
18 14031 26514473
19 14061 25019793
20 14092 25453219
21 14123 24093069
22 14153 24606648
23 14184 25895048
24 14214 26482519
25 14245 26261222
26 14276 24789612
```

```
27 14304 27145030
28 14335 26845326
29 14365 25618232
30 14396 26640631
31 14426 27146923
32 14457 26172580
33 14488 26246558
34 14518 26022770
35 14549 29703957
36 14579 32942675
37 14610 27336754
38 14641 28058397
39 14670 28618759
40 14701 28379100
41 14731 29204547
42 14762 30065538
43 14792 29078126
44 14823 31302699
45 14854 29705168
46 14884 32879520
47 14915 30744989
48 14945 33428391
49 14976 33922814
50 15007 31705570
51 15035 35421359
52 15066 39493265
53 15096 34053766

> Y<-Data[,2]
> tsmatrix<-tsmatrix(Y,lag(Y),diff(Y))
> diffY<-tsmatrix[,3]
> diffY
> tsmatrix1<-tsmatrix(diffY,lag(diffY))
> YT1<-tsmatrix1[,2]
> YT<-tsmatrix1[,1]
> YT1<-matrix(YT1)

> gsrgc(YT1,YT)
$intercept:
[1] 175216.9

$slopes:
[1] -0.4060088

> lsfit(YT,YT1)$coef
                Y1
Intercept  3.516350e+005
       X1 -4.508505e-001

correlation:
          Intercept        X1
Intercept  1.0000000 -0.1603314
       X1 -0.1603314  1.0000000
```