INFERENTIAL METHODS FOR BIVARIATE LOGISTIC MODEL

INFERENTIAL METHODS FOR BIVARIATE LOGISTIC MODEL

By

S.M. ENAYETUR RAHEEM, M.Sc.

A Project
Submitted to the School of Graduate Studies
in Partial Fulfillment of the Requirements
for the Degree
Master of Science

McMaster University

MASTER OF SCIENCE (2005) McMaster University
(Statistics) Hamilton, Ontario

TITLE: Inferential Methods for Bivariate Logistic Model

AUTHOR: S. M. Enayetur Raheem
B.Sc. (Honours), M.Sc. in Applied Statistics
University of Dhaka, Bangladesh

SUPERVISOR: Professor Narayanaswamy Balakrishnan

NUMBER OF PAGES: xii, 87

To

My parents–Mahfuzur Rahman & Monnu Jahan:

without their support I would not have come this far

&

my wife Rifat:

without her support this report would not have been completed in time!

Table of Contents

List of Tables

Abstract

There are several methods available for estimating the parameters of bivariate logistic model. In this report, we compare the method of Maximum Likelihood (MLM), weighted least squares cdf method (WLS), elemental percentile method (EPM) and Castillo's least square method (CLS) for estimating the parameters $\lambda, \delta, \sigma, \tau$, of bivariate logistic model. We perform Monte Carlo simulation to compare the MLM, WLS and CLS on the basis of mean squared errors (MSE) and bias of the estimators $\hat{\delta}$ and $\hat{\tau}$ by keeping $\lambda = 0$ and $\sigma = 1$ fixed. It has been found that no method is uniformly better than the others, but MLM and CLS perform better than the others in terms of MSE. We compared MLM and CLS on the basis of average confidence lengths for δ and τ. It has been found that MLM produces shorter confidence intervals than the CLS. In the CLS method, three different weights, $\beta = 0.5, 0.9, 1$, have been considered and comparative results for this method are also presented.

We applied four methods of estimation to the UK pig production data (1967-'78) as the bivariate logistic distribution has been found to be a good fit to this data (Castillo, Sarabia and Hadi 1997). We compared all four methods on the basis of MSE, bias and lengths of confidence intervals for the parameters $\lambda, \delta, \sigma, \tau$ using bootstrap resampling technique. Again, MLM and CLS are found to be performing better than the other two methods, which agrees with the results obtained using Monte Carlo simulation.

CLS has been found to be advantageous than MLM for small sample size (e.g., $n < 25$) and especially when the scale parameters are very small.

Acknowledgements

All praise goes to Almighty Allah, the most beneficent and merciful, who has given me the strength to do this research.

I would like to express my sincere gratitude to my supervisor Prof. N. Balakrishhnan, for his many suggestions and exhaustive guidance for the project to its completion.

I am thankful to Prof. Roman Viveros-Aguilera and Dr. Aaron Childs for serving on my supervisory committee and giving valuable advice. I am also grateful to Prof. P. D. M. Macdonald and the then admission committee, who opened the door for my higher studies in Canada by giving me admission in McMaster University.

Thanks are due to my friends especially Binod Prasad Neupane, Mallikarjuna Rao Rettiganti and Shahzaib Barlas for their friendship during my study at McMaster. I thank Ahmed Hossain (PhD student, Department of Public Health Science, University of Toronto) for his many support and useful guidance during my study. I would like to thank Prof. M. Sekander Hayat Khan, Prof. Pk Motiur Rahman, Prof. Syed Shahadat Hossain, Dr. Azmeri Khan, Md. Amir Hossain, Md. Shahid Ullah, Md. Asaduzzaman, and Mohammad Shahed Masud of ISRT, University of Dhaka, for their constant inspiration in pursuing my higher studies. I am grateful to Mahbub Latif (PhD student, Department of Medical Statistics, Goettingen, Germany) for his help in writing some **R** functions and to Jahrul Alam (PhD student of Mathematics, Department of Mathematics and Statistics, McMaster University) for his help in making slides using Prosper. Thanks are due to Md. Golam Faruk and my sister Mafruha Sultana for their support in buying my laptop computer which added immense flexibility in my research.

Last but not the least, I express my love to my sweet daughter Tasfia Tasneem and wife Rifat Ara Jahan for giving me excellent company and support during my stay in Canada.

Hamilton, Ontario, Canada — S.M. Enayetur Raheem
April 22, 2005

Chapter 1

An Overview

1.1 Introduction

The history of logistic distribution dates back to the mid eighteenth century when the logistic growth function was first proposed as a tool for use in demographic studies (see Balakrishnan (1992)) by Verhulst (1838) and Verhulst (1845). Reed and Berkson (1929) gave its present name. There are some other authors who used the logistic function for estimating the growth of human population; see, for example, Pearl and Reed (1920) and Schultz (1930), and more recently by Oliver (1964). Some applications of logistic function in bioassay problems were given by Pearl (1940), Wilson and Worcester (1943). The function was applied in the analysis of survival data by Plackett (1959) and Fisk (1961) used it in studying the distribution of income. Apart from the applications in growth studies, Dyke and Patterson (1952) and Grizzle (1961) applied it in public health research.

"Although multivariate data sets with logistic-like marginals have always been around" (Arnold 1992), it was not until Gumbel (1961) who proposed bivariate logistic model. Gumbel (1961) proposed three bivariate logistic distributions, the first of which takes the simple form

$$F_{X,Y}(x,y) = [1 + e^{-x} + e^{-y}]^{-1}, \quad x, y \in \mathbb{R}. \tag{1.1.1}$$

The bivariate logistic distribution is such that both marginal distributions are logistic. This can be shown by letting $y \to \infty$ in (1.1.1) to get $F_X(x) = [1+e^{-x}]^{-1}$ and similarly by letting $x \to \infty$ in (1.1.1) to get $F_Y(y) = [1+e^{-y}]^{-1}$. The function in (1.1.1) cannot be written as a product of the marginal distribution functions and therefore, the variables X, Y are not independent.

1.2 Objective of the Study

We intend to compare various estimation methods for the bivariate logistic model. Specifically, our objective is to

- Compare between MLM, CLS, WLS and EPM on the basis of bias and mean squared error of the estimators,
- Compare MLM and CLS on the basis of percentile bootstrap confidence interval.

1.3 Organization of the Report

This report is organized into seven chapters. In the first chapter, we outline the project. Bivariate logistic distribution is discussed briefly in the second chapter. We derive moment generating function for this distribution and present an algorithm for generating samples from this distribution.

In Chapter 3, we elaborate four methods of estimation for the bivariate logistic distribution. These are the maximum likelihood method, weighted least squares method, elemental percentile method, and a method based on least squares as proposed by Castillo et al. (1997).

Simulation algorithms and results are discussed in detail in Chapter 4. We compare the estimation methods on the basis of MSE, bias and length of the confidence intervals for the parameters. Results of simulation are presented at the end of this chapter.

Chapter 5 deals with comparison of the methods of estimation. We compare CLS method for various weights $\beta = 0.5, 0.9, 1$ and sample size $n = 25, 50, 100, 200$. Comparative results for the other methods are also presented.

The four methods of estimation are applied to a real-life data and the results are discussed in Chapter 6. We also perform bootstrap resampling to compare these methods and the corresponding results are presented in this chapter.

Finally, some conclusions are made in Chapter 7.

1.4 List of Notations

$BL(\lambda, \delta, \sigma, \tau)$	:	Bivariate logistic distribution with parameters $\lambda, \delta, \sigma, \tau$
Beta(m, n)	:	Beta function with parameters m and n
boot-p	:	Bootstrap percentile confidence interval
cdf	:	Cumulative distribution function
CLS	:	Castillo's least squares method
ecfd	:	Elemental cumulative distribution function
EPM	:	Elemental percentile method
iid	:	Independent and identically distributed
MGF	:	Moment generating function
MLE	:	Maximum likelihood estimate
MLM	:	Maximum likelihood method
MSE	:	Mean squared error
pdf	:	Probability density function
V-C	:	Variance-covariance matrix
WLS	:	Weighted least squares
Γ	:	Gamma function

Chapter 2

Bivariate Logistic Distribution

In this chapter, we briefly discuss the bivariate logistic distribution. Density and distribution functions are presented and moment generating function (MGF) is derived using the conditional distribution. In the end, we present a method of generating samples from bivariate logistic distribution.

2.1 Density and Distribution Functions

The cdf of a standard bivariate logistic distribution in its reduced form is given by

$$F_{X,Y}(x,y) = \left[1 + e^{-x} + e^{-y}\right]^{-1}; \quad x, y \in \mathbb{R} \tag{2.1.1}$$

and the joint pdf of (X, Y) is obtained by differentiating (2.1.1) w.r.t. x, y as

$$f(x,y) = \frac{2e^{-x}e^{-y}}{(1 + e^{-x} + e^{-y})^3}; \quad x, y \in \mathbb{R}. \tag{2.1.2}$$

The cdf of a bivariate logistic distribution $BL(\lambda, \sigma, \delta, \tau)$ is given by

$$F_{X,Y}(x,y;\boldsymbol{\theta}) = \left[1 + \exp\left(-\frac{x-\lambda}{\sigma}\right) + \exp\left(-\frac{y-\delta}{\tau}\right)\right]^{-1}; \quad x, y \in \mathbb{R} \tag{2.1.3}$$

where $\boldsymbol{\theta} = (\lambda, \delta, \sigma, \tau)$, $-\infty < \lambda, \delta < \infty$ are location parameters and $\sigma, \tau > 0$ are scale parameters. The joint pdf of (X, Y) can be obtained by differentiating the joint cdf in (2.1.3) w.r.t. x and y, which is:

$$f(x,y;\lambda,\delta,\sigma,\tau) = \frac{2e^{-(x-\lambda)/\sigma}e^{-(y-\delta)/\tau}}{\sigma\tau[1+e^{-(x-\lambda)/\sigma}+e^{-(y-\delta)/\tau}]^3}, \quad x,y \in \mathbb{R}. \tag{2.1.4}$$

The marginal distributions are obtained by letting $y \to \infty$ and $x \to \infty$ in (2.1.3) giving, respectively,

$$F_X(x;\lambda,\sigma) = \left[1+\exp\left(-\frac{x-\lambda}{\sigma}\right)\right]^{-1}, \quad x \in \mathbb{R} \tag{2.1.5}$$

$$F_Y(y;\delta,\tau) = \left[1+\exp\left(-\frac{y-\delta}{\tau}\right)\right]^{-1}, \quad y \in \mathbb{R}. \tag{2.1.6}$$

Hence, the marginal distributions in standard form are

$$F_X(x) = \left[1+e^{-x}\right]^{-1}, \quad x \in \mathbb{R} \tag{2.1.7}$$

$$F_Y(y) = \left[1+e^{-y}\right]^{-1}, \quad y \in \mathbb{R}. \tag{2.1.8}$$

Marginal pdfs can be obtained by differentiating the marginal cdfs in (2.1.5) and (2.1.6) giving

$$f(x) = \frac{1}{\sigma}\frac{e^{-(x-\lambda)/\sigma}}{[1+e^{-(x-\lambda)/\sigma}]^2}; \quad x \in \mathbb{R} \tag{2.1.9}$$

$$f(y) = \frac{1}{\tau}\frac{e^{-(y-\delta)/\tau}}{[1+e^{-(y-\delta)/\tau}]^2}; \quad y \in \mathbb{R}. \tag{2.1.10}$$

2.2 Conditional Density Function

The conditional density functions are defined as usual by

$$f(x|y) = f(x,y)/f(y); \quad f(y|x) = f(x,y)/f(x)$$

giving (in standard form)

$$\begin{aligned} f(x|y) &= \frac{f(x,y)}{f(y)} \\ &= \frac{2e^{-x}e^{-y}}{(1+e^{-x}+e^{-y})^3} \div \frac{e^{-y}}{(1+e^{-y})^2} \\ &= \frac{2e^{-x}(1+e^{-y})^2}{(1+e^{-x}+e^{-y})^3}. \end{aligned} \tag{2.2.1}$$

Similarly, it can be shown that

$$f(y|x) = \frac{2e^{-y}(1+e^{-x})^2}{(1+e^{-x}+e^{-y})^3}. \tag{2.2.2}$$

2.3 Moment Generating Function

Gumbel (1961) obtained moment generating function (MGF) of bivariate logistic distribution using conditional moment generating function approach. Conditional MGF, $M(t_1|y)$, is defined as

$$M(t_1|y) = \int_{-\infty}^{\infty} e^{xt_1} f(x|y)dx, \tag{2.3.1}$$

and the bivariate moment generating function $M(t_1, t_2)$ is defined as

$$M(t_1, t_2) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)e^{xt_1+yt_2}dxdy. \tag{2.3.2}$$

If the conditional MGF in (2.3.1) has been evaluated, the bivariate MGF can be obtained as

$$M(t_1, t_2) = \int_{-\infty}^{\infty} f(y)e^{yt_2}M(t_1|y)dy. \tag{2.3.3}$$

Substituting $f(x|y)$ in (2.3.1), we get

$$\begin{aligned} M(t_1|y) &= \int_{-\infty}^{\infty} e^{xt_1}\frac{2e^{-x}(1+e^{-y})^2}{(1+e^{-x}+e^{-y})^3}dx \\ &= 2(1+e^{-y})^2\int_{-\infty}^{\infty}\frac{e^{xt_1}e^{-x}}{(1+e^{-x}+e^{-y})^3}dx. \end{aligned} \tag{2.3.4}$$

Letting

$$\frac{1+e^{-y}}{1+e^{-x}+e^{-y}} = z, \tag{2.3.5}$$

we get

$$
\begin{aligned}
& \frac{(1+e^{-y})e^{-x}}{(1+e^{-x}+e^{-y})^2}dx = dz \\
\Rightarrow\ & \frac{ze^{-x}}{1+e^{-x}+e^{-y}}dx = dz \\
\Rightarrow\ & \frac{e^{-x}z^2}{1+e^{-y}}dx = dz \\
\Rightarrow\ & e^{-x}dx = (1+e^{-y})\frac{dz}{z^2}.
\end{aligned} \tag{2.3.6}
$$

Again, from (2.3.5), we get

$$
\begin{aligned}
& 1+e^{-y} = z(1+e^{-x}+e^{-y}) \\
\Rightarrow\ & ze^{-x} = (1+e^{-y})(1-z) \\
\Rightarrow\ & e^{x} = \left(\frac{z}{1-z}\right)(1+e^{-y})^{-1}.
\end{aligned} \tag{2.3.7}
$$

Substituting for e^x and $e^{-x}dx$, (2.3.4) becomes

$$
\begin{aligned}
M(t_1|y) &= 2(1+e^{-y})^2\int_0^1\left(\frac{z}{1-z}\frac{1}{1+e^{-y}}\right)^{t_1}(1+e^{-y})\left(\frac{1+e^{-y}}{z}\right)^{-3}\frac{dz}{z^2} \\
&= 2(1+e^{-y})^{-t_1}\int_0^1\left(\frac{z}{1-z}\right)^{t_1}zdz \\
&= 2(1+e^{-y})^{-t_1}\int_0^1 z^{t_1+1}(1-z)^{-t_1}dz \\
&= 2(1+e^{-y})^{-t_1}\int_0^1 z^{(2+t_1)-1}(1-z)^{(1-t_1)-1}dz \\
&= 2(1+e^{-y})^{-t_1}\mathrm{Beta}(2+t_1,1-t_1) \\
&= (1+e^{-y})^{-t_1}\Gamma(2+t_1)\Gamma(1-t_1).
\end{aligned}
$$

Thus, from (2.3.2), we get

$$
\begin{aligned}
M(t_1,t_2) &= \int_{-\infty}^{\infty} f(y)e^{yt_2}(1+e^{-y})^{-t_1}\Gamma(2+t_1)\Gamma(1-t_1)dy \\
&= \Gamma(2+t_1)\Gamma(1-t_1)\int_{-\infty}^{\infty} e^{-y}e^{yt_2}(1+e^{-y})^{-(2+t_1)}dy.
\end{aligned}
$$

Let

$$1/(1+e^{-y}) = u \quad \Rightarrow e^{-y}(1+e^{-y})^{-2}dy = du. \quad \text{Also } e^{y} = u/(1-u).$$

Thus, the moment generating function of bivariate logistic distribution is

$$\begin{aligned}
M(t_1, t_2) &= \Gamma(2+t_1)\Gamma(1-t_1) \int_0^1 \left(\frac{u}{1-u}\right)^{t_2} u^{t_1} du \\
&= \Gamma(2+t_1)\Gamma(1-t_1) \int_0^1 u^{t_1+t_2}(1-u)^{-t_2} du \\
&= \Gamma(2+t_1)\Gamma(1-t_1) \int_0^1 u^{(1+t_1+t_2)-1}(1-u)^{(1-t_2)-1} du \\
&= \Gamma(2+t_1)\Gamma(1-t_1) \ \mathrm{Beta}(1+t_1+t_2, 1-t_2) \\
&= \Gamma(2+t_1)\Gamma(1-t_1) \frac{\Gamma(1+t_1+t_2)\Gamma(1-t_2)}{\Gamma(1+t_1+t_2+1-t_2)} \\
&= \Gamma(1+t_1+t_2)\Gamma(1-t_1)\Gamma(1-t_2).
\end{aligned}$$

2.4 Generating Samples from $BL(\lambda, \delta, \sigma, \tau)$

In the following chapters, we will estimate the parameters of $BL(\lambda, \delta, \sigma, \tau)$ and perform simulations to obtain MSE and bias of the estimators. Therefore, we need to generate samples from $BL(\lambda, \delta, \sigma, \tau)$. The following theorem suggested by Castillo et al. (1997) will be useful to generate samples from $BL(\lambda, \delta, \sigma, \tau)$. The proof is presented here in detail.

Theorem 1. *Let U and V be two independent uniform $U(0,1)$ random variables; then (X, Y) defined by*

$$X = \lambda - \sigma \log\left(\frac{1}{U} - 1\right), \tag{2.4.1}$$

$$Y = \delta - \tau \log\left(\frac{1}{U\sqrt{V}} - \frac{1}{U}\right), \tag{2.4.2}$$

has a bivariate logistic distribution $BL(\lambda, \delta, \sigma, \tau)$.

Proof. Let

$$\begin{aligned} F(X) &= U, \text{ and} & (2.4.3) \\ F(Y|X) &= V & (2.4.4) \end{aligned}$$

From (2.4.3), we get

$$\begin{aligned} & F(X) = U = [1 + e^{-(X-\lambda)/\sigma}]^{-1} \\ \Rightarrow \quad & 1 + e^{-(X-\lambda)/\sigma} = 1/U \\ \Rightarrow \quad & e^{-(X-\lambda)/\sigma} = 1/U - 1 \\ \Rightarrow \quad & -\left(\frac{X-\lambda}{\sigma}\right) = \log(1/U - 1) \\ \Rightarrow \quad & \lambda - X = \sigma \log(1/U - 1) \\ \Rightarrow \quad & X = \lambda - \sigma \log(1/U - 1). \end{aligned} \qquad (2.4.5)$$

We can express the conditional cdf as follows

$$\begin{aligned} F_{Y|X}(y|X = x) &= \int_{-\infty}^{y} f(t|x)dt \\ &= \int_{-\infty}^{y} \frac{f(t,x)}{f_X(x)} dt \\ &= \frac{1}{f_X(x)} \int_{-\infty}^{y} f(t,x)dt \\ &= \frac{1}{f_X(x)} \int_{-\infty}^{y} \frac{\partial^2 F(t,x)}{\partial t \partial x} dt \\ &= \frac{1}{f_X(x)} \frac{\partial}{\partial x} \int_{-\infty}^{y} \frac{\partial F(t,x)}{\partial t} dt \\ &= \frac{1}{f_X(x)} \frac{\partial}{\partial x} F(y,x) \\ &= \frac{\partial F(y,x)/\partial x}{\partial F_X(x)/\partial x}. \end{aligned} \qquad (2.4.6)$$

From (2.4.4) and using (2.4.6), it can be shown that

$$F_{Y|X}(y|X = x) = \frac{\partial F(y,x)/\partial x}{\partial F_X(x)/\partial x} = \frac{[1 + e^{-(X-\lambda)/\sigma}]^2}{[1 + e^{-(X-\lambda)/\sigma} + e^{-(Y-\delta)/\tau}]^2} = V,$$

from which we get

$$
\begin{aligned}
& \left[1 + e^{-(X - \lambda)/\sigma} + e^{-(Y - \delta)/\tau}\right]^2 = \frac{1}{U^2 V} \\
\Rightarrow\ & [1 + e^{-(X - \lambda)/\sigma} + e^{-(Y - \delta)/\tau}] = \frac{1}{U\sqrt{V}} \\
\Rightarrow\ & e^{-(Y - \delta)/\tau} = \frac{1}{U\sqrt{V}} - \frac{1}{U} \quad \text{since } F(X) = U = [1 + e^{-(X-\lambda)/\sigma}]^{-1} \\
\Rightarrow\ & Y = \delta - \tau \log\left(\frac{1}{U\sqrt{V}} - \frac{1}{U}\right). \qquad (2.4.7)
\end{aligned}
$$

Chapter 3

Methods of Estimation

There are several methods available for estimating the parameters of bivariate logistic distribution $BL(\lambda, \delta, \sigma, \tau)$. We discuss four such methods, namely, the method of maximum likelihood (MLM), weighted least squares cdf (WLS) method, the elemental percentile method (EPM), and a method based on least squares proposed by Castillo et al. (1997). In this, and in the later chapters, we will denote the least squares method as Castillo's least square (CLS) method.

3.1 Maximum Likelihood Method (MLM)

This is the most widely used method of parameter estimation and is based on maximizing the likelihood of the observed sample. We use this method to find the point estimates of the parameters of $BL(\lambda, \delta, \sigma, \tau)$. Suppose $(x_i, y_i), i = 1, 2, \ldots, n$ is an independent random sample from the bivariate distribution of (X, Y) with probability density function (pdf) $f(x, y; \boldsymbol{\theta})$ where $\boldsymbol{\theta}$ is possibly a vector of parameters. Since the variables are independent, their joint pdf is

$$L(x, y|\boldsymbol{\theta}) = \prod_{i=1}^{n} f(x_i, y_i; \boldsymbol{\theta}). \tag{3.1.1}$$

After the sample has been collected, the values of $(\mathbf{x},\mathbf{y}) = \{(x_1,y_1),(x_2,y_2),\ldots,(x_n,y_n)\}$ becomes known and the above function (3.1.1) can be considered as a function of $\boldsymbol{\theta}$ given $(\mathbf{x},\mathbf{y})$ and is written as

$$L(\boldsymbol{\theta}|x,y) = \prod_{i=1}^{n} f(x_i, y_i; \boldsymbol{\theta}). \tag{3.1.2}$$

Sometimes, it is easier to deal with the *loglikelihood* of the function (which is the logarithm of the function in (3.1.2)). The loglikelihood function is given by

$$\ell(\boldsymbol{\theta}|x,y) = \log L(\boldsymbol{\theta}|x,y) = \sum_{i=1}^{n} \log f(x_i, y_i; \boldsymbol{\theta}). \tag{3.1.3}$$

The maximum likelihood estimate (MLE) of $\boldsymbol{\theta}$ is obtained by maximizing the likelihood function in (3.1.2), or equivalently, the loglikelihood function in (3.1.3), with respect to $\boldsymbol{\theta}$. We denote MLE of $\boldsymbol{\theta}$ by $\hat{\boldsymbol{\theta}}$. Thus,

$$\max_{\boldsymbol{\theta}} \ell(\boldsymbol{\theta}|x,y) = \ell(\hat{\boldsymbol{\theta}}|x,y). \tag{3.1.4}$$

If there exists a regular relative maximum $\hat{\boldsymbol{\theta}}$, the maximum likelihood estimator is obtained by solving the system of equations

$$\frac{\partial \ell(\boldsymbol{\theta}|x,y)}{\partial \theta_j} = 0, \quad j = 1,2,\ldots,k, \tag{3.1.5}$$

where $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_k)$.

3.1.1 Maximum Likelihood Estimation of $BL(\lambda,\delta,\sigma,\tau)$

Let $(x_1,y_1),\ldots,(x_n,y_n)$ be a random sample from a bivariate logistic distribution with joint pdf given in (2.1.4), where $\boldsymbol{\theta} = (\lambda,\delta,\sigma,\tau)$, $-\infty < \lambda, \delta < \infty$ are location parameters and $\sigma, \tau > 0$ are scale parameters. The likelihood function is given by

$$\begin{aligned} L(\lambda,\delta,\sigma,\tau) &= \prod_{i=1}^{n} f(x_i, y_i; \lambda,\delta,\sigma,\tau) \\ &= \frac{2^n e^{-n(\bar{x}-\lambda)/\sigma} e^{-n(\bar{y}-\delta)/\tau}}{\sigma^n \tau^n \prod_{i=1}^{n} \left[1 + e^{-(x_i-\lambda)/\sigma} + e^{-(y_i-\delta)/\tau}\right]^3}, \end{aligned} \tag{3.1.6}$$

where $\bar{x}$ denotes the mean of $x_1, x_2, \ldots, x_n$ and $\bar{y}$ denotes the mean of $y_1, y_2, \ldots, y_n$. The log-likelihood function is

$$\begin{aligned}\log L = \ell(\lambda, \delta, \sigma, \tau | x, y) &= n \log 2 - \frac{n(\bar{x} - \lambda)}{\sigma} - \frac{n(\bar{y} - \delta)}{\tau} \\ &\quad -n \log \sigma - n \log \tau - 3 \sum_{i=1}^{n} \log \left[1 + e^{-(x_i - \lambda)/\sigma} + e^{-(y_i - \delta)/\tau}\right].\end{aligned}$$

Now removing the constant term $n \log 2$, the log-likelihood function becomes

$$-\frac{n(\bar{x} - \lambda)}{\sigma} - \frac{n(\bar{y} - \delta)}{\tau} - n \log \sigma - n \log \tau - 3 \sum_{i=1}^{n} \log \left[1 + e^{-(x_i - \lambda)/\sigma} + e^{-(y_i - \delta)/\tau}\right]. \quad (3.1.7)$$

The partial derivatives of $\ell(\lambda, \delta, \sigma, \tau)$ with respect to $\lambda, \delta, \sigma, \tau$ are

$$\frac{\partial \ell(\lambda, \delta, \sigma, \tau)}{\partial \lambda} = \frac{n}{\sigma} - \frac{3}{\sigma} \sum_{i=1}^{n} \frac{e^{-(x_i - \lambda)/\sigma}}{1 + e^{-(x_i - \lambda)/\sigma} + e^{-(y_i - \delta)/\tau}}, \quad (3.1.8)$$

$$\frac{\partial \ell(\lambda, \delta, \sigma, \tau)}{\partial \delta} = \frac{n}{\tau} - \frac{3}{\tau} \sum_{i=1}^{n} \frac{e^{-(y_i - \delta)/\tau}}{1 + e^{-(x_i - \lambda)/\sigma} + e^{-(y_i - \delta)/\tau}}, \quad (3.1.9)$$

$$\frac{\partial \ell(\lambda, \delta, \sigma, \tau)}{\partial \sigma} = \frac{n(\bar{x} - \lambda)}{\sigma^2} - \frac{n}{\sigma} - \frac{3}{\sigma^2} \sum_{i=1}^{n} \frac{(x_i - \lambda) e^{-(x_i - \lambda)/\sigma}}{1 + e^{-(x_i - \lambda)/\sigma} + e^{-(y_i - \delta)/\tau}}, \quad (3.1.10)$$

$$\frac{\partial \ell(\lambda, \delta, \sigma, \tau)}{\partial \tau} = \frac{n(\bar{y} - \delta)}{\tau^2} - \frac{n}{\tau} - \frac{3}{\tau^2} \sum_{i=1}^{n} \frac{(y_i - \delta) e^{-(y_i - \delta)/\tau}}{1 + e^{-(x_i - \lambda)/\sigma} + e^{-(y_i - \delta)/\tau}}. \quad (3.1.11)$$

The MLEs $\hat{\lambda}, \hat{\delta}, \hat{\sigma}, \hat{\tau}$ of $\lambda, \delta, \sigma, \tau$ can be obtained by simultaneously solving the equations $\partial \ell(\lambda, \delta, \sigma, \tau)/\partial \lambda = 0$, $\partial \ell(\lambda, \delta, \sigma, \tau)/\partial \delta = 0$, $\partial \ell(\lambda, \delta, \sigma, \tau)/\partial \sigma = 0$, and $\partial \ell(\lambda, \delta, \sigma, \tau)/\partial \tau = 0$. Note that these equations can not be solved analytically and hence numerical methods must be employed. Newton-Raphson or some other type of iteration process can be used. Alternatively, we can use any optimization package to maximize the log-likelihood equation and obtain the MLEs for a given data. We have used **R** (R Development Core Team 2004) computational environment to compute the MLEs by maximizing the loglikelihood function in (3.1.7).

3.1.2 Score Functions and Fisher Information Matrix

Let $(\mathbf{x}, \mathbf{y}) = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ ba a random sample from the bivariate density function $f(x, y|\boldsymbol{\theta})$, where $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_k)$. Then the Fisher information matrix $I_n(\theta)$ with sample size n is based on the expected values of the second order partial derivatives, and is given by

$$I_n(\boldsymbol{\theta})_{i,j} = -E\left[\frac{\partial^2 \ln f(\mathbf{x}, \mathbf{y}|\boldsymbol{\theta})}{\partial\theta_i \partial\theta_j}\right]. \tag{3.1.12}$$

Strictly, this definition corresponds to the *expected* Fisher information. If taking the expectation is not possible or very complicated we can obtain a data-dependent quantity that is called the *observed* Fisher information.

For the bivariate logistic distribution, second order derivatives are not mathematically tractable and hence we resort to the observed Fisher information. The asymptotic variance-covariance matrix of the MLE can be obtained by inverting the observed Fisher Information matrix I_o evaluated at the MLEs of $\lambda, \delta, \sigma, \tau$. The observed Fisher Information matrix is given by

$$I_o = -\begin{pmatrix} \frac{\partial^2 \log L}{\partial\lambda^2} & \frac{\partial^2 \log L}{\partial\lambda\partial\delta} & \frac{\partial^2 \log L}{\partial\lambda\partial\sigma} & \frac{\partial^2 \log L}{\partial\lambda\partial\tau} \\ \frac{\partial^2 \log L}{\partial\delta\partial\lambda} & \frac{\partial^2 \log L}{\partial\delta^2} & \frac{\partial^2 \log L}{\partial\delta\partial\sigma} & \frac{\partial^2 \log L}{\partial\delta\partial\tau} \\ \frac{\partial^2 \log L}{\partial\sigma\partial\lambda} & \frac{\partial^2 \log L}{\partial\sigma\partial\delta} & \frac{\partial^2 \log L}{\partial\sigma^2} & \frac{\partial^2 \log L}{\partial\sigma\partial\tau} \\ \frac{\partial^2 \log L}{\partial\tau\partial\lambda} & \frac{\partial^2 \log L}{\partial\tau\partial\delta} & \frac{\partial^2 \log L}{\partial\tau\partial\sigma} & \frac{\partial^2 \log L}{\partial\tau^2} \end{pmatrix}_{(\hat{\lambda},\hat{\delta},\hat{\sigma},\hat{\tau})} \tag{3.1.13}$$

Second order derivatives are obtained using Maple (2003) and are as follows

$$\begin{aligned} \frac{\partial^2 \log L}{\partial\lambda^2} &= -\frac{3}{\sigma^2}\sum_{i=1}^{n}\left[\frac{e^{-(x_i-\lambda)/\sigma}}{1 + e^{-(x_i-\lambda)/\sigma} + e^{-(y_i-\delta)/\tau}}\right. \\ &\quad \left. - \frac{e^{-2(x_i-\lambda)/\sigma}}{(1 + e^{-(x_i-\lambda)/\sigma} + e^{-(y_i-\delta)/\tau})^2}\right], \end{aligned} \tag{3.1.14}$$

$$\begin{aligned} \frac{\partial^2 \log L}{\partial\delta^2} &= -\frac{3}{\tau^2}\sum_{i=1}^{n}\left[\frac{e^{-(y_i-\delta)/\tau}}{1 + e^{-(x_i-\lambda)/\sigma} + e^{-(y_i-\delta)/\tau}}\right. \\ &\quad \left. - \frac{e^{-2(y_i-\delta)/\tau}}{[1 + e^{-(x_i-\lambda)/\sigma} + e^{-(y_i-\delta)/\tau}]^2}\right], \end{aligned} \tag{3.1.15}$$

$$
\begin{aligned}
\frac{\partial^2 \log L}{\partial \sigma^2} &= \frac{2n(\bar{X}-\lambda)}{\sigma^3} + \frac{n}{\sigma^2} \\
&\quad - 3\sum_{i=1}^{n}\left[\frac{(x_i-\lambda)^2 e^{-(x_i-\lambda)/\sigma}}{\sigma^4(1+e^{-(x_i-\lambda)/\sigma}+e^{-(y_i-\delta)/\tau})}\right. \\
&\quad - \frac{2(x_i-\lambda)e^{-2(x_i-\lambda)/\sigma}}{\sigma^3\left[1+e^{-(x_i-\lambda)/\sigma}+e^{-(y_i-\delta)/\tau}\right]} \\
&\quad \left. - \frac{(x_i-\lambda)^2 e^{-2(x_i-\lambda)/\sigma}}{\sigma^4\left[1+e^{-(x_i-\lambda)/\sigma}+e^{-(y_i-\delta)/\tau}\right]^2}\right],
\end{aligned} \tag{3.1.16}
$$

$$
\begin{aligned}
\frac{\partial^2 \log L}{\partial \tau^2} &= -\frac{2n(\bar{y}-\delta)}{\tau^3} + \frac{n}{\tau^2} \\
&\quad - 3\sum_{i=1}^{n}\left[\frac{(y_i-\delta)^2 e^{-(y_i-\delta)/\tau}}{\tau^4\left[1+e^{-(x_i-\lambda)/\sigma}+e^{-(y_i-\delta)/\tau}\right]}\right. \\
&\quad - \frac{2(y_i-\delta)e^{-2(y_i-\delta)/\tau}}{\tau^3\left[1+e^{-(x_i-\lambda)/\sigma}+e^{-(y_i-\delta)/\tau}\right]} \\
&\quad \left. - \frac{(y_i-\delta)^2 e^{-2(y_i-\delta)/\tau}}{\tau^4\left[1+e^{-(x_i-\lambda)/\sigma}+e^{-(y_i-\delta)/\tau}\right]^2}\right],
\end{aligned} \tag{3.1.17}
$$

$$
\frac{\partial^2 \log L}{\partial \lambda \partial \delta} = \frac{3}{\sigma\tau}\sum_{i=1}^{n}\frac{e^{-(x_i-\lambda)/\sigma}e^{-(y_i-\delta)/\tau}}{\left[1+e^{-(x_i-\lambda)/\sigma}+e^{-(y_i-\delta)/\tau}\right]^2}, \tag{3.1.18}
$$

$$
\begin{aligned}
\frac{\partial^2 \log L}{\partial \lambda \partial \sigma} &= -\frac{n}{\sigma^2} - 3\sum_{i=1}^{n}\left[\frac{(x_i-\lambda)e^{-(x_i-\lambda)/\sigma}}{\sigma^3\left[1+e^{-(x_i-\lambda)/\sigma}+e^{-(y_i-\delta)/\tau}\right]}\right. \\
&\quad - \frac{e^{-(x_i-\lambda)/\sigma}}{\sigma^2\left[1+e^{-(x_i-\lambda)/\sigma}+e^{-(y_i-\delta)/\tau}\right]} \\
&\quad \left. - \frac{(x_i-\lambda)e^{-2(x_i-\lambda)/\sigma}}{\sigma^3\left[1+e^{-(x_i-\lambda)/\sigma}+e^{-(y_i-\delta)/\tau}\right]^2}\right],
\end{aligned} \tag{3.1.19}
$$

$$
\frac{\partial^2 \log L}{\partial \lambda \partial \tau} = \frac{3}{\sigma\tau^2}\sum_{i=1}^{n}\frac{(y_i-\delta)e^{-(x_i-\lambda)/\sigma}e^{-(y_i-\delta)/\tau}}{\left[1+e^{-(x_i-\lambda)/\sigma}+e^{-(y_i-\delta)/\tau}\right]^2}, \tag{3.1.20}
$$

$$
\frac{\partial^2 \log L}{\partial \delta \partial \sigma} = \frac{3}{\sigma^2\tau}\sum_{i=1}^{n}\frac{(x_i-\lambda)e^{-(x_i-\lambda)/\sigma}e^{-(y_i-\delta)/\tau}}{\left[1+e^{-(x_i-\lambda)/\sigma}+e^{-(y_i-\delta)/\tau}\right]^2}, \tag{3.1.21}
$$

$$
\begin{aligned}
\frac{\partial^2 \log L}{\partial \delta \partial \tau} &= -3\sum_{i=1}^{n}\Bigg[\frac{(y_i-\delta)e^{-(y_i-\delta)/\tau}}{\tau^3\left[1+e^{-(x_i-\lambda)/\sigma}+e^{-(y_i-\delta)/\tau}\right]} \\
&\quad - \frac{e^{-(y_i-\delta)/\tau}}{\tau^2\left[1+e^{-(x_i-\lambda)/\sigma}+e^{-(y_i-\delta)/\tau}\right]} \\
&\quad - \frac{(y_i-\delta)e^{-2(y_i-\delta)/\tau}}{\tau^3\left[1+e^{-(x_i-\lambda)/\sigma}+e^{-(y_i-\delta)/\tau}\right]^2}\Bigg], \qquad (3.1.22)
\end{aligned}
$$

$$
\frac{\partial^2 \log L}{\partial \sigma \partial \tau} = \frac{3}{\sigma^2\tau^2}\sum_{i=1}^{n}\frac{(x_i-\lambda)(y_i-\delta)e^{-(x_i-\lambda)/\sigma}e^{-(y_i-\delta)/\tau}}{\left[1+e^{-(x_i-\lambda)/\sigma}+e^{-(y_i-\delta)/\tau}\right]^2}. \qquad (3.1.23)
$$

3.2 Weighted Least Squares CDF Method (WLS)

Let (X, Y) be a bivariate random variable with cdf $F_{(X,Y)}(x, y;\ \boldsymbol{\theta})$, where $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_k)$ is a possibly vector-valued parameter and $(x_1, y_1), \ldots, (x_n, y_n)$ is a sample from F. Consider

$$
p^{xy} = \frac{m^{xy} - 0.5}{n}, \qquad (3.2.1)
$$

where m^{xy} = number of points in the sample where $X \leq x$ and $Y \leq y$. The parameter $\boldsymbol{\theta}$ is then estimated by

$$
\min_{\boldsymbol{\theta}} \sum_{i=1}^{n} \frac{n}{p^{x_iy_i}(1-p^{x_iy_i})}\left(F_{X,Y}(x_i, y_i; \boldsymbol{\theta}) - p^{x_iy_i}\right)^2, \qquad (3.2.2)
$$

where the factors $\frac{n}{p^{x_iy_i}(1-p^{x_iy_i})}$ are the weights that account for the variance of the different terms. This is why this method is called *weighted least squares cdf* method.

3.2.1 Application of WLS to $BL(\lambda, \delta, \sigma, \tau)$

Substituting the bivariate cdf in (3.2.2), we get

$$
\min_{\boldsymbol{\theta}} \sum_{i=1}^{n} \frac{n}{p^{x_iy_i}(1-p^{x_iy_i})}\left[(1+e^{-(x-\lambda)/\sigma}+e^{-(y-\delta)/\tau})^{-1} - p^{x_iy_i}\right]^2. \qquad (3.2.3)
$$

Equation (3.2.3) can be minimized for a given data using any optimization package. We have used **R** to optimize it for the parameters of $BL(\lambda, \delta, \sigma, \tau)$.

3.3 Castillo's Least Squares Method (CLS)

Castillo et al. (1997) proposed this method based on least squares. The main idea of this method is to write the predicted values as a function of the parameter, say, $\boldsymbol{\theta}$. The sum of squared deviations between the predicted and observed values are then taken. The parameter estimate of $\boldsymbol{\theta}$ is then obtained by minimizing the sum of squared deviations.

3.3.1 Description of CLS Method

Let X and Y be jointly a bivariate random variable with cdf $F_{(X,Y)}(x, y; \boldsymbol{\theta})$. Let us denote the marginal cdfs of X and Y by $F_X(x, \boldsymbol{\theta})$ and $F_Y(y; \boldsymbol{\theta})$, respectively. Let

$$\begin{aligned} p^x &= \text{proportion of points in the sample where } (X \le x), \\ p^y &= \text{proportion of points in the sample where } (Y \le y), \\ p^{xy} &= \text{proportion of points in the sample where } (X \le x \text{ and } Y \le y). \end{aligned}$$

In this method, the joint and the marginal cdfs are used for calculating the predicted values as functions of $\boldsymbol{\theta}$. This can be done in two possible ways:

1. Using $F_{(X,Y)}(x_i, y_i; \boldsymbol{\theta})$ and $F_X(x; \boldsymbol{\theta})$, we have

$$\left.\begin{array}{l} F_X(x_i; \boldsymbol{\theta}) = p^{x_i} \\ F_{(X,Y)}(x_i, y_i; \boldsymbol{\theta}) = p^{x_i y_i} \end{array}\right\} \Rightarrow \left\{\begin{array}{l} \hat{x}_i(\boldsymbol{\theta}) = F_X^{-1}(p^{x_i}; \boldsymbol{\theta}), \\ \hat{y}_i(\boldsymbol{\theta}) = F_{(X,Y)}^{-1}(p^{x_i y_i}; \hat{x}_i(\boldsymbol{\theta}), \boldsymbol{\theta}), \end{array}\right. \tag{3.3.1}$$

   where $F_X^{-1}(p, \boldsymbol{\theta})$ is the inverse of $F_X(x_i, \boldsymbol{\theta})$ and $F_{(X,Y)}^{-1}(p; x_i, \boldsymbol{\theta})$ is the inverse of $F_{(X,Y)}(x_i, y_i; \boldsymbol{\theta})$ with respect to its second argument.

2. Using $F_{(X,Y)}(x_i, y_i; \boldsymbol{\theta})$ and $F_Y(y; \boldsymbol{\theta})$, we have

$$\left.\begin{array}{l} F_Y(y_i; \boldsymbol{\theta}) = p^{y_i} \\ F_{(X,Y)}(x_i, y_i; \boldsymbol{\theta}) = p^{x_i y_i} \end{array}\right\} \Rightarrow \left\{\begin{array}{l} \hat{y}_i(\boldsymbol{\theta}) = F_Y^{-1}(p^{y_i}; \boldsymbol{\theta}), \\ \hat{x}_i(\boldsymbol{\theta}) = F_{(X,Y)}^{-1}(p^{x_i y_i}; \hat{y}_i(\boldsymbol{\theta}), \boldsymbol{\theta}), \end{array}\right. \tag{3.3.2}$$

   where $F_Y^{-1}(p, \boldsymbol{\theta})$ is the inverse of $F_Y(y_i, \boldsymbol{\theta})$ and $F_{(X,Y)}^{-1}(p; y_i, \boldsymbol{\theta})$ is the inverse of $F_{(X,Y)}(x_i, y_i; \boldsymbol{\theta})$ with respect to its first argument.

Taking the weighted average of (3.3.1) and (3.3.2), we obtain the new estimates

$$\begin{aligned} \hat{x}_i(\boldsymbol{\theta}) = &\ \beta F_X^{-1}(p^{x_i};\boldsymbol{\theta}) + (1-\beta)F_{(X,Y)}^{-1}(p^{x_i y_i}; F_Y^{-1}(p^{y_i};\boldsymbol{\theta}),\boldsymbol{\theta}), \\ \hat{y}_i(\boldsymbol{\theta}) = &\ \beta F_Y^{-1}(p^{y_i};\boldsymbol{\theta}) + (1-\beta)F_{(X,Y)}^{-1}(p^{x_i y_i}; F_X^{-1}(p^{x_i};\boldsymbol{\theta}),\boldsymbol{\theta}), \end{aligned} \tag{3.3.3}$$

where $0 \leq \beta \leq 1$ is an appropriately chosen weight.

An estimator of $\boldsymbol{\theta}$ can now be obtained by minimizing

$$E = \sum_{i=1}^{n}([x_i - \hat{x}_i(\boldsymbol{\theta})]^2 + [y_i - \hat{y}_i(\boldsymbol{\theta})]^2) \tag{3.3.4}$$

with respect to $\boldsymbol{\theta}$.

3.3.2 Choosing the Weight β

There are two options of choosing the appropriate weight β :

1. We may choose β to be equal to 0.5, which will put equal weight on both parts of the expressions in (3.3.3). Taking $\beta = 0.5$, (3.3.3) reduces to

$$\begin{aligned} \hat{x}_i(\boldsymbol{\theta}) = &\ \tfrac{1}{2}\left(F_X^{-1}(p^{x_i};\boldsymbol{\theta}) + F_{(X,Y)}^{-1}(p^{x_i y_i}; F_Y^{-1}(p^{y_i};\boldsymbol{\theta}),\boldsymbol{\theta})\right), \\ \hat{y}_i(\boldsymbol{\theta}) = &\ \tfrac{1}{2}\left(F_{(X,Y)}^{-1}(p^{x_i y_i}; F_X^{-1}(p^{x_i};\boldsymbol{\theta}),\boldsymbol{\theta}) + F_Y^{-1}(p^{y_i};\boldsymbol{\theta})\right), \end{aligned} \tag{3.3.5}$$

   which is the average of (3.3.1) and (3.3.2).

2. We may choose a value of β optimally, by minimizing (3.3.4) with respect to both β and $\boldsymbol{\theta}$.

3.3.3 Finding an Optimum β

Finding the optimum weight is very simple. All we need is to find the weight β and the set of parameters for which (3.3.4) is minimized. The steps are as follows.

Step 1: Choose a value of β between 0 and 1.

Step 2: Obtain the estimates of parameters for the given data using (3.3.4).

Step 3: Finally obtain E.

Repeat these steps to obtain E for all the values of β between (0,1) e.g., 0.01, 0.02, and so on. The optimum value of β is the one for which E is minimized.

3.3.4 Application of CLS to $BL(\lambda, \delta, \sigma, \tau)$

The joint cdf and marginal cdfs of $BL(\lambda, \delta, \sigma, \tau)$ are given in (2.1.3), (2.1.5) and (2.1.6). Setting $F_{(X,Y)}(\hat{x}, \hat{y}; \boldsymbol{\theta}) = p^{xy}$ and $F_X(\hat{x}; \boldsymbol{\theta}) = p^x$, we obtain the following system of equations in $\hat{x}$ and $\hat{y}$:

$$\begin{aligned} \alpha_1 e^{-\hat{x}/\sigma} + \alpha_2 e^{-\hat{y}/\tau} &= -1 + \tfrac{1}{p^{xy}}, \\ \alpha_1 e^{-\hat{x}/\sigma} &= -1 + \tfrac{1}{p^x}, \end{aligned} \tag{3.3.6}$$

where $\alpha_1 = e^{\lambda/\sigma}$ and $\alpha_2 = e^{\delta/\tau}$, which has the following solution:

$$\begin{aligned} \hat{x} &= \lambda - \sigma \log(\tfrac{1}{p^x} - 1), \\ \hat{y} &= \delta - \tau \log(\tfrac{1}{p^{xy}} - \tfrac{1}{p^x}), \end{aligned} \tag{3.3.7}$$

provided that $p^{xy} \neq p^x$. Similarly, setting $F_{X,Y}(x, y; \boldsymbol{\theta}) = p^{xy}$ and $F_Y(y; \boldsymbol{\theta}) = p^y$, it can be shown that

$$\begin{aligned} \hat{x} &= \lambda - \sigma \log(\tfrac{1}{p^{xy}} - \tfrac{1}{p^y}), \\ \hat{y} &= \delta - \tau \log(\tfrac{1}{p^y} - 1), \end{aligned} \tag{3.3.8}$$

provided that $p^{xy} \neq p^y$.

Thus, we propose using the following equations to compute the predicted values, which are obtained by averaging (3.3.7) and (3.3.8), and replacing $\hat{x}$ and $\hat{y}$ by $\hat{x}_i$ and $\hat{y}_i$:

$$\begin{aligned} \hat{x}_i(\boldsymbol{\theta}) &= \lambda - \sigma r_i, \\ \hat{y}_i(\boldsymbol{\theta}) &= \delta - \tau s_i, \end{aligned} \tag{3.3.9}$$

where

$$r_i = \begin{cases} \beta \log(\frac{1}{p^{x_i}} - 1) + (1-\beta)\log(\frac{1}{p^{x_i y_i}} - \frac{1}{p^{y_i}}), & \text{if } p^{x_i y_i} \neq p^{y_i}, \\ \log(\frac{1}{p^{x_i}} - 1), & \text{if } p^{x_i y_i} = p^{y_i}, \end{cases}$$

$$s_i = \begin{cases} \beta \log(\frac{1}{p^{y_i}} - 1) + (1-\beta)\log(\frac{1}{p^{x_i y_i}} - \frac{1}{p^{x_i}}), & \text{if } p^{x_i y_i} \neq p^{x_i}, \\ \log(\frac{1}{p^{y_i}} - 1), & \text{if } p^{x_i y_i} = p^{x_i}, \end{cases} \tag{3.3.10}$$

where β is the weight used for the solution (3.3.3). Note that when the sample size is finite, it is possible to have $p^{x_i y_i} = p^{x_i}$ or $p^{x_i y_i} = p^{y_i}$ for some sample values. Now, we minimize, with respect to $\lambda, \delta, \sigma,$ and τ,

$$\begin{aligned} E &= \sum_{i=1}^{n} [x_i - \hat{x}(\lambda, \delta, \sigma, \tau)]^2 + [y_i - \hat{y}(\lambda, \delta, \sigma, \tau)]^2 \\ &= \sum_{i=1}^{n} [(x_i - \lambda + \sigma r_i)^2 + (y_i - \delta + \tau s_i)^2]. \end{aligned} \tag{3.3.11}$$

Taking the derivatives of E with respect to each of the parameters, we obtain

$$\begin{aligned} \frac{\partial E}{\partial \lambda} &= -2\sum_{i=1}^{n}(x_i - \lambda + \sigma r_i), \\ \frac{\partial E}{\partial \sigma} &= -2\sum_{i=1}^{n}(x_i - \lambda + \sigma r_i) r_i, \\ \frac{\partial E}{\partial \delta} &= -2\sum_{i=1}^{n}(y_i - \delta + \tau s_i), \\ \frac{\partial E}{\partial \tau} &= -2\sum_{i=1}^{n}(y_i - \delta + \tau s_i) s_i. \end{aligned} \tag{3.3.12}$$

Equating each of the above equations to zero, we obtain the following system of equations:

$$\begin{aligned}
\lambda n - \sigma \sum_{i=1}^{n} r_i &= \sum_{i=1}^{n} x_i, \\
\lambda \sum_{i=1}^{n} r_i - \sigma \sum_{i=1}^{n} r_i^2 &= \sum_{i=1}^{n} x_i r_i, \\
\delta n - \tau \sum_{i=1}^{n} s_i &= \sum_{i=1}^{n} y_i, \\
\delta \sum_{i=1}^{n} s_i - \tau \sum_{i=1}^{n} s_i^2 &= \sum_{i=1}^{n} y_i s_i.
\end{aligned} \tag{3.3.13}$$

The solution of the above equations yield the estimators:

$$\begin{aligned}
\hat{\lambda} &= \frac{\sum_{i=1}^{n} x_i r_i \sum_{i=1}^{n} r_i - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} r_i^2}{\left(\sum_{i=1}^{n} r_i\right)^2 - n \sum_{i=1}^{n} r_i^2}, \quad \hat{\sigma} = \frac{n \sum_{i=1}^{n} x_i r_i - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} r_i}{\left(\sum_{i=1}^{n} r_i\right)^2 - n \sum_{i=1}^{n} r_i^2}, \\
\hat{\delta} &= \frac{\sum_{i=1}^{n} y_i s_i \sum_{i=1}^{n} s_i - \sum_{i=1}^{n} y_i \sum_{i=1}^{n} s_i^2}{\left(\sum_{i=1}^{n} s_i\right)^2 - n \sum_{i=1}^{n} s_i^2}, \quad \hat{\tau} = \frac{n \sum_{i=1}^{n} y_i s_i - \sum_{i=1}^{n} y_i \sum_{i=1}^{n} s_i}{\left(\sum_{i=1}^{n} s_i\right)^2 - n \sum_{i=1}^{n} s_i^2}.
\end{aligned} \tag{3.3.14}$$

3.4 The Elemental Percentile Method (EPM)

Classical estimation methods such as maximum likelihood method and method of moments work well, for example, in cases where the distribution belongs to the exponential family. In many other cases, they may not exist or may be computationally difficult or they may produce unsatisfactory results. EPM was originally proposed by Castillo and Hadi (1995), for estimating the parameters and quantiles of continuous distributions. There are some advantages of this method including the fact that the estimates are unique and well-defined for all parameter and sample values. Also, the estimates exist in case where other classical estimators do not exist. This method is most useful when the distribution function and its inverse is given in closed form.

In this section, we describe the EPM for estimating the parameters and quantiles of $F(x;\boldsymbol{\theta}), \boldsymbol{\theta} \in \Theta$. The method gives well-defined estimators for all values of $\boldsymbol{\theta} \in \Theta$.

3.4.1 Description of EPM

In this method, the estimates are obtained in two steps. First, some elemental estimates are obtained by solving equations relating the cdf to their percentile values for some elemental subsets of the observations. These elemental estimates are then used to obtain statistically more efficient estimates of the parameters. The steps are described below.

Elemental Percentile Estimates

Suppose $X = \{X_1, X_2, \ldots, X_n\}$ are *iid* random variables having a common cdf $F(x, \boldsymbol{\theta})$, then we have

$$F(x_{i:n}; \boldsymbol{\theta}) \cong p_{i:n} \quad i = 1, 2, \ldots, n, \tag{3.4.1}$$

or equivalently,

$$x_{i:n} \cong F^{-1}(p_{i:n}; \boldsymbol{\theta}), \quad i = 1, 2, \ldots, n, \tag{3.4.2}$$

where $x_{i:n}$ are the order statistics and $p_{i:n}$ are empirical estimates of $F(x_i; \boldsymbol{\theta})$ or suitable plotting positions. One such plotting position is given in (3.2.1).

Let $I = \{i_1, i_2, \ldots, i_k\}$ be a set of indices of k distinct order statistics (for order statistics, see Arnold, Balakrishnan and Nagaraja (1992)). We refer to a subset of size k observations as an elemental subset and to the resultant estimates as elemental estimates of $\boldsymbol{\theta}$. For each observation in an elemental subset I, we set

$$x_{i:n} = F^{-1}(p_{i:n}; \boldsymbol{\theta}), \quad i \in I, \tag{3.4.3}$$

where we have replaced the approximation in (3.4.2) by an equality. The set I is chosen so that the system in (3.4.3) contains k independent equations in k unknowns $\boldsymbol{\theta} = \{\theta_1, \theta_2, \ldots, \theta_k\}$. An elemental estimate of $\boldsymbol{\theta}$ can then be obtained by solving (3.4.3) for $\boldsymbol{\theta}$.

Final Estimates

The estimates obtained in the first step as described in the previous section, depend on k distinct order statistics. For large n and k, the number of elemental subsets may be too large for the computations of all possible elemental estimates to be feasible. In such cases, instead of computing all possible elemental subsets, one may select a prespecified number, N, of elemental subsets either systematically, based on some theoretical considerations, or completely at random. For each of these subsets, an elemental estimate of $\boldsymbol{\theta}$ is computed. We denote these elemental estimates by $\hat{\theta}_{j1}, \hat{\theta}_{j2}, \ldots, \hat{\theta}_{jN}$, $j = 1, 2, \ldots, k$. The elemental estimates are then combined, using some suitable robust functions, to obtain an overall final estimate of $\boldsymbol{\theta}$. Examples of robust function include the median (MED) and the $\alpha-$trimmed mean (TM_α), where α indicates the percentage of trimming. Thus, a final estimate of $\boldsymbol{\theta} = \{\theta_1, \theta_2, \ldots, \theta_k\}$, can be obtained as

$$\hat{\theta}_j(\mathrm{MED}) = \mathrm{Median}(\hat{\theta}_{j1}, \hat{\theta}_{j2}, \ldots, \hat{\theta}_{jN}), \quad j = 1, 2, \ldots, k, \tag{3.4.4}$$

or

$$\hat{\theta}_j(\mathrm{TM}_\alpha) = \mathrm{TM}_\alpha(\hat{\theta}_{j1}, \hat{\theta}_{j2}, \ldots, \hat{\theta}_{jN}), \quad j = 1, 2, \ldots, k, \tag{3.4.5}$$

where $\mathrm{Median}(y_1, y_2, \ldots, y_N)$ is the median of the set of numbers $\{y_1, y_2, \ldots, y_N\}$, and $\mathrm{TM}_\alpha(y_1, y_2, \ldots, y_N)$ is the mean obtained after trimming the $(\alpha/2)\%$ largest and the $(\alpha/2)\%$ smallest order statistics of $(y_1, y_2, \ldots, y_N)$.

The MED estimators are very robust but inefficient. The TM_α estimators are less robust but more efficient than the MED estimators. The larger the trimming, the more robust and less efficient are the TM_α estimators (Castillo, Hadi, Balakrishnan and Sarabia 2005).

3.4.2 Application of EPM to $BL(\lambda, \delta, \sigma, \tau)$

The EPM described in the previous section can be easily extended for bivariate logistic distribution. Let (X, Y) be a bivariate random variable with cdf $F_{(X,Y)}(x, y;\ \boldsymbol{\theta})$, where

$\boldsymbol{\theta} = \{\theta_1, \ldots, \theta_k\}$, that is, there are k parameters in $\boldsymbol{\theta}$. Now consider a subset I_1 of k different sample points

$$I_1 = \{i_r | i_r \in \{1, 2, \ldots, n\},\ i_{r_1} \neq i_{r_2} \text{ if } r_1 \neq r_2,\ r = 1, 2, \ldots, k\},$$

and assume that the system of k equations in k unknowns $\{\theta_1, \theta_2, \ldots, \theta_k\}$

$$F_{(X,Y)}(x_{i_r}, y_{i_r}; \theta_{1r}, \theta_{2r}, \ldots, \theta_{kr}) = p^{x_{i_r} y_{i_r}}, \tag{3.4.6}$$

allow obtaining a set of elemental estimates $\{\hat{\theta}_1, \hat{\theta}_2, \ldots, \hat{\theta}_k\}$. Now we select m different sets $I_1, I_2, \ldots, I_m$ instead of just one I_1. Thus, we obtain m elemental set of estimates $\{\hat{\theta}_{1m}, \hat{\theta}_{2m}, \ldots, \hat{\theta}_{km}\}$. Finally, we select an appropriate robust estimate $\hat{\theta}_j^*$ of θ_j, $j = 1, 2, \ldots, k$ using the MED or TM_α.

In the case of bivariate logistic distribution with parameters $\lambda, \delta, \sigma, \tau$, we choose an elemental subset I_1 of $k = 4$ different sample points

$$I_1 = \{i_r | i_r \in \{1, 2, \ldots, n\},\ i_{r_1} \neq i_{r_2} \text{ if } r_1 \neq r_2,\ r = 1, 2, 3, 4\}.$$

Now using (3.4.6) the system of four equations in four unknowns $(\lambda, \delta, \sigma, \tau)$ is:

$$\begin{aligned}
F_{(X,Y)}(x_{i_1}, y_{i_1}; \lambda, \delta, \sigma, \tau) &= p^{x_{i_1} y_{i_1}} \\
F_{(X,Y)}(x_{i_2}, y_{i_2}; \lambda, \delta, \sigma, \tau) &= p^{x_{i_2} y_{i_2}} \\
F_{(X,Y)}(x_{i_3}, y_{i_3}; \lambda, \delta, \sigma, \tau) &= p^{x_{i_3} y_{i_3}} \\
F_{(X,Y)}(x_{i_4}, y_{i_4}; \lambda, \delta, \sigma, \tau) &= p^{x_{i_4} y_{i_4}}.
\end{aligned} \tag{3.4.7}$$

Replacing the cdfs of bivariate logistic distribution in (3.4.7), we get the following system of equations:

$$\begin{aligned}
1 + \exp\left(\frac{x_1 - \lambda}{\sigma}\right) + \exp\left(\frac{y_1 - \delta}{\tau}\right) &= \frac{1}{p^{x_{1_1} y_{1_1}}} \\
1 + \exp\left(\frac{x_2 - \lambda}{\sigma}\right) + \exp\left(\frac{y_2 - \delta}{\tau}\right) &= \frac{1}{p^{x_{1_2} y_{1_2}}} \\
1 + \exp\left(\frac{x_3 - \lambda}{\sigma}\right) + \exp\left(\frac{y_3 - \delta}{\tau}\right) &= \frac{1}{p^{x_{1_3} y_{1_3}}} \\
1 + \exp\left(\frac{x_4 - \lambda}{\sigma}\right) + \exp\left(\frac{y_4 - \delta}{\tau}\right) &= \frac{1}{p^{x_{1_4} y_{1_4}}}.
\end{aligned} \tag{3.4.8}$$

Elemental estimates are obtained by solving the system of equations (3.4.8). These equations are nonlinear in parameters and hence they can not be solved analytically. Any Newton-type algorithm can be used to find a solution of the system. Given a set of n equations in n unknowns, seeking a solution $r(x) = 0$ is equivalent to minimizing the sum of squares $r(x) \cdot r(x)$ when the residual is zero at the minimum. (Bates and Watts 1988). We used `optim()` in **R** to get the solution of (3.4.8).

3.4.3 An Example

Consider the following data simulated from $BL(\lambda = 3, \delta = 1, \sigma = 0.5, \tau = 0.25)$.

x:	3.69	1.94	2.68	3.43	2.78	1.37	3.6	3.71	3.65	3.20
y:	0.68	0.30	1.22	1.07	1.30	0.15	0.8	1.34	1.04	0.96

and we want to estimate the parameters. By EPM, we take an elemental subset of size four (as there are four parameters) and then obtain an elemental estimate. There are $\binom{10}{4}$=210 possible elementary subsets to choose from. Let us consider all the possible subsets to obtain the elementary estimates. For each of the subsets, we will get elementary estimates. In this way, we will have 210 elemental estimates for each of the four parameters $(\hat{\lambda}_1, \hat{\lambda}_2, \ldots, \hat{\lambda}_{210})$, $(\hat{\delta}_1, \hat{\delta}_2, \ldots, \hat{\delta}_{210})$, $(\hat{\sigma}_1, \hat{\sigma}_2, \ldots, \hat{\sigma}_{210})$, $(\hat{\tau}_1, \hat{\tau}_2, \ldots, \hat{\tau}_{210})$.

Final estimates can be obtained by using any of the functions MED or TM_α as discussed in the previous section.

3.5 Confidence Intervals

For the maximum likelihood method, we obtain the standard deviations of the estimators from the asymptotic variance-covariance matrix, which is obtained by inverting the observed Fisher information matrix. But for the other three methods, variances of the resultant estimates may not be available analytically. In such cases, an estimate of the standard deviation can be obtained using some sampling methods such as the jackknife and the bootstrap methods; (see Efron (1979), Diaconis and Efron (1974)). Since the parameter estimates are well defined for all feasible combinations of parameters and sample values, the standard error of the estimates and hence the confidence intervals for the corresponding estimates can be computed easily. The bootstrap sampling (Gomes and Oliveira 2001) can be done in two ways: the samples can be drawn with replacement directly from the data, or they can be drawn from the parametric cdf, $F(x, y; \hat{\boldsymbol{\theta}})$. However, we preferred the parametric bootstrap to obtain the variance of the estimates for a particular method.

To obtain a confidence interval for a parameter θ_j, we simulate a large number of bootstrap samples and obtain the corresponding estimates for each parameter. We use these estimates to obtain an elemental cdf (ecdf) for that parameter estimate $\hat{\theta}_j$. From each of these ecdfs, we calculate confidence intervals and calculate probability coverages.

Chapter 4

Simulation Study

We use mean squared errors (MSE) and bias to assess the performance of the estimators of the parameters of $BL(\lambda, \delta, \sigma, \tau)$ by Monte Carlo simulation. For simplicity, we generate data for only $BL(\lambda = 0, \delta, \sigma = 1, \tau)$ with $\delta = 0, 0.5, 1$ and $\tau = 0.25, 0.5, 1$. That is, we calculate MSE and bias for $\hat{\delta}$ and $\hat{\tau}$. Because parameter estimation by EPM is computationally very time consuming, we did not perform Monte Carlo simulation for this method. However, boostrap resampling has been done for all four methods using a real-life data and the results are discussed in the next chapter. In the following, simulation algorithm and results are discussed for the MLM, WLS and CLS methods.

4.1 Simulation

4.1.1 Maximum Likelihood Method

We estimate the parameters of $BL(\lambda, \delta, \sigma, \tau)$ using the method of maximum likelihood as described in Section 3.1. We generate data from bivariate logistic distribution with $\lambda = 0$, $\sigma = 1$ and $\delta = 0, 0.5, 1$, $\tau = 0.25, 0.50, 1$ for sample sizes $n = 25, 50, 100, 200$. The maximum likelihood estimates are then obtained using the methods described in Section 3.1 for the generated data. The steps are summarized below.

Step 1. Generate a sample of size n from bivariate logistic distribution with $\lambda = 0$, $\sigma = 1$, $\delta = 0, 0.5, 1$, $\tau = 0.25, 0.5, 1$.

Step 2. Obtain the MLEs of the parameters.

Step 3. Repeat steps 1 and 2 R times and calculate MSE and bias of the estimators.

The MSE and bias of $\hat{\delta}$ and $\hat{\tau}$ for different sample sizes are presented in Table 4.1 and Table 4.2, respectively.

MLM: Discussion of Results

For a fixed τ and $n = 25$, MSE($\hat{\delta}$) $= 0.00699, 0.00694, 0.00704$ for $\delta = 0.25, 0.5, 1$, respectively. This implies that for a fixed τ and n, MSE($\hat{\delta}$) does not vary according to varying δ. The behavior is same for all sample sizes. Similarly, for a fixed τ, and $n = 25$, MSE($\hat{\tau}$) $= 0.00138, 0.00142, 0.00143$ for $\delta = 0.25, 0.5, 1$, respectively. As we can see, the MSEs are very close to each other and the behavior is same for all sample sizes. For both $\hat{\delta}$ and $\hat{\tau}$, MSE decreases with the increase of sample size n, and the decrease is inversely proportional to the increase of n. As τ increases, MSE also increases. In general, location parameter $\hat{\delta}$ has a larger MSE than that of scale parameter $\hat{\tau}$.

Bias of $\hat{\delta}$ and $\hat{\tau}$ are presented in Table 4.2. We see that $\hat{\tau}$ globally underestimates τ while $\hat{\delta}$ overestimates δ in most of the cases.

4.1.2 Weighted Least Squares Method

MSE and bias of the parameters of bivariate logistic distribution have been obtained by simulation. Parameters are estimated for four different sample sizes– $n = 25, 50, 100$, and 200 for each of 9 different combinations of parameters, keeping the location $\lambda = 0$ and scale $\sigma = 1$ fixed. Other values of the parameters that are considered include: $\delta = 0, 0.5, 1$ and $\tau = 0.25, 0.50, 1$. The estimates are based on 1,000 Monte Carlo runs. The steps of MC simulation is the same as that of MLM.

WLS: Discussion of Results

For fixed τ and $n = 25$, MSE($\hat{\delta}$) $= 0.01125, 0.01212, 0.00119$ for $\delta = 0.25, 0.5, 1$, respectively. This implies that for a fixed τ and n, MSE($\hat{\delta}$) does not depend on location parameter. We observe the same behaviour for all sample size. Similarly, for a fixed τ, and $n = 25$, MSE($\hat{\tau}$) $= 0.00413, 0.00402, 0.00424$ for $\delta = 0.25, 0.5, 1$, respectively. As we can see, the MSEs are very close to each other and the behavior is same for all sample sizes. This implies– MSE depends only on the scale parameter. For both $\hat{\delta}$ and $\hat{\tau}$, MSE decreases with the increase of sample size n, and the decrease is inversely proportional to the increase of n. Also, as the value of scale parameter τ increases, MSE increases. Like the maximum likelihood method, location parameter $\hat{\delta}$ has a larger MSE than that of scale parameter $\hat{\tau}$.

Bias of $\hat{\delta}$ and $\hat{\tau}$ are presented in Table 4.4. We see that $\hat{\delta}$ and $\hat{\tau}$ globally underestimates τ and δ.

4.1.3 Castillo's Least Squares Method

After estimating the parameters using Castillo's least squares method, we calculate MSE and bias of $\hat{\delta}, \hat{\tau}$ by simulation. Results are obtained for three different weights, $\beta = 0.5, 0.9, 1$. Like the MLM and WLS, four different sample sizes ($n = 25, 50, 100, 200$) have been considered for each of 9 different combinations of (δ, τ), while keeping $\lambda = 0$ and $\sigma = 1$ fixed. The estimates are based on 1,000 Monte Carlo runs.

CLS: Discussion of Results

MSE($\hat{\delta}$), MSE($\hat{\tau}$) are presented in Tables (4.5- 4.7) and bias($\hat{\delta}$), bias($\hat{\tau}$) are presented in Tables (4.8- 4.10). Like the MLM, we observe that MSE($\hat{\delta}$) and MSE($\hat{\tau}$) does not depend on the location parameter. We observe that MSE($\hat{\delta}$) and MSE($\hat{\tau}$) increases with the increase of sample size. Also, for a given sample size, MSE increases with the increase of τ. Overall, MSE($\hat{\tau}$) is smaller than MSE($\hat{\delta}$) for any combination of (δ, τ) and β.

Bias($\hat{\delta}$), bias($\hat{\tau}$) produce negative values for most of the cases, indicating that they underestimate the parameters δ and τ, respectively.

Comparison within MSE($\hat{\delta}$) and MSE($\hat{\tau}$) for different β are presented in Table 4.11. Except on a few occasions, for both estimators, MSE decreases with increasing β. Of the three values of β, MSE is the highest for $\beta = 0.5$ and the lowest for $\beta = 1.0$.

4.1.4 Elemental Percentile Method

The steps of calculating MSE and bias of the estimators of the parameters of $BL(\lambda, \delta, \sigma, \tau)$ using elemental percentile method is given below:

1. First we choose an elemental subset from the available subsets. For large sample size, there are hundreds of such elemental subsets to choose from. In such case, we randomly select a predefined number N, of subsets.

2. For each of the N elemental subsets, we obtain elemental estimates giving N elemental set of estimates.

3. We use MED and the TM_α functions to obtain the final estimates. For TM_α function, we consider $\alpha = 10$ and 20.

This method is computationally very time consuming especially if the number of parameters to be estimated is large. In our case, we have four parameters and four different sample sizes, $n = 25, 50, 100, 200$. Because of time constraints, we did not perform Monte Carlo simulation on this method. However, we presented the results of bootstrap simulation for this method in Chapter 6.

4.2 Coverage Probability

Coverage probability of an estimator may be defined as the probability that the confidence interval based on the estimator includes the parameter of interest. One can easily calculate coverage probability by simulation. We construct pivotal quantities, say, P_i for the parameters and simulate the probability coverage

$$Pr(-1.96 \leq P_i \leq 1.96)$$

which should approximately be 95 percent. This way, we calculated probability coverage for the location parameter δ and scale parameter τ. The procedure is discussed in the following sections. In the following subsections, we discuss coverage percentage for the MLEs of δ and τ. We also present the steps of calculating probability coverage for $\hat{\delta}$ and $\hat{\tau}$ for CLS method.

4.2.1 95% Coverage Percentage For MLEs

To compute confidence intervals or to conduct tests of the hypothesis for the location and scale parameters of $BL(\lambda, \delta, \sigma, \tau)$, we need to construct pivotal quantities. Since the MLEs are asymptotically normally distributed, we have the asymptotic distribution of

$$P_{\hat{\delta}} = \frac{\hat{\delta} - \delta}{\sqrt{\mathrm{Var}(\hat{\delta})}}, \qquad P_{\hat{\tau}} = \frac{\hat{\tau} - \tau}{\sqrt{\mathrm{Var}(\hat{\tau})}} \tag{4.2.1}$$

to be standard normal. The quantities in (4.2.1) are pivotal quantities because they are functions of the data and the parameters; but their distributions do not depend on the unknown parameter. The steps of calculating probability coverage is given below:

1. For a given set of the initial values of the parameters, we generate a sample of size n from the bivariate logistic distribution.

2. Calculate the MLEs by the maximum likelihood method.

3. Obtain the asymptotic variance-covariance (V-C) matrix by inverting the Fisher information matrix evaluated at the MLEs. The diagonal elements of V-C matrix are the variance of the parameters.

4. Compute the pivotal quantities in (4.2.1). If the pivotal quantity lies between (-1.96, 1.96), we add 1 to the counter.

5. The steps 1-4 are repeated R times and probability coverage is calculated as

$$95\% \text{ Probability coverage } = \frac{\# \text{ of times } P_i \text{ is between } (-1.96, 1.96)}{R} \times 100$$

where P_i is the pivotal quantity.

Table 4.12 shows 95% probability coverage for δ and τ. For large n, probability coverage is approximately 95%.

4.2.2 95% Coverage Percentage for CLS

We have seen in the previous section that calculation of probability coverage requires variance and hence standard deviations of the estimators. But variance of the estimators are not analytically available for the WLS, CLS and EPM methods. Therefore, we use bootstrap within each Monte Carlo run to calculate the variance. The steps are described below. Monte Carlo steps are denoted by *MC Step* and bootstrap steps are denoted by *Boot Step*.

MC Step-1: For a given set of the initial values of the parameters, generate a sample of size n from bivariate logistic distribution.

MC Step-2: Obtain the estimates $\hat{\lambda}, \hat{\delta}, \hat{\sigma}, \hat{\tau}$ by Castillo's method based on least squares.

Boot Step-1: Generate bootstrap sample with the estimates obtained in MC Step-2 as the initial values.

Boot Step-2: Repeat Boot Step-2 B=999 (say) times to obtain 999 bootstrap replicates $\hat{\lambda^*}, \hat{\delta^*}, \hat{\sigma^*}, \hat{\tau^*}$.

Boot Step-3: Calculate variance of the estimators using the bootstrap replicates found in Boot Step-2.

MC Step-3: Using the variance of the estimators found in Boot Step-3, we compute pivotal quantities using (4.2.1). If the pivotal quantity lies between (-1.96, 1.96), we add 1 to the counter.

MC Step-4: Repeat MC Steps 1-3 R times and calculate probability coverage as

$$95\% \text{ Probability coverage } = \frac{\#\text{ of times } P_i \text{ is between } (-1.96, 1.96)}{R} \times 100,$$

where P_i is the pivotal quantity.

Remark 1. Table 4.13 and Table 4.14 shows 95% probability coverage for $\hat{\delta}$ and $\hat{\tau}$, respectively based on 200 Monte Carlo runs. Overall coverage is around 95%. However lower (e.g., 89.5%) or higher (e.g., 97.5%) percentages might be due to small number of Monte Carlo runs.

Table 4.1: MLM: MSE($\hat{\delta}$) and MSE($\hat{\tau}$) for $\lambda = 0$ and $\sigma = 1$ based on 1,000 Monte Carlo runs.

Parameters		MSE($\hat{\delta}$)				MSE($\hat{\tau}$)			
δ	τ	$n = 25$	$n = 50$	n=100	n=200	n=25	n=50	n=100	n=200
0	0.25	0.00699	0.00346	0.00174	0.00087	0.00138	0.00068	0.00034	0.00017
	0.50	0.02789	0.01396	0.00698	0.00339	0.00574	0.00272	0.00137	0.00071
	1.00	0.11273	0.05601	0.02841	0.01371	0.02221	0.01103	0.00542	0.00268
0.5	0.25	0.00694	0.003537	0.00183	0.00082	0.00142	0.00069	0.00031	0.00017
	0.50	0.02844	0.013636	0.00688	0.00342	0.00552	0.00239	0.00137	0.00071
	1.00	0.11689	0.055271	0.02886	0.01380	0.02347	0.01142	0.00519	0.00246
1	0.25	0.00706	0.00349	0.00174	0.00088	0.00143	0.00070	0.00034	0.00017
	0.50	0.02805	0.01387	0.00681	0.00346	0.00559	0.00276	0.00135	0.00067
	1.00	0.11170	0.05577	0.02713	0.01395	0.02296	0.01083	0.00545	0.00274

Table 4.2: MLM: Bias($\hat{\delta}$) and bias($\hat{\tau}$)for $\lambda = 0$ and $\sigma = 1$ based on 1,000 Monte Carlo runs.

Parameters		Bias($\hat{\delta}$)				Bias($\hat{\tau}$)			
δ	τ	n=25	n=50	n=100	n=200	n=25	n=50	n=100	n=200
0	0.25	0.00049	0.00189	-0.00016	0.00002	-0.00661	-0.00311	-0.00185	-0.00098
	0.50	0.00303	0.00156	0.00158	-0.00013	-0.01393	-0.00680	-0.00326	-0.00182
	1.00	0.00557	0.00400	0.00156	0.00092	-0.02651	-0.01299	-0.00613	-0.00340
0.5	0.25	0.00205	0.00389	0.00048	0.00071	-0.00905	-0.00321	-0.00210	-0.00090
	0.50	0.00090	0.00334	-0.00305	-0.00083	-0.01501	-0.00637	-0.00413	-0.00175
	1.00	0.01104	0.00388	-0.00021	0.00011	-0.02443	-0.01808	-0.00959	-0.00462
1	0.25	0.00160	0.00046	0.00014	-0.00007	-0.00658	-0.00345	-0.00179	-0.00098
	0.50	0.00201	0.00253	0.00003	0.00015	-0.13870	-0.00660	-0.00342	-0.00180
	1.00	0.00478	0.00297	0.00256	-0.00073	-0.02883	-0.01181	-0.00674	-0.00358

Table 4.3: WLS: MSE($\hat{\delta}$) and MSE($\hat{\tau}$) for $\lambda = 0$ and $\sigma = 1$ based on 1000 Monte Carlo runs.

Parameters		MSE($\hat{\delta}$)				MSE($\hat{\tau}$)			
δ	τ	n=25	n=50	n=100	n=200	n=25	n=50	n=100	n=200
0	0.25	0.01125	0.00535	0.00246	0.00122	0.00413	0.00160	0.00069	0.00036
	0.50	0.04553	0.02007	0.01039	0.00499	0.01647	0.00699	0.00332	0.00135
	1.00	0.17958	0.07870	0.03823	0.01886	0.06543	0.02843	0.01323	0.00579
0.5	0.25	0.01212	0.00539	0.00256	0.00116	0.00402	0.00165	0.00073	0.00038
	0.50	0.04594	0.02053	0.01014	0.00481	0.01685	0.00674	0.00298	0.00149
	1.00	0.19285	0.08522	0.04239	0.01790	0.06288	0.02703	0.01306	0.00577
1	0.25	0.01119	0.00496	0.00272	0.00125	0.00424	0.00169	0.00079	0.00035
	0.50	0.04490	0.01968	0.00956	0.00471	0.01636	0.00711	0.00331	0.00145
	1.00	0.17710	0.07938	0.04344	0.02000	0.06734	0.02707	0.01263	0.00554

Table 4.4: WLS: Bias($\hat{\delta}$) and bias($\hat{\tau}$)for $\lambda = 0$ and $\sigma = 1$ based on 1000 Monte Carlo runs.

Parameters		Bias($\hat{\delta}$)				Bias($\hat{\tau}$)			
δ	τ	n=25	n=50	n=100	n=200	n=25	n=50	n=100	n=200
0	0.25	-0.01147	-0.00208	-0.00097	-0.00103	-0.00548	-0.00528	-0.00518	-0.00142
	0.50	-0.01714	-0.00836	-0.00168	-0.00075	-0.02055	-0.01171	-0.00842	-0.00379
	1.00	-0.04639	-0.00382	-0.00007	-0.00648	-0.02293	-0.02043	-0.02152	-0.00742
0.5	0.25	-0.01306	-0.00423	-0.00457	-0.00175	-0.00508	-0.00499	-0.00269	-0.00374
	0.50	-0.01052	-0.01148	-0.00130	-0.00303	-0.00955	-0.00570	-0.00651	-0.00269
	1.00	-0.05000	-0.01699	-0.00065	-0.00351	-0.03193	-0.02622	-0.01485	-0.01182
1	0.25	-0.01542	-0.00291	-0.00173	-0.00192	-0.00910	-0.00422	-0.00476	-0.00124
	0.50	-0.02319	-0.00191	-0.00003	-0.00324	-0.01146	-0.01021	-0.01076	-0.00371
	1.00	-0.06081	-0.01166	-0.00692	-0.00770	-0.03595	-0.01689	-0.01902	-0.00497

Table 4.5: CLS: MSE($\hat{\delta}$) and MSE($\hat{\tau}$) for $\lambda = 0$ and $\sigma = 1$, $\beta = 0.50$ based on 1,000 Monte Carlo runs.

Parameters		MSE($\hat{\delta}$)				MSE($\hat{\tau}$)			
δ	τ	$n = 25$	$n = 50$	n=100	n=200	n=25	$n = 50$	$n = 100$	$n = 200$
0	0.25	0.00983	0.00528	0.00288	0.00157	0.00209	0.00109	0.00058	0.00033
	0.50	0.04028	0.02203	0.01130	0.00572	0.00823	0.00437	0.00231	0.00126
	1.00	0.15729	0.08265	0.04605	0.02335	0.03243	0.01770	0.00923	0.00510
0.5	0.25	0.01002	0.00538	0.00282	0.00143	0.00202	0.00111	0.00055	0.00032
	0.50	0.04037	0.02141	0.01175	0.00585	0.00815	0.00439	0.00243	0.00131
	1.00	0.15770	0.08501	0.04348	0.02162	0.03384	0.01779	0.00964	0.00515
1	0.25	0.01008	0.00543	0.00261	0.00148	0.00203	0.00110	0.00057	0.00031
	0.50	0.03989	0.08565	0.04532	0.02343	0.00832	0.00442	0.00232	0.00123
	1.00	0.16344	0.08714	0.04309	0.02472	0.03424	0.01757	0.01006	0.00497

Table 4.6: CLS: MSE($\hat{\delta}$) and MSE($\hat{\tau}$) for $\lambda = 0$ and $\sigma = 1$, $\beta = 0.90$ based on 1,000 Monte Carlo runs.

Parameters		MSE($\hat{\delta}$)				MSE($\hat{\tau}$)			
δ	τ	$n = 25$	$n = 50$	n=100	n=200	$n = 25$	$n = 50$	$n = 100$	$n = 200$
0	0.25	0.00839	0.00424	0.00220	0.00114	0.00195	0.00098	0.00051	0.00024
	0.50	0.03354	0.01706	0.00882	0.00440	0.00765	0.00396	0.00207	0.00102
	1.00	0.13590	0.06784	0.03209	0.01853	0.03110	0.01583	0.00752	0.00416
0.5	0.25	0.00840	0.00424	0.00209	0.00114	0.00190	0.00097	0.00051	0.00025
	0.50	0.03412	0.01751	0.00920	0.00445	0.00767	0.00392	0.00202	0.00099
	1.00	0.13680	0.06910	0.03522	0.01758	0.03151	0.01538	0.00804	0.00404
1	0.25	0.00853	0.00417	0.00220	0.00111	0.00192	0.00099	0.00050	0.00024
	0.50	0.03393	0.01725	0.00905	0.00464	0.00786	0.00395	0.00212	0.00106
	1.00	0.13508	0.07090	0.03375	0.01652	0.03092	0.01587	0.00749	0.00411

Table 4.7: CLS: MSE($\hat{\delta}$) and MSE($\hat{\tau}$) for $\lambda = 0$ and $\sigma = 1$, $\beta = 1.00$ based on 1,000 Monte Carlo runs.

Parameters		MSE($\hat{\delta}$)				MSE($\hat{\tau}$)			
δ	τ	$n = 25$	$n = 50$	n=100	n=200	$n = 25$	$n = 50$	$n = 100$	$n = 200$
0	0.25	0.00807	0.00417	0.00205	0.00103	0.00192	0.00096	0.00048	0.00025
	0.50	0.03318	0.01623	0.00777	0.00434	0.00787	0.00386	0.00189	0.00098
	1.00	0.13358	0.06537	0.02969	0.01774	0.03093	0.01561	0.00766	0.00376
0.5	0.25	0.00832	0.00402	0.00222	0.00104	0.00188	0.00095	0.00047	0.00022
	0.50	0.03232	0.01655	0.00837	0.00425	0.00754	0.00379	0.00174	0.00101
	1.00	0.12929	0.06642	0.03318	0.01581	0.03013	0.01536	0.00774	0.00370
1	0.25	0.00827	0.00412	0.00203	0.00100	0.00190	0.00096	0.00046	0.00025
	0.50	0.03259	0.01633	0.00800	0.00437	0.00765	0.00389	0.00184	0.00095
	1.00	0.13029	0.06445	0.03087	0.01650	0.03077	0.01555	0.00844	0.00370

Table 4.8: CLS: Bias($\hat{\delta}$) and bias($\hat{\tau}$) for $\lambda = 0$ and $\sigma = 1$, $\beta = 0.50$ based on 1,000 Monte Carlo runs.

Parameters		Bias($\hat{\delta}$)				Bias($\hat{\tau}$)			
δ	τ	$n = 25$	$n = 50$	n=100	n=200	$n = 25$	$n = 50$	n=100	n=200
0	0.25	-0.00395	-0.00467	-0.00251	-0.00522	-0.00531	-0.00408	-0.00303	-0.00166
	0.50	-0.00789	-0.00935	-0.01105	-0.00528	-0.01015	-0.00636	-0.00577	-0.00468
	1.00	-0.01250	-0.01767	-0.01212	0.00001	-0.01883	-0.01385	-0.01138	-0.01117
0.5	0.25	-0.00416	-0.00347	-0.00180	-0.00428	-0.00501	-0.00310	-0.00310	-0.00177
	0.50	-0.00464	-0.01133	-0.00801	-0.00643	-0.01011	-0.00753	-0.00555	-0.00515
	1.00	-0.01300	-0.01542	-0.00911	-0.00991	-0.02123	-0.01729	-0.00945	-0.00474
1	0.25	-0.00402	-0.00437	-0.00341	-0.00335	-0.00492	-0.00399	-0.00210	-0.00161
	0.50	-0.00755	-0.00972	-0.00489	-0.00330	-0.00928	-0.00811	-0.00599	-0.00446
	1.00	-0.01724	-0.01688	-0.02827	-0.01221	-0.01743	-0.01458	-0.00865	-0.00651

Table 4.9: CLS: Bias($\hat{\delta}$) and bias($\hat{\tau}$) for $\lambda = 0$ and $\sigma = 1$, $\beta = 0.90$ based on 1,000 Monte Carlo runs.

Parameters		Bias($\hat{\delta}$)				Bias($\hat{\tau}$)			
δ	τ	$n = 25$	$n = 50$	n=100	n=200	$n = 25$	$n = 50$	n=100	n=200
0	0.25	0.00065	-0.00068	-0.00201	-0.00073	-0.00318	-0.00259	-0.00254	-0.00056
	0.50	-0.00314	-0.00228	0.00081	0.00086	-0.00639	-0.00308	-0.00250	-0.00185
	1.00	-0.00359	-0.00064	0.00090	0.00045	-0.01608	-0.00888	-0.00414	-0.00068
0.5	0.25	-0.00081	-0.00051	-0.00096	-0.00021	-0.00338	-0.00200	-0.00090	-0.00060
	0.50	0.00005	0.00093	-0.00570	-0.00056	-0.00715	-0.00360	-0.00325	-0.00209
	1.00	-0.00697	-0.00692	-0.00702	-0.00591	-0.01408	-0.00841	-0.00435	-0.00112
1	0.25	-0.00095	-0.00193	-0.00092	-0.00166	-0.00390	-0.00190	-0.00032	-0.00106
	0.50	-0.00122	-0.00089	-0.00163	-0.00207	-0.00539	-0.00308	-0.00106	-0.00087
	1.00	0.00308	-0.00088	-0.00815	-0.00600	-0.01173	-0.00674	-0.00784	-0.00333

Table 4.10: CLS: Bias($\hat{\delta}$) and bias($\hat{\tau}$) for $\lambda = 0$ and $\sigma = 1$, $\beta = 1$ based on 1,000 Monte Carlo runs.

Parameters		Bias($\hat{\delta}$)				Bias($\hat{\tau}$)			
δ	τ	$n = 25$	$n = 50$	n=100	n=200	$n = 25$	$n = 50$	n=100	n=200
0	0.25	-0.00122	-0.00009	0.00021	0.00004	-0.00296	-0.00194	-0.00139	-0.00047
	0.50	0.00195	0.00033	0.00293	-0.00234	-0.00706	-0.00481	-0.00290	-0.00213
	1.00	0.00372	-0.00046	-0.00451	-0.00216	-0.01556	-0.00745	-0.00065	-0.00567
0.5	0.25	0.00832	0.00402	0.00222	0.00104	-0.00344	-0.00215	-0.00023	-0.00106
	0.50	-0.00060	-0.00207	0.00418	0.00570	-0.00731	-0.00293	-0.00248	-0.00256
	1.00	-0.00119	-0.00206	0.00208	-0.00294	-0.01461	-0.00563	-0.00264	-0.00348
1	0.25	0.00134	0.00046	-0.00304	0.00097	-0.00317	-0.00181	-0.00121	-0.00051
	0.50	-0.00081	0.00197	0.00085	0.00121	-0.00777	-0.00442	-0.00278	-0.00114
	1.00	-0.00329	-0.00392	0.00711	-0.00724	-0.01535	-0.00714	-0.00737	-0.00051

Table 4.11: Comparative MSEs of $\hat{\delta}$ and $\hat{\tau}$ for different β using CLS method.

n	Parameter		MSE($\hat{\delta}$)			MSE($\hat{\tau}$)		
	δ	τ	$\beta = 0.5$	$\beta = 0.9$	$\beta = 1.0$	$\beta = 0.5$	$\beta = 0.9$	$\beta = 1.0$
25	0	0.25	0.00983	0.00839	0.00807	0.00209	0.00195	0.00192
		0.50	0.04028	0.03354	0.03318	0.00823	0.00765	0.00787
		1.00	0.15729	0.13590	0.13358	0.03243	0.03110	0.03093
50	0	0.25	0.00528	0.00424	0.00417	0.00109	0.00098	0.00096
		0.50	0.02203	0.01706	0.01623	0.00437	0.00396	0.00386
		1.00	0.08265	0.06784	0.06537	0.01770	0.01583	0.01561
100	0	0.25	0.00288	0.00220	0.00205	0.00058	0.00051	0.00048
		0.50	0.01130	0.00882	0.00777	0.00231	0.00207	0.00189
		1.00	0.04605	0.03209	0.02969	0.00923	0.00752	0.00766
200	0	0.25	0.00157	0.00114	0.00103	0.00033	0.00024	0.00025
		0.50	0.00572	0.00440	0.00434	0.00126	0.00102	0.00098
		1.00	0.02335	0.01853	0.01774	0.00510	0.00416	0.00376

Table 4.12: MLM : Probability coverage for $\hat{\delta}$ and $\hat{\tau}$ with $\lambda = 0$, $\sigma = 1$ based on 1,000 Monte Carlo runs.

Parameters		95% Coverage($\hat{\delta}$)				95% Coverage($\hat{\tau}$)			
δ	τ	$n = 25$	$n = 50$	n=100	n=200	n=25	n=50	n=100	n=200
0	0.25	0.935	0.946	0.959	0.947	0.908	0.938	0.947	0.948
	0.50	0.938	0.945	0.941	0.934	0.907	0.939	0.947	0.938
	1.00	0.938	0.951	0.957	0.945	0.906	0.916	0.939	0.948
0.5	0.25	0.930	0.947	0.944	0.965	0.897	0.940	0.961	0.941
	0.50	0.932	0.949	0.944	0.941	0.914	0.947	0.940	0.946
	1.00	0.923	0.951	0.938	0.947	0.913	0.921	0.941	0.959
1	0.25	0.934	0.952	0.950	0.956	0.917	0.936	0.925	0.930
	0.50	0.945	0.953	0.944	0.941	0.912	0.939	0.948	0.956
	1.00	0.946	0.929	0.963	0.948	0.928	0.922	0.941	0.944

Table 4.13: CLS: 95% Probability coverage for $\hat{\delta}$ with $\lambda = 0$ and $\sigma = 1$ based on R=200 Monte Carlo runs.

Weight	Parameter		n			
β	δ	τ	25	50	100	200
0.50	0	0.25	0.930	0.945	0.945	0.940
		0.50	0.975	0.945	0.960	0.960
		1.00	0.910	0.955	0.935	0.960
0.90	0	0.25	0.965	0.975	0.960	0.970
		0.50	0.955	0.945	0.980	0.960
		1.00	0.960	0.965	0.985	0.965
1.00	0	0.25	0.975	0.950	0.985	0.980
		0.50	0.970	0.960	0.975	0.980
		1.00	0.960	0.970	0.970	0.965

Table 4.14: CLS: 95% Probability coverage for $\hat{\tau}$ with $\lambda = 0$ and $\sigma = 1$ based on R=200 Monte Carlo runs.

Weight	Parameter		n			
β	δ	τ	25	50	100	200
0.50	0	0.25	0.925	0.940	0.910	0.945
		0.50	0.915	0.930	0.950	0.965
		1.00	0.895	0.945	0.955	0.955
0.90	0	0.25	0.935	0.955	0.965	0.975
		0.50	0.950	0.965	0.965	0.970
		1.00	0.930	0.920	0.955	0.955
1.00	0	0.25	0.925	0.940	0.925	0.970
		0.50	0.950	0.955	0.975	0.965
		1.00	0.930	0.955	0.940	0.960

Chapter 5

Comparison of the Methods

Methods of parameter estimation and Monte Carlo simulation steps are presented in the previous chapter. Here, we compare the methods discussed in the earlier chapters on the basis of MSE and bias of the estimators. We conducted simulation study for MLM, WLS and CLS methods as they are computationally less time consuming as compared to the EPM. We also compare MLM and CLS on the basis of average confidence lengths (ACL). For MLM, we use asymptotic confidence interval whereas bootstrap percentile (boot-p) confidence intervals are constructed for CLS method with the weights $\beta = 0.5, 0.9, 1$. Finally, comparative tables are presented at the end of this chapter.

5.1 Comparison Based on MSE

In order to compare the MLM, CLS and WLS methods, we tabulate MSE($\hat{\delta}$) in Table 5.1 and MSE($\hat{\tau}$) in Table 5.2. We observe that MLM gives the smallest MSE regardless of the sample size. For CLS method, MSE is the smallest when $\beta = 1$. Between CLS and WLS, CLS($\beta = 1$) gives smaller MSE irrespective of sample size. However, WLS gives smaller MSE when compared with CLS ($\beta = 0.5$) for sample size 50 or more. For large sample size ($n = 200$), MSEs are approximately equal for all the methods. For $\hat{\tau}$, similar conclusion can be drawn except that WLS always produces larger MSE than that of CLS. In general, MSE gets smaller as the sample size increases.

5.2 Comparison Based on Bias

From bias($\hat{\delta}$) in Table 5.3, we find that sometimes the parameter is underestimated and sometimes overestimated. Particularly for MLM, overestimation occurs more frequently than underestimation. For WLS, parameter is mostly underestimated and so for the CLS. The irregular underestimation (in MLM) and overestimation in WLS and CLS might be due to simulation.

On the other hand, bias($\hat{\tau}$) in Table 5.4 clearly shows that $\hat{\tau}$ underestimates τ irrespective of the method of estimation.

In general, bias becomes smaller for both δ and τ as the sample size increases.

5.3 Comparison Based on Boot-p Confidence Interval

Bootstrap percentile confidence intervals have been constructed in order to compare the different methods of estimation. We have observed in the previous chapter that MSE mainly depends on the scale parameter τ, when we fixed $\lambda = 0$ and $\sigma = 1$. That is why it is sufficient to construct bootstrap percentile confidence intervals for location parameter $\delta = 0$ with varying scale parameter τ. That is, we run the simulations for the parameter combinations $BL(\lambda = 0, \delta = 0, \sigma = 1, \tau = 0, 0.5, 1)$. Also, we limit our comparison between MLE and CLS methods only due to computational simplicity.

In the following we construct boot-p confidence intervals for δ and τ. This method was first proposed by Efron (1982). We shall illustrate the procedure for the location parameter δ. Confidence intervals for τ can be obtained in a similar fashion.

Step 1. Estimate the parameters of $BL(\lambda, \delta, \sigma, \tau)$from the available data using a suitable method.

Step 2. Generate a sample from the bivariate logistic distribution with parameters estimated in Step 1.

Step 3. Repeat Step 2 R times. This gives R estimates $\hat{\lambda^*}, \hat{\delta^*}, \hat{\sigma^*}, \hat{\tau^*}$ for each of the parameters $\lambda, \delta, \sigma, \tau$.

Step 4. Then a $100(1-\xi)\%$ confidence interval for δ is given by $(\hat{\delta}^*_{(R+1)(\frac{\xi}{2})}, \hat{\delta}^*_{(R+1)(1-\frac{\xi}{2})})$. That is, we sort the R $\hat{\delta}^*$'s in ascending order and take the $(R+1)(\frac{\xi}{2})^{\text{th}}$ and $(R+1)(1-\frac{\xi}{2})^{\text{th}}$ values. In other words, we take the $\left(\frac{\xi}{2}\right)^{\text{th}}$ and $\left(1-\frac{\xi}{2}\right)^{\text{th}}$ percentile point of the distribution of $\hat{\delta}^*$. Percentile bootstrap confidence intervals for τ is obtained in an analogous manner.

Following the above steps, boot-p confidence intervals for $\hat{\delta}$ and $\hat{\tau}$ have been obtained and the average lengths of the confidence intervals based on 200 Monte Carlo runs are presented in Table 5.5 and Table 5.6, respectively. In each Monte Carlo step, standard deviations of the estimates are obtained from 999 bootstrap replicates of the estimators.

For δ, we observe that for a particular combination of (δ, τ), MLM has a smaller confidence length as compared to CLS for any sample size. We see that confidence length for δ decreases with increasing sample size. We also observe that confidence length for δ increases with increasing τ. Similar conclusion can be drawn for confidence lengths for τ as given in Table 5.6.

Remark 2. Comparison of average confidence lengths (ACL) for δ and τ gives rise to the fact that ACL of τ is globally smaller than that of δ. For large sample, while there is noticeable difference in ACL(δ) between the methods, ACL(τ) do not vary that much.

Table 5.1: Comparison of MSE($\hat{\delta}$) between MLE, WLS, and CLS methods.

n	Parameter		MLM	WLS	CLS		
	δ	τ			$\beta = 0.5$	$\beta = 0.9$	$\beta = 1.0$
25	0	0.25	0.00699	0.01125	0.00983	0.00839	0.00807
		0.50	0.02789	0.04553	0.04028	0.03354	0.03318
		1.00	0.11273	0.17958	0.15729	0.13590	0.13358
50	0	0.25	0.00346	0.00535	0.00528	0.00424	0.00417
		0.50	0.01396	0.02007	0.02203	0.01706	0.01623
		1.00	0.05601	0.07870	0.08265	0.06784	0.06537
100	0	0.25	0.00174	0.00246	0.00288	0.00220	0.00205
		0.50	0.00698	0.01039	0.01130	0.00882	0.00777
		1.00	0.02841	0.03823	0.04605	0.03209	0.02969
200	0	0.25	0.00087	0.00122	0.00157	0.00114	0.00103
		0.50	0.00339	0.00499	0.00572	0.00440	0.00434
		1.00	0.01371	0.01886	0.02335	0.01853	0.01774

Table 5.2: Comparison of MSE($\hat{\tau}$) between MLM, WLS, and CLS methods.

n	Parameter		MLM	WLS	CLS		
	δ	τ			$\beta = 0.5$	$\beta = 0.9$	$\beta = 1.0$
25	0	0.25	0.00138	0.00413	0.00209	0.00195	0.00192
		0.50	0.00574	0.01647	0.00823	0.00765	0.00787
		1.00	0.02221	0.06543	0.03243	0.03110	0.03093
50	0	0.25	0.00068	0.00160	0.00109	0.00098	0.00096
		0.50	0.00272	0.00699	0.00437	0.00396	0.00386
		1.00	0.01103	0.02843	0.01770	0.01583	0.01561
100	0	0.25	0.00034	0.00069	0.00058	0.00051	0.00048
		0.50	0.00137	0.00332	0.00231	0.00207	0.00189
		1.00	0.00542	0.01323	0.00923	0.00752	0.00766
200	0	0.25	0.00017	0.00036	0.00033	0.00024	0.00025
		0.50	0.00071	0.00135	0.00126	0.00102	0.00098
		1.00	0.00268	0.00579	0.00510	0.00416	0.00376

Table 5.3: Comparison of Bias($\hat{\delta}$) between MLM, WLS, and CLS methods.

n	Parameter		MLM	WLS	CLS		
	δ	τ			$\beta = 0.5$	$\beta = 0.9$	$\beta = 1.0$
25	0	0.25	0.00049	-0.01147	-0.00395	0.00065	-0.00122
		0.50	0.00303	-0.01714	-0.00789	-0.00314	0.00195
		1.00	0.00557	-0.04639	-0.01250	-0.00359	0.00372
50	0	0.25	0.00189	-0.00208	-0.00467	-0.00068	-0.00009
		0.50	0.00156	-0.00836	-0.00935	-0.00228	0.00033
		1.00	0.00400	-0.00382	-0.01767	-0.00064	-0.00046
100	0	0.25	-0.00016	-0.00097	-0.00251	-0.00201	0.00021
		0.50	0.00158	0.00168	-0.01105	0.00081	0.00293
		1.00	0.00156	0.00007	-0.01212	0.00090	-0.00451
200	0	0.25	0.00002	-0.00103	-0.00522	-0.00073	0.00004
		0.50	-0.00013	-0.00075	-0.00528	0.00086	-0.00234
		1.00	0.00092	-0.00648	-0.00001	0.00045	-0.00216

Table 5.4: Comparison of Bias($\hat{\tau}$) between MLM, WLS, and CLS methods.

n	Parameter		MLM	WLS	CLS		
	δ	τ			$\beta = 0.5$	$\beta = 0.9$	$\beta = 1.0$
25	0	0.25	-0.00661	-0.00548	-0.00531	-0.00318	-0.00296
		0.50	-0.01393	-0.02055	-0.01015	-0.00639	-0.00706
		1.00	-0.02651	-0.02293	-0.01883	-0.01608	-0.01556
50	0	0.25	-0.00311	-0.00528	-0.00408	-0.00259	-0.00194
		0.50	-0.00680	-0.01171	-0.00636	-0.00308	-0.00481
		1.00	-0.01299	-0.02043	-0.01385	-0.00888	-0.00745
100	0	0.25	-0.00185	-0.00518	-0.00303	-0.00254	-0.00139
		0.50	-0.00326	-0.00842	-0.00577	-0.00250	-0.00290
		1.00	-0.00613	-0.02152	-0.01138	-0.00414	-0.00065
200	0	0.25	-0.00098	-0.00142	-0.00166	-0.00056	-0.00047
		0.50	-0.00182	-0.00379	-0.00468	-0.00185	-0.00213
		1.00	-0.00340	-0.00742	-0.01117	-0.00068	-0.00567

Table 5.5: Comparison of average confidence lengths for δ between MLM and CLS methods based on B=999, R=200.

n	Parameter		MLM	CLS		
	δ	τ		$\beta = 0.5$	$\beta = 0.9$	$\beta = 1.0$
25	0	0.25	0.3175	0.3843	0.3881	0.3815
		0.50	0.6474	0.7827	0.7686	0.7824
		1.00	1.2624	1.5408	1.5458	1.5577
50	0	0.25	0.2316	0.2825	0.2852	0.2861
		0.50	0.4576	0.5739	0.5786	0.5693
		1.00	0.9225	1.1421	1.1348	1.1543
100	0	0.25	0.1601	0.2071	0.2085	0.2085
		0.50	0.3252	0.4127	0.4188	0.4216
		1.00	0.6503	0.8324	0.8253	0.8283
200	0	0.25	0.1157	0.1492	0.1506	0.1503
		0.50	0.2310	0.3019	0.2993	0.3014
		1.00	0.4606	0.5995	0.5969	0.6046

Table 5.6: Comparison of average confidence lengths for τ between MLM and CLS methods based on B=999, R=200.

n	Parameter		MLM	CLS		
	δ	τ		$\beta=0.5$	$\beta=0.9$	$\beta=1.0$
25	0	0.25	0.1391	0.1732	0.1743	0.1723
		0.50	0.2827	0.3537	0.3475	0.3543
		1.00	0.5555	0.6961	0.6933	0.7030
50	0	0.25	0.1024	0.1271	0.1283	0.1292
		0.50	0.2013	0.2580	0.2600	0.2576
		1.00	0.4070	0.5135	0.5082	0.5196
100	0	0.25	0.0708	0.0936	0.0938	0.0945
		0.50	0.1438	0.1875	0.1895	0.1905
		1.00	0.2879	0.3758	0.3763	0.3760
200	0	0.25	0.0510	0.0681	0.0688	0.0688
		0.50	0.1023	0.1381	0.1366	0.1369
		1.00	0.2032	0.2734	0.2722	0.2757

Chapter 6

Illustrative Example

In this chapter we apply the methods discussed in the previous chapters for estimating parameters of bivariate logistic distribution using a real-life data. We use bootstrap resampling technique to calculate bias and MSE of the estimators of the parameters. We also present the percentile bootstrap confidence intervals for the parameters at the end of the chapter.

6.1 The UK Pig Production Data

We use the data obtained from UK pig production during the period 1967–78. The data are given by Andrews and Herzberg (1985) and used by Castillo et al. (1997). The data are presented in Table 6.1. 'Pig slaughter' is the number of clean pigs (in thousands), reared for meat as opposed to being culled from the breeding herd, which are slaughtered during one quarter of a year. It is the main measure of pig production. 'Herd size' is a measure of the actual size of the breeding herd.

Castillo et al. (1997) used CLS method to fit bivariate logistic distribution to this data. They have shown, in their paper, the hypothesis that the "sample (of the pig production) cannot be rejected from coming from the bivariate logistic population."

Table 6.1: UK pig production data (1967-'78)

Clean pig slaughter	Herd size	Clean pig slaughter	Herd size	Clean pig slaughter	Herd size
2.645	0.703	2.540	0.722	2.565	0.738
2.776	0.747	2.725	0.755	2.623	0.780
2.722	0.806	3.004	0.807	2.952	0.805
2.968	0.801	2.961	0.821	3.243	0.809
3.027	0.797	2.902	0.831	3.057	0.867
3.331	0.862	3.266	0.871	3.290	0.864
3.223	0.854	3.501	0.846	3.402	0.854
3.278	0.851	3.258	0.876	3.400	0.876
3.303	0.888	3.228	0.903	3.269	0.922
3.396	0.902	3.396	0.820	3.386	0.819
3.385	0.797	3.262	0.751	3.113	0.743
2.851	0.744	2.752	0.747	2.919	0.764
2.842	0.759	2.834	0.807	2.957	0.798
3.305	0.811	3.256	0.752	3.151	0.761
3.141	0.719	3.266	0.741	3.061	0.745
3.018	0.764	3.085	0.764	3.242	0.786

$n = 48$

6.2 Estimation of Parameters

Since bivariate logistic distribution reasonably fits the UK pig production data, we use MLM, CLS, WLS and EPM to estimate the parameters $\lambda, \delta, \sigma, \tau$. The results are presented in Table 6.2. The relevant **R** functions are given in Appendix Sections A.1, A.2, A.3, and A.4.

To estimate the parameters using MLM, WLS and EPM, we need to supply initial values for the parameters. We use the estimates obtained by CLS method (with $\beta = 0.5$) as the initial values. For CLS method, three different weights ($\beta = 0.5, 0.9, 1$) along with the optimal weight $\beta = 0.8468$ have been considered.

In the elemental percentile method, we randomly choose 4000 elemental subsets out of $\binom{48}{4}$= 194,580 possible subsets. Using these elemental subsets, we obtain 4000 elemental estimates for each of the parameters. Finally, we obtain the estimates using the functions MED and TM_α, with $\alpha = 10$ and 20.

In order to compare between the methods of estimation, we conduct simulation study based on bootstrap resampling with 999 replications. The results are presented and discussed in the following sections.

To obtain the MLEs of the parameters, we maximize the likelihood function using the optimization function `optim()` available in **R** (R Development Core Team 2004). The **R** functions used to obtain the MLEs are given in Appendix A.1. We got the estimates from the UK pig data as $\hat{\lambda} = 3.10, \hat{\delta} = 0.8, \hat{\sigma} = 0.157, \hat{\tau} = 0.03$.

Later, we applied Castillo's method to the data for weights $\beta = 0.50, 0.90, 1.0$. The optimum weight obtained for this data set is 0.8468. We have also estimated the parameters for the optimum weight and the results are tabulated in Table 6.2

From the UK pig production data we use bootstrap resampling to calculate the MSE, bias, and percentile bootstrap confidence intervals for the parameters of $BL(\lambda, \delta, \sigma, \tau)$. The bootstrap sampling can be done in two ways: the samples can be drawn directly from the data or they can be drawn parametrically from $F(x, y; \hat{\boldsymbol{\theta}})$ of the bivariate logistic distribution. Here, we followed the second approach to generate 999 bootstrap samples and the estimated the MSE and bias of the estimators are presented in Table 6.3 and Table 6.4.

Table 6.2: Estimated parameters of $BL(\lambda, \delta, \sigma, \tau)$ by four different methods.

Parameter	MLE	WLS	CLS				EPM		
			$\beta = 0.5$	$\beta = 0.9$	$\beta = 1.0$	$\beta^* = 0.847$	$\mathrm{TM}_{\alpha=10}$	$\mathrm{TM}_{\alpha=20}$	Median
$\hat{\lambda}$	3.1019	3.1121	3.0837	3.0838	3.0838	3.0838	3.0227	3.0246	3.0209
$\hat{\delta}$	0.8049	0.7935	0.8015	0.8023	0.8025	0.8022	0.7584	0.7781	0.7853
$\hat{\sigma}$	0.1575	0.1757	0.1344	0.1364	0.1365	0.1363	0.1628	0.1553	0.1440
$\hat{\tau}$	0.0318	0.0256	0.0291	0.0299	0.0301	0.0299	0.0862	0.0512	0.0361

β^* is the optimum weight obtained from the UK pig production data.

6.3 Discussion of Results

MSE of the estimators are presented in Table 6.3. We do not see much difference between the methods as far as MSE is concerned. However, MLM shows the smallest MSE as compared to the other three methods. CLS gives smaller MSE than that of WLS and EPM whereas EPM gives smaller MSE than that of WLS. This implies, WLS has the largest MSE while MLM has the smallest.

In Table 6.4 bias of the estimators are compared between different methods. From the negative bias of the estimators, we can say that all four methods underestimate the parameters.

Boot-p confidence intervals and length of the intervals are presented in Table 6.5 and Table 6.6, respectively. It is reasonable to say that no method is uniformly better than the others on the basis of confidence lengths. For example, MLM gives the smallest interval for δ and τ but produces largest interval for λ. However, after comparing all the methods, we can reasonably say that MLM and CLS ($\beta = 0.5$) performs well as compared to WLS and EPM.

Table 6.3: MSE of $\hat{\lambda}$, $\hat{\delta}$, $\hat{\sigma}$, and $\hat{\tau}$ based on 999 bootstrap replications for different methods.

MSE	MLM	WLS	CLS				EPM		
of			$\beta = 0.5$	$\beta = 0.9$	$\beta = 1.0$	$\beta^* = 0.8468$	$\mathrm{TM}_{\alpha=10}$	$\mathrm{TM}_{\alpha=20}$	MED
$\hat{\lambda}$	0.00030	0.00153	0.00005	0.00046	0.00005	0.00045	0.00056	0.00072	0.00087
$\hat{\delta}$	0.00000	0.00005	0.00006	0.00004	0.00000	0.00007	0.00228	0.00067	0.00030
$\hat{\sigma}$	0.00005	0.00012	0.00004	0.00001	0.00008	0.00006	0.00018	0.00001	0.00025
$\hat{\tau}$	0.00001	0.00001	0.00002	0.00000	0.00001	0.00000	0.01112	0.00089	0.00002

β^* is the optimum weight obtained from the UK pig production data.

Table 6.4: Bias of $\hat{\lambda}$, $\hat{\delta}$, $\hat{\sigma}$, and $\hat{\tau}$ based on 999 bootstrap replications for different methods.

Bias of	MLM	WLS	CLS				EPM		
			$\beta = 0.5$	$\beta = 0.9$	$\beta = 1.0$	$\beta^* = 0.8468$	$TM_{\alpha=10}$	$TM_{\alpha=20}$	MED
$\hat{\lambda}$	0.01743	0.03909	-0.00727	-0.02137	0.00680	-0.02131	-0.02356	-0.02680	-0.02958
$\hat{\delta}$	-0.00207	0.00682	-0.00748	-0.00627	-0.00056	-0.00837	-0.04772	-0.02582	-0.01724
$\hat{\sigma}$	0.00679	-0.01076	-0.00638	0.00241	-0.00910	0.00754	0.01325	0.00231	-0.01581
$\hat{\tau}$	-0.00368	-0.00301	-0.00388	0.00212	0.00349	0.00175	0.10543	0.02982	-0.00449

β^* is the optimum weight obtained from the UK pig production data.

Table 6.5: Bootstrap percentile confidence intervals for $\lambda, \delta, \sigma, \tau$ based on 999 bootstrap replications for different methods.

CI	MLM	WLS	CLS			
for			$\beta = 0.5$	$\beta = 0.9$	$\beta = 1.0$	$\beta^* = 0.8468$
$\hat{\lambda}$	(3.043, 3.201)	(3.0453, 3.2474)	(3.010, 3.143)	(2.982, 3.132)	(3.017, 3.160)	(2.978, 3.134)
$\hat{\delta}$	(0.785, 0.811)	(0.7935, 0.8195)	(0.777, 0.808)	(0.774, 0.813)	(0.779, 0.819)	(0.772, 0.811)
$\hat{\sigma}$	(0.129, 0.199)	(0.1139, 0.2314)	(0.092, 0.155)	(0.099, 0.169)	(0.091, 0.153)	(0.103, 0.175)
$\hat{\tau}$	(0.020, 0.032)	(0.0146, 0.0297)	(0.020, 0.033)	(0.024, 0.040)	(0.024, 0.043)	(0.023, 0.041)

β^* is the optimum weight obtained from the UK pig production data.

CI	EPM		
for	$\mathrm{TM}_{\alpha=10}$	$\mathrm{TM}_{\alpha=20}$	MED
$\hat{\lambda}$	(2.8084, 3.0102)	(2.8177, 3.0044)	(2.8238, 3.0021)
$\hat{\delta}$	(0.5505, 0.7507)	(0.5608, 0.7465)	(0.5758, 0.7451)
$\hat{\sigma}$	(0.1542, 0.3122)	(0.1371, 0.2446)	(0.1178, 0.2062)
$\hat{\tau}$	(0.1478, 0.2955)	(0.1332, 0.2354)	(0.1139, 0.1980)

Table 6.6: Confidence lengths for $\lambda, \delta, \sigma, \tau$ and $\hat{\tau}$ based on 999 bootstrap replications for different methods.

Confidence length for	MLM	WLS	CLS				EPM		
			$\beta = 0.5$	$\beta = 0.9$	$\beta = 1.0$	$\beta^* = 0.8468$	$\mathrm{TM}_{\alpha=10}$	$\mathrm{TM}_{\alpha=20}$	MED
$\hat{\lambda}$	0.15840	0.2021	0.13220	0.15010	0.14300	0.15590	0.2018	0.1867	0.17838
$\hat{\delta}$	0.02600	0.0260	0.03180	0.03950	0.04030	0.03840	0.2001	0.1857	0.16931
$\hat{\sigma}$	0.06930	0.1175	0.06340	0.06950	0.06250	0.07160	0.1580	0.1075	0.08832
$\hat{\tau}$	0.01190	0.0150	0.01320	0.01660	0.01820	0.01750	0.1477	0.1022	0.08410

β^* is the optimum weight obtained from the UK pig production data.

Chapter 7

Conclusion

Our objective was to compare between different estimation methods for estimating the parameters of bivariate logistic distribution. In this study, we compared maximum likelihood method (MLM), weighted least squares method (WLS), and Castillo's least squares method (CLS) on the basis of bias and mean squared errors.

There are four parameters in the bivariate logistic distribution namely, λ, δ as the location parameters, and σ, τ as the scale parameters. In this study, we limited our simulation for the parameters δ and τ keeping $\lambda = 0$ and $\sigma = 1$ as fixed. Thus, we simulated only for $\delta = 0, 0.5, 1$ and $\tau = 0.25, 0.5, 1$. We compute MSE and bias for $\hat{\delta}$ and $\hat{\tau}$. It has been found that MSE for both $\hat{\delta}$ and $\hat{\tau}$ are approximately equal for (δ, τ)= (0, 0.25), (0.5, 0.25), (1, 0.25). But when we allow τ to vary, MSE of $\hat{\delta}$ and $\hat{\tau}$ also varies. This implies, MSE depends on scale parameter when we fix λ and σ.

Comparative results based on MSE show that MLM produces smaller MSE as compared to WLS and CLS while CLS has smaller MSE than that of WLS. In terms of bias, we can conclude from the negative bias that the parameters are underestimated in most of the situations for all the methods.

Because MLM and CLS produce smaller MSE than that of WLS, we compared MLM and CLS on the basis of confidence lengths. Average lengths of asymptotic confidence intervals for MLM have been compared with the average percentile bootstrap confidence lengths for the CLS. For both δ and τ, MLM has smaller length than that of CLS. However, when sample becomes large ($n = 200$) the difference becomes negligible.

We applied all four methods of estimation to a real life-data. Bootstrap resampling was employed to compute bias, mean squared error and confidence lengths for the parameters using those data. Bootstrap resampling result shows that EPM has higher MSE than that of other methods. Although there is not much difference in MSE between MLM, CLS and WLS, it is reasonable to say that MLM and CLS produce similar result. There is no clear division between the methods in terms of bias of the estimators. Comparing the lengths of confidence intervals between methods (Table 6.6), we can say that MLM and CLS produce similar results.

In summary, MLM and CLS are found to be better than the other two methods of estimation and they can be used alternatively. CLS has some advantages over MLM, especially when the range of the variable depends on the parameters. In such cases, CLS can be used without any difficulty. It has been found that MLM takes less time but it is not straight forward computationally, for all values of the parameters. Also, for small sample sizes ($n < 25$) and especially when the scale parameters are very small, optimization of the loglikelihood function often fails. In those situations, even constrained optimization gives unsatisfactory results. On the other hand, CLS is computationally fast and can be applied without any difficulty for any sample size and acceptable parameter values.

Appendix A

R Functions: Simulation on UK Pig Production Data

A.1 Functions Related to MLM

```
mle.bivlog<-
function(ini, data, ...)
{
        xx<-data[,1]
        yy<-data[,2]
        n<-nrow(data)
    BLLL<-function(x,data)
    {
        lambda<-x[1]
        delta<-x[2]
        sigma<-x[3]
        tau<-x[4]
        xx<-data[,1]
        yy<-data[,2]
        n<-length(xx)
        log(2^n)- n*(mean(xx)-lambda)/sigma - n*(mean(yy)-delta)/tau -
        n*log(sigma) - n*log(tau) - 3*sum(log(1+exp(-(xx-lambda)/sigma)+
        exp(-(yy-delta)/tau)))
    }
    out<-optim(ini,BLLL,method="L-BFGS-B", lower=c(2,0,0.05,0.01),
    upper=c(6,2,1,.5),data=data,control=list(fnscale=-1),hessian=T)
    var.para=diag(solve(abs(out$hessian)))
    para=out$par
    if(is.na(sqrt(var.para[1]))){sd.lambda=1e-4}
    else sd.lambda=sqrt(var.para[1])
    if(is.na(sqrt(var.para[2]))){sd.delta=1e-4}
```

```
    else sd.delta=sqrt(var.para[2])
    if(is.na(sqrt(var.para[3]))){sd.sigma=1e-4}
    else sd.sigma=sqrt(var.para[3])
    if(is.na(sqrt(var.para[4]))){sd.tau=1e-4}
    else sd.tau=sqrt(var.para[4])
    list(para=para,var.para=var.para)
}

sim.mle<-
function(ini,nsize,R=1000,B=999, ...)
# ini= initial lamda, delta, sigma, tau
# nsize is the sample size
{
    options(digits=4)
    para.store<-matrix(0,nrow=R,ncol=4)
    asy.var<-matrix(0,nrow=R,ncol=4)
    p.val.store.lambda=NULL
    p.val.store.delta=NULL
    p.val.store.sigma=NULL
    p.val.store.tau=NULL
    bp.conf<-matrix(0,nrow=R,ncol=12)
    for (i in 1:R)
    {
        data1= rbivlog(nsize,ini)
        tmp=mle.bivlog(ini,data1)
        para.store[i,]=tmp$para
        asy.var[i,]<-tmp$var.para
        p.val.store.lambda[i]=tmp$p.val.lambda
        p.val.store.delta[i]=tmp$p.val.delta
        p.val.store.sigma[i]=tmp$p.val.sigma
        p.val.store.tau[i]=tmp$p.val.tau
        conf1<-boot.mle(c(para.store[i,]),nsize,B)
        bp.conf[i,]<-round(c(conf1$p.lam.ci[1],conf1$p.lam.ci[2],
        conf1$p.del.ci[1],conf1$p.del.ci[2],conf1$p.sig.ci[1],
        conf1$p.sig.ci[2],conf1$p.tau.ci[1],conf1$p.tau.ci[2],
        (conf1$p.lam.ci[2]-conf1$p.lam.ci[1]),(conf1$p.del.ci[2]-
        conf1$p.del.ci[1]),(conf1$p.sig.ci[2]-conf1$p.sig.ci[1]),
        (conf1$p.tau.ci[2]-conf1$p.tau.ci[1])),digits=4)
    }
    #para.store
    mse.lambda=sum((para.store[,1]-ini[1])^2)/R
    mse.delta=sum((para.store[,2]-ini[2])^2)/R
    mse.sigma=sum((para.store[,3]-ini[3])^2)/R
    mse.tau=sum((para.store[,4]-ini[4])^2)/R
    bias.lambda=mean(para.store[,1])-ini[1]
    bias.delta=mean(para.store[,2])-ini[2]
```

```
    bias.sigma=mean(para.store[,3])-ini[3]
    bias.tau=mean(para.store[,4])-ini[4]
    # pivotal quantities of the parameters
    p.lambda=(para.store[,1]-ini[1])/sqrt(asy.var[,1])
    p.delta=(para.store[,2]-ini[2])/sqrt(asy.var[,2])
    p.sigma=(para.store[,3]-ini[3])/sqrt(asy.var[,3])
    p.tau=(para.store[,4]-ini[4])/sqrt(asy.var[,4])
    # Asymptotic 95% Pr coverage
    aprx95.lambda=length(p.lambda[p.lambda>=-1.96 & p.lambda<=1.96])/R
    aprx95.delta=length(p.delta[p.delta>=-1.96 & p.delta<=1.96])/R
    aprx95.sigma=length(p.sigma[p.sigma>=-1.96 & p.sigma<=1.96])/R
    aprx95.tau=length(p.tau[p.tau>=-1.96 & p.tau<=1.96])/R
    # BootP average length of CI
    conf.len.lam<-mean(bp.conf[,9])
    conf.len.del<-mean(bp.conf[,10])
    conf.len.sig<-mean(bp.conf[,11])
    conf.len.tau<-mean(bp.conf[,12])
    result<-matrix(c(mse.lambda,mse.delta,mse.sigma,mse.tau,
    bias.lambda,bias.delta,bias.sigma,bias.tau,
    aprx95.lambda,aprx95.delta,aprx95.sigma,aprx95.tau,
    conf.len.lam, conf.len.del, conf.len.sig, conf.len.tau),
    ncol=4,byrow=T)
    rownames(result)<-c("mse", "bias", "aprx95", "aveconflen")
    colnames(result)<-c("lambda", "delta", "sigma", "tau")
    result
    # Comment the part below for MC simulation
    # list(result=round(result,digits=4), bp.conf=bp.conf)
 }

sim.mle.nb<-
function(ini,nsize,R=1000,B=999, ...)
{
    options(digits=4)
    para.store<-matrix(0,nrow=R,ncol=4)
    asy.var<-matrix(0,nrow=R,ncol=4)
    for (i in 1:R)
    {
        data1= rbivlog(nsize,ini)
        tmp=mle.bivlog(ini,data1)
        para.store[i,]=tmp$para
        asy.var[i,]<-tmp$var.para
    }
    #para.store
    mse.lambda=sum((para.store[,1]-ini[1])^2)/R
    mse.delta=sum((para.store[,2]-ini[2])^2)/R
    mse.sigma=sum((para.store[,3]-ini[3])^2)/R
    mse.tau=sum((para.store[,4]-ini[4])^2)/R
```

```
    bias.lambda=mean(para.store[,1])-ini[1]
    bias.delta=mean(para.store[,2])-ini[2]
    bias.sigma=mean(para.store[,3])-ini[3]
    bias.tau=mean(para.store[,4])-ini[4]
    # pivotal quantities of the parameters
    p.lambda=(para.store[,1]-ini[1])/sqrt(asy.var[,1])
    p.delta=(para.store[,2]-ini[2])/sqrt(asy.var[,2])
    p.sigma=(para.store[,3]-ini[3])/sqrt(asy.var[,3])
    p.tau=(para.store[,4]-ini[4])/sqrt(asy.var[,4])
    # Approximate 95% Pr coverage
    aprx95.lambda=length(p.lambda[p.lambda>=-1.96 & p.lambda<=1.96])/R
    aprx95.delta=length(p.delta[p.delta>=-1.96 & p.delta<=1.96])/R
    aprx95.sigma=length(p.sigma[p.sigma>=-1.96 & p.sigma<=1.96])/R
    aprx95.tau=length(p.tau[p.tau>=-1.96 & p.tau<=1.96])/R
    result<-matrix(c(mse.lambda,mse.delta,mse.sigma,mse.tau,
    bias.lambda,bias.delta,bias.sigma,bias.tau,
    aprx95.lambda,aprx95.delta,aprx95.sigma,aprx95.tau),
    ncol=4,byrow=T)
    rownames(result)<-c("mse", "bias", "aprx95")
    colnames(result)<-c("lambda", "delta", "sigma", "tau")
    result
    # Comment the part below for MC simulation
    # list(result=round(result,digits=5), asy.var=asy.var)
}
```

A.2 Functions Related to CLS

```
lsest<-
function(data,wt=0.5,...)
{
    x<-data[,1]
    y<-data[,2]
    r<-NULL
    s<-NULL
    n<-length(x)
    for (i in 1:length(x))
    {
        if(pxy(data,x[i],y[i])==px(y,y[i]))
        {
            r[i]<-log(1/px(x,x[i])-1)
        }
        else
        {
            r[i]<-wt*log(1/px(x,x[i])-1)+(1-wt)*
            log(1/pxy(data,x[i],y[i])-1/px(y,y[i]))
```

```
        }
        if(pxy(data,x[i],y[i])==px(x,x[i]))
        {
            s[i]<-log(1/px(y,y[i])-1)
        }   else
            {
                s[i]<-wt*log(1/px(y,y[i])-1)+(1-wt)*
                log(1/pxy(data,x[i],y[i])-1/px(x,x[i]))
            }
    }
    x.dnominator<-(sum(r))^2-n*sum(r^2)
    y.dnominator<-(sum(s))^2-n*sum(s^2)
    lam<-(sum(x*r)*sum(r)-sum(x)*sum(r^2))/x.dnominator
    sig<-(n*sum(x*r)-sum(x)*sum(r))/x.dnominator
    del<-(sum(y*s)*sum(s)-sum(y)*sum(s^2))/y.dnominator
    tau<-(n*sum(y*s)-sum(y)*sum(s))/y.dnominator
    pred.x<-lam-sig*r
    pred.y<-del-tau*s
    E<-sum((x-pred.x)^2+(y-pred.y)^2)
    list(para=c(lam,del, sig, tau),E=E)
}

optim.weight<-
function(data, lo=0,up=1, fig=F,...)
    {
    weight<- round(seq(lo,up,len=1000),digits=4)
    xx=NULL
    for (i in 1:length(weight))
    {
        x=lsest(data=data,weight[i])
        xx[i]=x$E
    }
    if(fig)
    {
        plot(weight,xx,type="l",xlab="Weight",
        ylab="Value of E in CLS method")
    }
    weight[which.min(xx)]
}

boot.lsest<-
function(ini,boot.ini,nsize,op.wt=0.5, B=999, ...)
{
    boot.para<-matrix(0,nrow=B,ncol=4)
    boot.var<-matrix(0,nrow=B,ncol=4)
    for (i in 1:B)
    {
```

```
        data.gen<-rbivlog(nsize,boot.ini)
        if (op.wt==0)
        {
            wt<-optim.weight(data1)
        }   else if (op.wt==0.5)
            {
                wt<-0.5
            }   else if (op.wt==0.9)
                {
                    wt<-0.9
                }   else if (op.wt==1)
                    {
                        wt<-1
                    }else wt<-op.wt

        boot.para[i,]<-lsest(data.gen, wt)$para
    }
    boot.sd<-apply(boot.para,2,sd)
    # Boot P conf
    p.lam.ci<-c(sort(boot.para[,1])[(B+1)*0.025],
    sort(boot.para[,1])[(B+1)*0.975])
    p.del.ci<-c(sort(boot.para[,2])[(B+1)*0.025],
    sort(boot.para[,2])[(B+1)*0.975])
    p.sig.ci<-c(sort(boot.para[,3])[(B+1)*0.025],
    sort(boot.para[,3])[(B+1)*0.975])
    p.tau.ci<-c(sort(boot.para[,4])[(B+1)*0.025],
    sort(boot.para[,4])[(B+1)*0.975])
    sd.lambda=boot.sd[1]
    sd.delta=boot.sd[2]
    sd.sigma=boot.sd[3]
    sd.tau=boot.sd[4]
    if(is.na(sd.lambda)){sd.lambda=1e-4}
    if(is.na(sd.delta)){sd.delta=1e-4}
    if(is.na(sd.sigma)){sd.sigma=1e-4}
    if(is.na(sd.tau)){sd.tau=1e-4}
   list(p.lam.ci=p.lam.ci, p.del.ci=p.del.ci,
   p.sig.ci=p.sig.ci,p.tau.ci=p.tau.ci,
   sd.lambda=sd.lambda,sd.delta=sd.delta,
   sd.sigma=sd.sigma, sd.tau=sd.tau)
}

sim.lsest<-
function(ini, nsize, op.wt=0.5, R=1000, B=999,...)
{   options(digits=5)
    para.store=matrix(0,nrow=R,ncol=4)
    sd.store<-matrix(0,nrow=R,ncol=4)
    bp.conf<-matrix(0,nrow=R,ncol=12)
```

```
for (i in 1:R)
{
    if (op.wt==0)
    {
        wtt<-optim.weight(data1)
    }   else if (op.wt==0.5)
        {
            wtt<-0.5
        }   else if (op.wt==0.9)
            {
                wtt<-0.9
            }   else if (op.wt==1)
                {
                    wtt<-1
                } else wtt<-op.wt
    data1= rbivlog(nsize,ini)
    tmp=lsest(data1,wt=wtt)
    para.store[i,]=tmp$para
    conf1<-boot.lsest(ini,boot.ini=c(para.store[i,]),nsize,B=B)
    sd.store[i,]<-round(c(conf1$sd.lambda,conf1$sd.delta,
    conf1$sd.sigma,conf1$sd.tau),digits=4)
    bp.conf[i,]<-round(c(conf1$p.lam.ci[1],conf1$p.lam.ci[2],
    conf1$p.del.ci[1],conf1$p.del.ci[2],conf1$p.sig.ci[1],
    conf1$p.sig.ci[2],conf1$p.tau.ci[1],conf1$p.tau.ci[2],
    (conf1$p.lam.ci[2]-conf1$p.lam.ci[1]),
    (conf1$p.del.ci[2]-conf1$p.del.ci[1]),
    (conf1$p.sig.ci[2]-conf1$p.sig.ci[1]),
    (conf1$p.tau.ci[2]-conf1$p.tau.ci[1])),digits=4)
    # p95.lam's are boot percentile conf.
}
# pivotal quantities of the parameters
p.lambda=(para.store[,1]-ini[1])/sd.store[,1]
p.delta=(para.store[,2]-ini[2])/sd.store[,2]
p.sigma=(para.store[,3]-ini[3])/sd.store[,3]
p.tau=(para.store[,4]-ini[4])/sd.store[,4]
# Approximate 95% Pr coverage
aprx95.lambda=length(p.lambda[p.lambda>=-1.96 & p.lambda<=1.96])/R
aprx95.delta=length(p.delta[p.delta>=-1.96 & p.delta<=1.96])/R
aprx95.sigma=length(p.sigma[p.sigma>=-1.96 & p.sigma<=1.96])/R
aprx95.tau=length(p.tau[p.tau>=-1.96 & p.tau<=1.96])/R
#para.store
mse.lambda=sum((para.store[,1]-ini[1])^2)/R
mse.delta=sum((para.store[,2]-ini[2])^2)/R
mse.sigma=sum((para.store[,3]-ini[3])^2)/R
mse.tau=sum((para.store[,4]-ini[4])^2)/R
bias.lambda=mean(para.store[,1])-ini[1]
```

```
    bias.delta=mean(para.store[,2])-ini[2]
    bias.sigma=mean(para.store[,3])-ini[3]
    bias.tau=mean(para.store[,4])-ini[4]
    # BootP average length of CI
    conf.len.lam<-mean(bp.conf[,9])
    conf.len.del<-mean(bp.conf[,10])
    conf.len.sig<-mean(bp.conf[,11])
    conf.len.tau<-mean(bp.conf[,12])
    result<-matrix(c(mse.lambda,mse.delta,mse.sigma,mse.tau,
    bias.lambda,bias.delta,bias.sigma,bias.tau,
    aprx95.lambda,aprx95.delta,aprx95.sigma,aprx95.tau,
    conf.len.lam, conf.len.del, conf.len.sig, conf.len.tau,),
    ncol=4,byrow=T)
    rownames(result)<-c("mse", "bias","aprxp", "aveconflen")
    colnames(result)<-c("lambda", "delta", "sigma", "tau")
    #result
    # Comment the part below for MC simulation
    list(result=round(result,digits=5), bp.conf=bp.conf)
} #end
```

A.3 Functions Related to WLS

```
wls<-
function(ini,data,...)
{
    E<-function(ini,data,...)
    {
        lambda<-ini[1]
        delta<-ini[2]
        sigma<-ini[3]
        tau<-ini[4]
        n<-nrow(data)
        pxiyi=NULL
        for (i in 1:n)
        {
            data$pxiyi[i]<-pxy(data[,1:2],data[,1][i],data[,2][i])
        }
       sum((n/(data[,3]*(1-data[,3])))*(1/(1+ exp(-(data[,1]-
       lambda)/sigma)+  exp(-(data[,2]-delta)/tau))-data[,3])^2)
    }
    out<-optim(ini,E,data=data,control=list(maxit=2000))$par
list(para=out)
}

boot.wls<-
function(ini,boot.ini,nsize,B=999, ...)
```

```
{
    boot.para<-matrix(0,nrow=B,ncol=4)
    for (i in 1:B)
    {
        data.gen<-rbivlog(nsize,boot.ini)
        boot.para[i,]<-wls(boot.ini, data.gen)$para
    }
    boot.sd<-apply(boot.para,2,sd)
    # Boot P conf
    p.lam.ci<-c(sort(boot.para[,1])[(B+1)*0.025],
    sort(boot.para[,1])[(B+1)*0.975])
    p.del.ci<-c(sort(boot.para[,2])[(B+1)*0.025],
    sort(boot.para[,2])[(B+1)*0.975])
    p.sig.ci<-c(sort(boot.para[,3])[(B+1)*0.025],
    sort(boot.para[,3])[(B+1)*0.975])
    p.tau.ci<-c(sort(boot.para[,4])[(B+1)*0.025],
    sort(boot.para[,4])[(B+1)*0.975])
    sd.lambda=boot.sd[1]
    sd.delta=boot.sd[2]
    sd.sigma=boot.sd[3]
    sd.tau=boot.sd[4]
    if(is.na(sd.lambda)){sd.lambda=1e-4}
    if(is.na(sd.delta)){sd.delta=1e-4}
    if(is.na(sd.sigma)){sd.sigma=1e-4}
    if(is.na(sd.tau)){sd.tau=1e-4}
    list(p.lam.ci=p.lam.ci, p.del.ci=p.del.ci,
    p.sig.ci=p.sig.ci,p.tau.ci=p.tau.ci,
    sd.lambda=sd.lambda,sd.delta=sd.delta,
    sd.sigma=sd.sigma, sd.tau=sd.tau)
}

sim.wls<-
function(ini,nsize,R=1000, B=999,...)
{
    options(digits=5)
    para.store=matrix(0,nrow=R,ncol=4)
    sd.store<-matrix(0,nrow=R,ncol=4)
    bp.conf<-matrix(0,nrow=R,ncol=12)
    for (i in 1:R)
    {
        data1= rbivlog(nsize,ini)
        tmp=wls(ini,data1)
        para.store[i,]=tmp$para
        conf1<-boot.wls(ini,boot.ini=c(para.store[i,]),nsize,B=B)
        sd.store[i,]<-round(c(conf1$sd.lambda,conf1$sd.delta,
        conf1$sd.sigma,conf1$sd.tau),digits=4)
        bp.conf[i,]<-round(c(conf1$p.lam.ci[1],conf1$p.lam.ci[2],
```

```
        conf1$p.del.ci[1],conf1$p.del.ci[2],conf1$p.sig.ci[1],
        conf1$p.sig.ci[2],conf1$p.tau.ci[1],conf1$p.tau.ci[2],
        (conf1$p.lam.ci[2]-conf1$p.lam.ci[1]),
        (conf1$p.del.ci[2]-conf1$p.del.ci[1]),
        (conf1$p.sig.ci[2]-conf1$p.sig.ci[1]),
        (conf1$p.tau.ci[2]-conf1$p.tau.ci[1])),digits=4)
    }
    #para.store
    mse.lambda=sum((para.store[,1]-ini[1])^2)/R
    mse.delta=sum((para.store[,2]-ini[2])^2)/R
    mse.sigma=sum((para.store[,3]-ini[3])^2)/R
    mse.tau=sum((para.store[,4]-ini[4])^2)/R
    bias.lambda=mean(para.store[,1])-ini[1]
    bias.delta=mean(para.store[,2])-ini[2]
    bias.sigma=mean(para.store[,3])-ini[3]
    bias.tau=mean(para.store[,4])-ini[4]
    # pivotal quantities of the parameters
    p.lambda=(para.store[,1]-ini[1])/sd.store[,1]
    p.delta=(para.store[,2]-ini[2])/sd.store[,2]
    p.sigma=(para.store[,3]-ini[3])/sd.store[,3]
    p.tau=(para.store[,4]-ini[4])/sd.store[,4]
    # Approximate 95% Pr coverage
    aprx95.lambda=length(p.lambda[p.lambda>=-1.96 & p.lambda<=1.96])/R
    aprx95.delta=length(p.delta[p.delta>=-1.96 & p.delta<=1.96])/R
    aprx95.sigma=length(p.sigma[p.sigma>=-1.96 & p.sigma<=1.96])/R
    aprx95.tau=length(p.tau[p.tau>=-1.96 & p.tau<=1.96])/R
    # BootP average length of CI
    conf.len.lam<-mean(bp.conf[,9])
    conf.len.del<-mean(bp.conf[,10])
    conf.len.sig<-mean(bp.conf[,11])
    conf.len.tau<-mean(bp.conf[,12])
    result<-matrix(c(mse.lambda,mse.delta,mse.sigma,mse.tau,
    bias.lambda,bias.delta,bias.sigma,bias.tau,
    aprx95.lambda,aprx95.delta,aprx95.sigma,aprx95.tau,
    conf.len.lam, conf.len.del, conf.len.sig, conf.len.tau,),
    ncol=4,byrow=T)
    rownames(result)<-c("mse", "bias","aprxp", "aveconflen")
    colnames(result)<-c("lambda", "delta", "sigma", "tau")
    #result
    # Comment the part below for MC simulation
    list(result=round(result,digits=5), bp.conf=bp.conf)
}
```

A.4 Functions Related to EPM

```
epm.bivlog<-
function(ini, data, eset,...)
{
    data.origin=data
    N=nrow(eset)
    out=matrix(0,nrow=N,ncol=4)
    for(i in 1:N)
    {
        data=rbind(data[eset[i,1],],data[eset[i,2],],
        data[eset[i,3],],data[eset[i,4],])
        data$k=1/biv.cdf(data)-1
        tmp.out=nlest(ini,data)
        out[i,]=tmp.out$para
        data=data.origin
    }
        para.mean20=c(trim.mean(out[,1],trim=20),
        trim.mean(out[,2],trim=20),trim.mean(out[,3],trim=20),
        trim.mean(out[,4],trim=20))
        para.mean10=c(trim.mean(out[,1],trim=10),
        trim.mean(out[,2],trim=10),trim.mean(out[,3],trim=10),
        trim.mean(out[,4],trim=10))
        para.median=c(median(out[,1]),median(out[,2]),
        median(out[,3]),median(out[,4]))
        list(para.mean10=para.mean10,
        para.mean20=para.mean20,para.median=para.median)
}
```

A.5 Miscellaneous R Functions

```
biv.cdf<-function(data, ...)
{
    x=data[,1]
    y=data[,2]
    k=NULL
    for (i in 1: length(x))
    {
        k[i]= pxy(data,x[i],y[i])
    }
    k
}

item<-
function(nsize,r=4,...)
# required package "gtools"; library(gtools)
```

```
# nsize=nrow(data)
# r = # of parameters, 4 in our case.
{
    options(expressions=1e5)
    comb=combinations(nsize,r)
    if(choose(nsize,r)>4000)
    {
        comb=comb[c(sample(choose(nsize,4),4000)),]
    } else
        {
            comb<-combinations(nsize,4)
        }
    comb
}

nlest<-
function(ini, data, method="Nelder-Mead",...)
{
    E<-function(x,data)
    {
        lambda<-x[1]
        delta<-x[2]
        sigma<-x[3]
        tau<-x[4]
        k<-data[,3]
        sum((exp(-(data[,1]-lambda)/sigma) +
        exp(-(data[,2]-delta)/tau)-k)^2)
    }
    out=optim(ini,E, method=method, data=data,control=list(maxit=4000))
    para=out$par
    list(para=para)
}

optim.weight<-function(data, lo=0,up=1, fig=F,...)
{
    weight<- round(seq(lo,up,len=1000),digits=4)
    xx=NULL
    for (i in 1:length(weight))
    {
        x=lsest(data=data,weight[i])
        xx[i]=x$E
    }
    if(fig)
    {
        plot(weight,xx,type="l",xlab="Weight",
        ylab="Value of E in CLS method")
    }
```

```
    weight[which.min(xx)]
}

 px<-function(data,val,...)
    {
        (length(data[data<=val])-0.5)/length(data)
    }

 pxy<-function(data,valx,valy,...)
    {
        x=data[,1]
        y=data[,2]
        (nrow(matrix((as.matrix(data)[x<=valx & y<=valy]),
        ncol=2))-0.5)/length(x)
    }

# Generates bivariate logistic random numbers
# for a given set of parameter values
rbivlog<-function(n,par)
{
    lambda<-par[1]
    delta<-par[2]
    sigma<-par[3]
    tau<-par[4]
    random<-matrix(c(runif(n),runif(n)),ncol=2)
    u<-random[,1]
    v<-random[,2]
    x<-NULL
    y<-NULL
    for(i in 1:n)   {
        x[i]<-lambda-sigma*log(1/u[i]-1)
        y[i]<-delta-tau*log(1/(u[i]*sqrt(v[i]))-1/u[i])
    }
    data.frame(x,y)
}

trim.mean<-
function(x, trim=20,...)
# trimming is top 10% and bottom 10%
{
    n=length(x)
    x=sort(x)
    if(n<20)
    {
        alpha=ceiling(trim*n/200)
        if(alpha<1){alpha=1}
```

```
        mean(x[x>x[alpha]& x<=x[n-alpha]])
    }
    else
    {
        alpha=trim*n/200
        if(alpha<1){alpha=1}
        mean(x[x>x[alpha] & x<=x[n-alpha]])
    }
}
```

Bibliography

Andrews, D. F. and Herzberg, A. M. (1985), *Data: A collection of problems from many fields for the student and research worker*, Springer, New York.

Arnold, B. C. (1992), *In Handbook of the Logistic Distribution*, Marcel Dekker, New York, chapter Multivariate Logistic Distribution, pp. 237–261.

Arnold, B. N., Balakrishnan, N. and Nagaraja, H. N. (1992), *A First Course in Order Statistics*, John Wiley & Sons, New York.

Balakrishnan, N. (1992), *In Handbook of the logistic distribution*, Marcel Dekker Inc., chapter Introduction and historical remarks, pp. 1–16.

Bates, D. and Watts, D. (1988), *Nonlinear Regression Analysis and its Applications*, John Wiley & Sons, New York.

Castillo, E. and Hadi, A. S. (1995), 'A method of estimating parameters and quantiles of distributions of continuous random variable', *Computational Statistics & Data Analysis* **20**, 421–439.

Castillo, E., Hadi, A. S., Balakrishnan, N. and Sarabia, J. M. (2005), *Extreme Value and Related Models with Applications in Engineering and Science*, John Wiley & Sons, New Jersey.

Castillo, E., Sarabia, J. M. and Hadi, A. S. (1997), 'Fitting continuous bivariate distributions to data', *The Statistician* **46**(3), 355–369.

Diaconis, P. and Efron, B. (1974), 'Computer intensive methods in statistics', *Scientific American* **248**, 116–130.

Dyke, G. V. and Patterson, H. D. (1952), 'Analysis of factorial arrangements when the data are proportions', *Biometrics* **8**, 1–12.

Efron, B. (1979), 'Bootstrap methods: Another look at the jackknife', *The Annals of Statistics* **7**, 1–26.

Efron, B. (1982), 'The jackknife, the boostrap and other resampling plans', *In:CBMS-NSF Regional Conference series in Applied Mathematics* **38**.

Fisk, P. R. (1961), 'The graduation of income distribution', *Econometrica* **29**, 171–185.

Gomes, M. I. and Oliveira, O. (2001), 'The bootstrap methodology in statistics of extreme-choice of the optimal sample fraction', *Extremes* **4**(4), 331–358.

Grizzle, J. E. (1961), 'A new method of testing hypotheses and estimating parameters for the logistic model', *Biometrics* **17**, 372–385.

Gumbel, E. J. (1961), 'Bivariate logistic distributions', *Journal of the American Statistical Association* **56**, 335–349.

Maple (2003), *Maplesoft, a division of Waterloo Maple Inc. 1981-2003*, Waterloo Maple Inc.

Oliver, F. R. (1964), 'Methods of estimating the logistic growth function', *Applied Statistics* **13**, 57–66.

Pearl, R. (1940), *Medical Biometry and Statistics*, Saunders, Philadelphia.

Pearl, R. and Reed, L. J. (1920), 'On the rate of growth of the population of the united states since 1790 and its mathematical representation', *Proc. Nat. Acad. Sci.* **6**, 275–288.

Plackett, R. L. (1959), 'The analysis of life test data', *Technometrics* **1**, 9–19.

R Development Core Team (2004), *R: A language and environment for statistical computing*, R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0.
*http://www.R-project.org

Reed, L. J. and Berkson, J. (1929), 'The application of the lgistic function to experimental data', *J. Phys. Chem.* **33**, 760–770.

Schultz, H. (1930), 'The standard error of a forecast from a curve.', *Journal of the American Statistical Association* **25**, 139–185.

Verhulst, P. J. (1838), 'Notice sur la lois que la population suit dans sons accroissement', *Corr. Math. Phys.* **10**, 113–121.

Verhulst, P. J. (1845), 'Recherches mathematiques sur la loi d'accroissement de la population', *Acad. Bruxelles* **18**, 1–38.

Wilson, E. B. and Worcester, J. (1943), 'The determination of l.d.50 and its sampling error in bio-assay', *Proc. National Acad. Sci.* **29**, 79–85.