UNIVERSITY ***of the***
WESTERN CAPE

Estimation of Pareto Distribution Functions from Samples Contaminated by Measurement Errors

Lwando Orbet Kondlo

A thesis submitted in partial fulfillment of the requirements for the degree of Magister Scientiae in Statistics in the Faculty of Natural Sciences at the University of the Western Cape.

Supervisor : Professor Chris Koen

March 1, 2010

Keywords

Deconvolution
Distribution functions
Error-Contaminated samples
Errors-in-variables
Jackknife
Maximum likelihood method
Measurement errors
Nonparametric estimation
Pareto distribution

ABSTRACT

MSc Statistics Thesis, Department of Statistics, University of the Western Cape.

Estimation of population distributions, from samples which are contaminated by measurement errors, is a common problem. This study considers the problem of estimating the population distribution of independent random variables X_j, from error-contaminated samples Y_j $(j = 1, \ldots, n)$ such that $Y_j = X_j + \epsilon_j$, where ϵ is the measurement error, which is assumed independent of X. The measurement error ϵ is also assumed to be normally distributed. Since the observed distribution function is a convolution of the error distribution with the true underlying distribution, estimation of the latter is often referred to as a *deconvolution problem*. A thorough study of the relevant deconvolution literature in statistics is reported.

We also deal with the specific case when X is assumed to follow a truncated Pareto form. If observations are subject to Gaussian errors, then the observed Y is distributed as the convolution of the finite-support Pareto and Gaussian error distributions. The convolved probability density function (PDF) and cumulative distribution function (CDF) of the finite-support Pareto and Gaussian distributions are derived.

The intention is to draw more specific connections between certain deconvolution methods and also to demonstrate the application of the statistical theory of estimation in the presence of measurement error.

A parametric methodology for deconvolution when the underlying distribution is of the Pareto form is developed.

Maximum likelihood estimation (MLE) of the parameters of the convolved distributions is considered. Standard errors of the estimated parameters are calculated from the inverse Fisher's information matrix and a jackknife method. Probability-probability (P-P) plots and Kolmogorov-Smirnov (K-S) goodness-of-fit tests are used to evaluate the fit of the posited distribution. A bootstrapping method is used to calculate the critical values of the K-S test statistic, which are not available.

Simulated data are used to validate the methodology. A real−life application of the methodology is illustrated by fitting convolved distributions to astronomical data

March 1, 2010

Declaration

I declare that *Estimation of Pareto Distribution Functions from Samples Contaminated by Measurement Errors* is my work, that it has not been submitted before for any degree or examination in any other university, and that all sources I have used or quoted have been indicated and acknowledged by complete references.

Lwando Orbet Kondlo March 1, 2010

Signed .....

Table of Contents

List of Figures

List of Tables

List of Acronyms

AIC	Akaike information criterion
BIC	Bayes information criterion
CDF	Cumulative distribution function
CF	Characteristic function
EDF	Empirical distribution function
GMC	Gaint molecular cloud
K-S	Kolmogorov-Smirnov
LMC	Large Magellanic Cloud
PDF	Probability density function
MLE	Maximum likelihood estimation
FSPD	Finite−support Pareto distribution

Acknowledgments

I would like to sincerely thank my MSc supervisor Professor Chris Koen, thank you for your support, time, encouragement and enthusiasm for my work, the right advice at the right time and everything else. It was a pleasure working with you.
To my family and friends: thank you for your support and encouragement.
Funding for the research: the Department of Statistics (University of the Western Cape), Square Kilometer Array South Africa (especially to Kim, Anna and Daphne), Eskom, and the National Research Foundation.

Dedication

This work is dedicated to my family: To the soul of my father, Wiseman Lumkile Kondlo, who has valued and loved education and invested generously in me and my brothers and sisters; to my mother, Sylvia Kondlo; to my son and daughter Imihlali-Emihle and Uthando-Lunje; and their mother Athi-Anathi Badi.

Chapter 1

General Introduction and Objectives

1.1 Introduction

A statistical problem which is common in many areas of research such as econometrics, astronomy, public health or biostatistics is the need to derive population properties from samples which are contaminated by measurement errors. In the statistical literature, a sample is defined as a subset of a population. Typically, "population" is the conceptual totality of objects under consideration. Samples are selected from the population and statistics are calculated from the samples so that one can make inferences or extrapolations from the sample to the population. In this study, the sample values are measured, but subject to measurement errors. The observed contaminated sample values are then used to estimate the underlying population (i.e., general) properties. An example includes density or distribution estimation of a variable X given data Y. Estimation of these values is known to be problematic if the variable X is measured with errors. Previous work with this problem generally has used the assumption that the measurement errors have a Gaussian distribution (Eltinge, 1999). Problems of this nature are commonly called *measurement error problems*, and statistical models and methods for analysing such sample data are called *measurement error models* (Stefanski, 2000).

The first part of this study reviews the literature on nonparametric density/distribution estimation based on contaminated samples. Some relevant references are Mendel-

sohn and Rice (1982), Stefanski and Carroll (1990), Masry and Rice (1992), Efromovich (1997), Carroll and Hall (2004), Chen *et al.* (2003), Devroye (1989) amongst others. The second part is concerned with parametric estimation methods. The approach is to assume a certain distribution family for the observed contaminated sample values and estimate the parameters of the specific family member. Chapter 4 contains an extensive discussion of the parametric method. In the statistical literature, researchers often focus on the theoretical underpinnings whereas here, the focus is only the methodology and theory of reasonably immediate practical relevance.

1.2 Measurement error problems

The model for a variable X measured with error ϵ is

$$Y = X + \epsilon \tag{1.1}$$

where

(i) X is the underlying variable of interest

(ii) Y is the observed value

(iii) ϵ is the difference between the underlying variable X and the observation Y; ϵ is also called measurement error and assumed to be independent of X.

In many areas of application, statistically meaningful models are defined in terms of variables X that for some reason are not directly observable. In such situations, it is common for substitute variables Y to be observable instead. The substitution of the variable Y for X complicates the statistical analysis of the observed data when the purpose of analysis is inference about a model defined in terms of X (Stefanski, 2000). Problems of this nature are commonly called measurement error problems. Some examples include, measurements of systolic blood pressure (Stefanski, 2000); environmental risk factors, case-control studies of disease and serum hormone levels, food intake records, 24-hour recalls and biomarkers (Carroll, 1997). Other more detailed examples of measurement error problems have been given by Carroll *et al.* (1995) and Meister (2009).

Generally, there are two statistical methods for estimating the underlying population properties. One is the parametric approach and the other is a nonparametric one.

1.2.1 The nonparametric approach

The nonparametric method does not specify a particular family of distributions. There is a particularly extensive literature on solving the problem posed by model (1.1), where one wishes to estimate the distribution function of X.

Suppose the underlying variable X has the probability density function (PDF) $f_X(\cdot)$, and ϵ has the PDF $f_\epsilon(\cdot)$. The random variable Y in (1.1) has the PDF $f_Y(\cdot)$ given by the convolution integral

$$f_Y(y) = \int_{-\infty}^{\infty} f_\epsilon(y-x) f_X(x)\, dx \tag{1.2}$$

(Mood *et al.*, 1974). The corresponding cumulative distribution function (CDF) of Y can be expressed as

$$F_Y(y) = \int_{-\infty}^{\infty} f_\epsilon(y-x) F_X(x)\, dx \tag{1.3}$$

(Billingsley, 1994 and Gaffey, 1959). The form of $f_\epsilon(\cdot)$ is usually assumed to be known.

The problem is to estimate the density $f_X(\cdot)$ and/or the CDF $F_X(\cdot)$ given observations $\{y_1, y_2, \ldots, y_n\}$ and the density $f_\epsilon(\cdot)$. The estimation of $f_X(\cdot)$ in (1.2) and $F_X(\cdot)$ in (1.3) is referred to as the *deconvolution problem*. Measurement error problems are sometimes called deconvolution problems.

Density deconvolution has been addressed by several authors, including amongst others Carroll and Hall (1988), Fan (1992), Liu and Taylor (1989), Stefanski and Carroll (1990), Carroll and Hall (2004) using kernel methods. Turlach and Hazelton (2008) proposed an approach to density deconvolution, based on the use of weighted kernel estimators. Masry and Rice (1992) address the problem using estimates of the derivatives of the convoluted density. Cordy and Thomas (1997), Mendelsohn and Rice (1982), Chen *et al.* (2003) treated the settings where distribution function

can be expressed as a finite sum of known distribution functions. Gaffey (1959) and Stefanski and Bay (1996) considered estimating the distribution F_X in the presence of normally distributed measurement error.

The methodologies for several nonparametric deconvolution methods used to estimate the density $f_X(\cdot)$ and the distribution $F_X(\cdot)$ are briefly discussed in Chapter 3.

1.2.2 The parametric approach

The parametric approach is to assume a certain distribution family for the observed contaminated sample values and estimate the parameters of the specific family member. Specifically, the general forms of the densities $f_X(\cdot)$ and $f_\epsilon(\cdot)$ in (1.2) are (assumed) known, but one wants to estimate some parameters in one or both densities. In Chapter 4, we deal with a specific case where X has a power-law (i.e., Pareto) form. The parameters of the distribution of X, given data y and assuming Gaussian errors ϵ are estimated. The standard method of maximum likelihood for parameter estimation is indicated, and developed here.

1.3 Assumptions

In order to construct any estimation method to analyse the effect of measurement error, one needs to make some assumptions about the process which generates the differences between the underlying variable of interest X and the observed value Y. The two critical common assumptions that underlie the measurement error model are as follows:

(i) The independence between the random variables X and ϵ. This describes that the targeted random variable X does not have any effect or influence on the measurement error ϵ.

(ii) The measurement error ϵ is known to have a normal distribution with zero mean and variance σ^2. The normal error variance σ^2, which may be unknown, is assumed constant across the observations.

These assumptions have been documented in all studies related to deconvolution problems. Therefore, except where otherwise stated, these assumptions are made

throughout this research project.

1.4 Motivation of the Study

The work presented in this project was motivated by an application of statistical theory to the Astronomy and more precisely to the statistical analysis of masses of giant molecular clouds. That is, the estimation of the distribution of the masses of molecular clouds in the galaxy M33 and the Large Magellanic Cloud (LMC), and to the mass distribution of H I (neutron hydrogen) clouds in the LMC. The analysis is presented in Chapter 5. Refer to Rosolowsky (2005) for a clear review of the importance of the mass spectra of giant molecular clouds.

1.5 Objectives

In summary, the work presented in this project aims to:

1. provide a review of the statistical literature on deconvolution of distribution functions;
2. draw more specific connections between certain techniques;
3. contrast the various methods;
4. develop methodology for deconvolution when the underlying distribution is known to be of power-law (i.e. Pareto) form;
5. apply the methodology to the estimation of the distribution of the masses of giant molecular and H I clouds.

1.6 Project Structure

Basic definitions of the methods used here are introduced in Chapter 2.

Chapter 3 consists of a study of the relevant deconvolution literature. The selection of the methods proposed in the literature is primarily based on three criteria namely popularity, appealing logic, and simplicity. The various methods are described in Chapter 3 in summarised form. Parametric estimation based on a sample drawn

from a population with a truncated Pareto distribution is dealt with in Chapter 4.

An application of the methodology to real data is reported in Chapter 5.

This project will conclude in Chapter 6 with recommendations and outlooks for possible extensions.

Computer programs, which were used to obtain the results reported, are given in the Appendices. Coding of the numerical routines is in MATLAB.

Chapter 2

Basic Definitions

The definitions given below are very useful.

2.1 Empirical distribution function

The CDF of the random variable Y is defined as $F_Y(y) = P(Y \leq y)$ and is estimated from the sample by the empirical distribution function (EDF)

$$\begin{aligned} F_n(y) &= \frac{1}{n}\,(\text{number of observations} \leq y) \\ &= \frac{\sum_{j=1}^{n} I(Y_j \leq y)}{n} \end{aligned}$$

where $I(Y_j \leq y)$ is an indicator function which assumes the value one, when the inequality is satisfied and zero when it is not. Therefore, in one dimension, the EDF is a step function with jumps of size $1/n$ at every point.

2.2 Kernel function

The kernel function K is a weighting function used in nonparametric density estimation (Silverman, 1986). The function K is usually, but not always, piecewise continuous, bounded, symmetric around zero, and for convenience often integrates to one. Kernel functions can be probability density functions, such as the normal density.

2.3 Goodness-of-fit of a statistical model

The goodness-of-fit of a statistical model describes how well the model fits the data. Probability-Probability (P-P) and Quantile-Quantile (Q-Q) plots are the most commonly used informal goodness-of-fit tests. Both P-P and Q-Q plots are used to see if a given set of data follows some specified distribution. These plots should be approximately linear if the specified distribution is the correct statistical model.

There are a number of statistical tests which can be performed that are more formal – the Kolmogorov-Smirnov (K-S), Anderson-Darling, Cramer-von-Mises are probably the best known. All three of these statistics measure discrepancy between the theoretical and empirical CDFs, and in that sense are quantitative (rather than visual) methods. Another well known goodness-of-fit test is the Chi-square (χ^2). It compares the observed and predicted numbers of data elements in selected intervals.

2.3.1 Probability-Probability plot

The probability-probability plot is constructed using the theoretical CDF, $F_Y(y)$, of the specified statistical model. $F_Y(y_{(j)})$ is plotted against the empirical CDF defined in Section 2.1, where $y_{(j)}$ is the j^{th} ordered sample observation $(j = 1, 2, \ldots, n)$. Parameters which occur in $F_Y(y)$ are replaced by estimated values. The same applies to other procedures discussed below.

2.3.2 Quantile-Quantile plot

For the quantile-quantile plot - $y_{(j)}$ is plotted against the quantile $F_Y^{-1}\left[(j - 0.5)/n\right]$.

2.3.3 The Kolmogorov-Smirnov test

The Kolmogorov-Smirnov test is based on the empirical CDF. The K-S test statistic is the maximum difference between the theoretical CDF and the empirical CDF:

$$D = \sup_y |F_Y(y) - F_n(y)|$$

2.3.4 The Anderson-Darling test

The Anderson-Darling test is a modification of the K-S test. The test statistic is

$$A^2 = -n - S$$

where

$$S = \sum_{j=1}^{n} \frac{(2j-1)}{n} \left[\log F_Y(y_{(j)}) + \log \left(1 - F_Y(y_{(n+1-j)}) \right) \right].$$

This test is one-sided, like those described in subsections 2.3.3, 2.3.5 and 2.3.6.

2.3.5 The Cramer-von-Mises test

The Cramer-von-Mises test is also based on the empirical CDF. The test statistic is

$$T = nW^2 = \frac{1}{12n} + \sum_{j=1}^{n} \left[\frac{(2j-1)}{2n} - F_Y(y_{(j)}) \right]^2$$

where

$$W^2 = \int_{-\infty}^{\infty} \left[F_Y(y) - F_n(y) \right]^2 dF_Y(y).$$

Tables of critical values (CVs) of these tests can be found in several texts. These CVs are valid when the distribution parameters are known. When the parameters are estimated from the data, these CVs are only approximate. One solution is to use a Monte Carlo method, which is based on generating a large number of distribution functions with the same population parameters and calculating the test statistic for each of the test cases, from which empirical values for quantiles can be extracted. This also applies to the tests in subsections 2.3.3, 2.3.4 and 2.3.6.

2.3.6 Chi-square test

The Chi-square goodness-of-fit test is applied to binned (i.e. grouped) data. This test is very simple to perform but has problems related to the choice of the number of intervals to use. Another disadvantage of the χ^2 test is that it requires a sufficiently large sample size in order for the χ^2 approximation to be valid. The χ^2 test is based

on the following test statistic:

$$T = \sum_{j=1}^{B} \frac{(O_j - E_j)^2}{E_j} \sim \chi^2_{B-m-1}$$

where O_j and E_j are respectively the observed and expected numbers of data elements in the j-th of the B bins, and m is the number of estimated parameters. In the present context the expected frequency is calculated by

$$E_j = n\left[F_Y(y_u) - F_Y(y_l)\right]$$

where y_u is the upper limit for bin j and y_l is the lower limit for bin j.

Chapter 3

Literature Review

This chapter, provides a description of various deconvolution procedures proposed in the literature. Both density and distribution estimation methods are reviewed in this chapter. The selection of the methods proposed is primarily based on popularity, appealing logic, and simplicity.

3.1 Nonparametric density estimation

3.1.1 Deconvolution kernel density estimator

The most popular approach to deconvolution is the use of a kernel estimator of f_X, obtained by applying the Fourier inversion formula to the empirical characteristic function of X. Contributions to the methodology have come from Liu and Taylor (1989), Carroll and Hall (1988), Stefanski and Carroll (1990), Fan (1992) amongst others. This technique can only be used if the density f_ϵ is fully specified and the characteristic function corresponding to the density f_ϵ is non-zero everywhere. The procedure is explained as follows:

The characteristic function of the random variable X is given by

$$\varphi_X(t) = \mathrm{E}\left(e^{itX}\right) = \int_{-\infty}^{\infty} e^{itx} f_X(x)\, dx$$

and the inverse transform is

$$f_X(x) = \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{-itx}\varphi_X(t)dt \tag{3.1}$$

where $\mathrm{E}(\cdot)$ denotes the expected value, i is the imaginary unit ($i = \sqrt{-1}$) and t is a real number.

Under the assumption that ϵ and X are independent, the characteristic function of the density f_Y is given by

$$\varphi_Y(t) = \mathrm{E}\left(e^{itX}\right)\mathrm{E}\left(e^{it\epsilon}\right) = \varphi_X(t)\cdot\varphi_\epsilon(t).$$

Then the characteristic function of X is

$$\varphi_X(t) = \frac{\varphi_Y(t)}{\varphi_\epsilon(t)}.$$

The problem now is to estimate the density f_X. Substituting in (3.1) the density f_X becomes

$$f_X(x) = \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{-itx}\frac{\varphi_Y(t)}{\varphi_\epsilon(t)}dt$$

and its estimator

$$\hat{f}_X(x) = \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{-itx}\frac{\hat{\varphi}_Y(t)}{\varphi_\epsilon(t)}dt. \tag{3.2}$$

The characteristic function $\varphi_Y(t)$ can be estimated by

$$\hat{\varphi}_Y(t) = \hat{\varphi}_n(t)\cdot\varphi_K(bt),$$

where $\hat{\varphi}_n(t) = n^{-1}\sum_{j=1}^{n} e^{itY_j}$ is the estimated empirical characteristic function corresponding to f_Y, $\varphi_K(t)$ is the characteristic function corresponding to the kernel function K and b is a smoothing parameter called the *bandwidth*: $b > 0$. The optimal selection of b is not obvious. One common way is to choose the bandwidth b that minimises the asymptotic mean integrated squared error. But the asymptotic mean integrated squared error also depends on the true underlying density f_X. One can consider alternative ways of selecting the bandwidth b. In the kernel estimator, the bandwidth b is treated as a function depending on the observations

$$b = \hat{b}(y_1, \ldots, y_n)$$

(Meister, 2009). Therefore, the observations $(y_1, \ldots, y_n)$ are used twice. First, to select the bandwidth, and, then, to estimate the density f_X where the $\hat{b}$ has been inserted. The fully data-driven methods, such as, cross validation and bootstrapping, can also be used for bandwidth selection. Cross validation is one of the most popular data-driven methods of bandwidth selection for the kernel estimator, in general. Diggle and Hall (1993) considered cross validation in the deconvolution kernel estimation.

Substituting $\hat{\varphi}_Y(t)$ in (3.2) the density estimator

$$\hat{f}_X(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{-itx} \hat{\varphi}_n(t) \frac{\varphi_K(bt)}{\varphi_\epsilon(t)} dt \tag{3.3}$$

follows. If the function $\varphi_K(bt)/\varphi_\epsilon(t)$ is integrable then the density estimator $\hat{f}_X$ in (3.3) can be represented in kernel form as

$$\hat{f}_X(x) = \frac{1}{nb} \sum_{j=1}^{n} K\left(\frac{x - Y_j}{b}\right) \tag{3.4}$$

where

$$K(z) = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{-itz} \frac{\varphi_K(t)}{\varphi_\epsilon(t/b)} dt. \tag{3.5}$$

The deconvolution kernel density estimator in (3.4) is just an ordinary kernel estimator but with specific kernel function (3.5). Furthermore, the kernel K has to be carefully chosen, to guarantee that the integral exists (Proença, 2003). The interested reader is referred to Stefanski and Carroll (1990) for details.

Remarks

1. Under the assumption that the measurement error is known to belong to the normal family with zero mean and variance σ^2, the characteristic function corresponding to the error density f_ϵ is

$$\varphi_\epsilon(t) = \mathrm{E}\left(e^{it\epsilon}\right) \quad = \quad \exp\left[-\frac{\sigma^2 t^2}{2}\right].$$

2. Stefanski and Carroll (1990) used the sinc kernel $K(y) = (\pi y)^{-1} \sin(y)$ having the characteristic function $\varphi_K(t) = I(-1, 1)(t)$ for Gaussian errors to ensure

the existence of the estimator. $I(-1,1)$ represents an indicator function (step function).

3. Delaigle (2008) used a kernel with characteristic function

$$\varphi_K(t) = (1-t^2)^3 \cdot I(-1,1)(t).$$

4. In Liu and Taylor (1989) the estimation of the characteristic function $\varphi_Y(t)$ was also based on a kernel density estimate of f_Y, but with finite limits of integration in (3.2).

3.1.2 Simple deconvolving kernel density estimator

A simple deconvolving kernel density estimator (Proença, 2003) avoids the typical numerical integration, making calculations much easier. The idea behind the proposed estimator is to replace the kernel function in (3.5) by the approximate deconvolving kernel equal to

$$K^*(z) = \phi(z) - \frac{\sigma^2}{2b^2}\phi''(z) \tag{3.6}$$

where $\phi(z)$ is the standard normal density function and $\phi''(z)$ is its second derivative. Consequently, a simple deconvolving kernel density estimator is

$$\hat{f}_X^*(x) = \frac{1}{nb}\sum_{j=1}^{n} K^*\left(\frac{x - Y_j}{b}\right)$$

where $K^*(\cdot)$ is symmetric and $\int \hat{f}_X^*(x)dx = 1$.

3.1.3 Low-order approximations in deconvolution

Some recent studies (e.g. by Carroll and Hall, 2004) argue that finding consistent estimators for the deconvolution problem is a goal that is often *unattainable* and, in practice, one may obtain better practical results by constructing a less ambitious low-order approximation of f_X, and accurately estimate that approximation rather than the density f_X. The kernel methods proposed by Carroll and Hall (2004) are based on the observation that one can express the expected value of a kernel estimator of f_X as a series expansion in expectations of kernel estimators of derivatives of

the density f_Y and that coefficients in the series depend only on moment derivatives of the distribution of ϵ.

The (ordinary) kernel density estimators of f_Y and f_X are given by

$$\hat{f}_Y(y) = \frac{1}{nb}\sum_{j=1}^{n} K\left(\frac{y - Y_j}{b}\right)$$
$$\hat{f}_X(x) = \frac{1}{nb}\sum_{j=1}^{n} K\left(\frac{x - X_j}{b}\right)$$

respectively. Note that the X_j are not observable, therefore $\hat{f}_X$ cannot be computed directly from the data. Nonetheless, Carroll and Hall (2004) constructed an approximation to $\hat{f}_X$. The method is explained as follows:

Assume that all central moments μ_k of the distribution of ϵ are finite and that the kernel K is an analytic function. That is, all derivatives of kernel K are defined on the whole real line. Denoting the k-th derivative of f by $f^{(k)}$, an estimation of the rth-order approximation to the expected value of $\hat{f}_X^{(k)}$, given by

$$\mathrm{E}\left\{\hat{f}_X^{(k)}(x)\right\} = \mathrm{E}\left\{\hat{f}_Y^{(k)}(x)\right\} + \sum_{v=1}^{\infty}\sum_{d_1=1}^{\infty}\cdots\sum_{d_v=1}^{\infty}\frac{(-1)^{d_1+\ldots+d_v+v}}{d_1!\ldots d_v!}\hat{\mu}_{d_1}\ldots\hat{\mu}_{d_v}\mathrm{E}\left\{\hat{f}_Y^{(d_1+\ldots+d_v+k)}(x)\right\}$$

is

$$\hat{f}_{X,r}^{(k)}(x) = \hat{f}_Y^{(k)}(x) + \sum_{v\geq 1} S_r \hat{f}_Y^{(d_1+\ldots+d_v+k)}(x) \tag{3.7}$$

where

$$S_r = \sum_{d_1,\ldots,d_v\geq 1 : d_1+\ldots+d_v\leq r} \frac{(-1)^{d_1+\ldots+d_v+v}}{d_1!\ldots,d_v!}\hat{\mu}_{d_1}\ldots\hat{\mu}_{d_v}$$

and $\hat{\mu}_k$ is an estimator of (or another type of approximation to) μ_k.

The assumption that the error ϵ is symmetric (e.g., normal) leads to $\mu_d = 0$ for odd d, and only approximations of even order are relevant. Then (3.7) simplifies to

$$\hat{f}_{X,2r}^{(k)}(x) = \hat{f}_Y^{(k)}(x) + \sum_{v\geq 1} S_{(2r)} \hat{f}_Y^{(2d_1+\ldots+2d_v+k)}(x)$$

where

$$S_{(2r)} = \sum_{d_1,\ldots,d_v \geq 1 : d_1+\ldots+d_v \leq r} \frac{(-1)^v}{(2d_1)!\ldots,(2d_v)!} \hat{\mu}_{(2d_1)} \cdots \hat{\mu}_{(2d_v)}.$$

The high order approximations can be expressed in terms of standard kernel density estimators as

$$\hat{f}_{X,r}^{(k)}(x) = \frac{1}{nb^{k+1}} \sum_{j=1}^{n} K_r^{(k)} \left(\frac{x - Y_j}{b} \right)$$

based on adjusted kernels

$$K_r(x) = K(x) + \sum_{v \geq 1} S_r K^{(d_1+\ldots+d_v)}(x).$$

The second-, fourth- and sixth-order approximations to $\mathrm{E}\left\{\hat{f}_X^{(k)}(x)\right\}$ (assuming that the error density is symmetric), are given by

$$\begin{aligned}
\hat{f}_{X,2}^{(k)}(x) &= \hat{f}_Y^{(k)}(x) - \frac{\hat{\mu}_2}{2} \hat{f}_Y^{(k+2)}(x); \\
\hat{f}_{X,4}^{(k)}(x) &= \hat{f}_{X,2}^{(k)}(x) + \frac{1}{24} \left(\hat{\mu}_2^2 - \hat{\mu}_4 \right) \hat{f}_Y^{(k+4)}(x); \\
\hat{f}_{X,6}^{(k)}(x) &= \hat{f}_{X,4}^{(k)}(x) - \frac{1}{720} \left(90\hat{\mu}_2^3 - 30\hat{\mu}_2\hat{\mu}_4 + \hat{\mu}_6 \right) \hat{f}_Y^{(k+6)}(x).
\end{aligned}$$

The first of these equations is commonly used in kernel density estimation, where it is usually assumed that $r = 2$. That is, the finite moment is $\mu_2 = \sigma^2$, therefore the estimator (approximation) proposed becomes

$$\hat{f}_X(x) = \hat{f}_Y(x) - \frac{1}{2}\sigma^2 \hat{f}_Y^{(2)}(x). \tag{3.8}$$

Remember that $\hat{f}_Y$ is the error-free kernel density estimator of f_Y, given earlier, and can be calculated from the observed data. $\hat{f}_Y^{(2)}$ is its second derivative. The only unknown in (3.8) is σ^2. Delaigle (2008) studied the properties of this low-order approximation developed by Carroll and Hall (2004) and estimated the unknowns via the empirical variance of the difference of replicated observations. On the other

hand, these parameters can be estimated by the method of moments via instrumental variables; see Delaigle (2008) for more details.

Remarks

1. A low-order approximation in deconvolution does not require any estimation of the characteristic function, which expresses detailed properties of the error distribution.

2. The technique enables estimation of the derivative of f_X, as well as f_X itself.

3.1.4 Density derivative deconvolution

This technique is based on estimating the derivatives of f_Y and expanding the inverse characteristic function $\varphi_\epsilon(t)$ in a power series and integrating term by term. From the inverse transform the $2k$th derivative of f_Y becomes

$$f_Y^{(2k)}(y) = \frac{1}{2\pi}\int_{-\infty}^{\infty} t^{2k}(-1)^k e^{-ity}\varphi_Y(t)\,dt$$

and for Gaussian errors with zero mean and known variance σ^2,

$$[\varphi_\epsilon(t)]^{-1} = \exp\left[\frac{\sigma^2 t^2}{2}\right] = \sum_{k=0}^{\infty}\frac{(\sigma t)^{2k}}{2^k k!}.$$

The characteristic function corresponding to f_X becomes

$$\varphi_X(t) = \sum_{k=0}^{\infty}\frac{(\sigma t)^{2k}}{2^k k!}\varphi_Y(t)$$

and the density

$$\begin{aligned} f_X(x) &= \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{-itx}\left[\sum_{k=0}^{\infty}\frac{(\sigma t)^{2k}}{2^k k!}\varphi_Y(t)\right]dt \\ &= \sum_{k=0}^{\infty}\frac{\sigma^{2k}}{2^k k!}(-1)^k f_Y^{(2k)}(x) \end{aligned} \tag{3.9}$$

follows. In principle, one can use (3.9) to form estimates of f_X from estimates of the derivatives of f_Y, though in practice the sum must be truncated - see examples with

illustrations (truncating the sum at $k = 1$ and $k = 2$) in Masry and Rice (1992). Also, in practice, the characteristic function $\varphi_Y(t)$ can be estimated from the kernel density of f_Y, specified earlier as $\hat{\varphi}_Y(t) = \hat{\varphi}_n(t)\varphi_K(bt)$. Therefore the estimator of f_X becomes

$$\hat{f}_X(x) = \sum_{k=0}^{\infty} \frac{\sigma^{2k}}{2^k k!}(-1)^k \hat{f}_Y^{(2k)}(x). \tag{3.10}$$

Truncating the sum at $k = 1$ in (3.10), the resulting estimator is equivalent to the one in (3.8). In this case the error variance σ^2 is assumed known.

3.1.5 Deconvolution via differentiation

Deconvolution via differentiation may be based on representing f_X in terms of Hermite polynomials generated by the error density f_ϵ (Masry and Rice, 1992). The density estimate $\hat{f}_X$ has the orthogonal series expansion given by

$$\hat{f}_X(x) = \sum_{k=0}^{\infty} \hat{a}_k H_k(x), \tag{3.11}$$

where

$$\hat{a}_k = \frac{1}{k!}\hat{f}_Y^{(k)}(0),$$

and $\hat{f}_Y^{(k)}(0)$ is the kth derivative kernel estimates of $f_Y(0)$. The proposed estimator is analogous to the orthogonal series method proposed by Carroll and Hall (2004).

3.1.6 Spline estimators

The proposed estimator (Chen, *et al.,* 2003) is a spline (piecewise polynomial) function that transforms the variable X into a standard normal. The transformation to normality makes distribution estimation more efficient for a large class of distributions frequently encountered in practice (Chen, *et al.,* 2003). The method uses two estimators, namely a weighted normal quantile regression estimator and a maximum likelihood estimator. A weighted quantile regression estimator is used as the starting value for the nonlinear optimization procedure of the maximum likelihood estimator. It is required that the MLE or *final estimator* be in a neighbourhood of the *initial estimator* or *weighted quantile regression estimator*. The proposed esti-

mators are transformed semi-parametric spline estimators that have a parametric form with the number of parameters chosen as a function of the data. The Akaike Information Criterion (AIC) may be used for order selection. The proposed estimator is constructed as follows:

Let $S(\cdot)$ represent a monotonic spline function, i.e., $S(\cdot)$ is restricted to be a nondecreasing function. The spline function (grafted polynomial) can be written as

$$S(t) = \mathbf{p}^T \mathbf{w}(t) \tag{3.12}$$

where $\mathbf{p}$ is the unknown parameter vector and $\mathbf{w}(t)$ is a vector of polynomial components. Assume that $S(X)$ has the standard normal distribution, that is, the CDF of X is

$$F_X(x) = \Phi[S(x)]. \tag{3.13}$$

Under assumption (3.13), the convolution integral of Y given in (1.2) becomes

$$f_Y(y) = \int_{-\infty}^{\infty} f_\epsilon(y - x)\Phi'\left[S(x)\right] S'(x)dx$$

where Φ is the CDF of the normal distribution, and primes denote first derivatives. The weighted quantile regression spline estimator of $\mathbf{p}$ is obtained in two major steps:

(i) weight estimation: EDF of X is obtained.

(ii) quantile regression: EDF of X is smoothed by normal transformation with a cubic spline.

Furthermore, let $S_0(x)$ be the grafted polynomial described in (3.12) with parameter vector $\mathbf{p}_0$. Let the weighted quantile regression estimator of $f_X(x)$ be

$$f_{wqr}(x) = \Phi'\left[\hat{\mathbf{p}}_0^T \mathbf{w}(x)\right] \hat{\mathbf{p}}_0^T \mathbf{w}'(x),$$

where $\hat{\mathbf{p}}_0$ is the weighted quantile estimator of $\mathbf{p}_0$. A mixture scheme was considered to locate the joinpoints (knots) of $\mathbf{p}_0$. The number of joint points used by $\mathbf{p}_0$ was selected by AIC. See Chen *et al.* (2003) for further discussion of weighted quantile regression.

The maximum likelihood estimator of f_X is in the interval

$$[L(x)f_{wqr}(x), U(x)f_{wqr}(x)]$$

for $0 < L(x) < 1$ and $U(x) > 1$. Then one can get

$$L(x)\hat{S}_0'(x) \leq S'(x) \leq U(x)\hat{S}_0'(x).$$

where $\hat{S}_0$ is the estimated spline transformation of the initial estimator with parameter $\mathbf{p}_0$. The maximum likelihood spline estimator can be obtained from the following nonlinear optimization problem with linear constraints (Chen *et al.*, 2003):

$$\max_{\mathbf{p}} \sum_{j=1}^{n} \log\{f_Y(y_j)\} \quad \text{subject to} \quad L(\cdot)\hat{S}_0'(\cdot) < \mathbf{p}^T\mathbf{w}'(\cdot) < U(\cdot)\hat{S}_0'$$

where

$$f_Y(y_j) = \int_{-\infty}^{\infty} f_\epsilon(y_j - x)\Phi'[\mathbf{p}^T\mathbf{w}(x)]\mathbf{p}^T\mathbf{w}'(x)dx. \tag{3.14}$$

3.1.7 Bayesian method

The deconvolution problem can be viewed in the format of an empirical Bayes problem (Stefanski and Carroll, 1990).

The model requires specifying a likelihood and a prior distribution for the parameters, the latter representing knowledge about the parameters prior to data collection (Carroll, 1997). The product of the prior and likelihood is the joint density of the data and the parameters. Using Bayes's Theorem, one can in principle obtain the posterior density, i.e., the conditional density of the parameters given the data. The posterior summarizes all of the information about the values of the parameters (Carroll, 1997).

3.2 Nonparametric distribution estimation

3.2.1 The CDF based on kernel methods

The distribution function F_X may be estimated by integrating the density f_X. Therefore, e.g. the kernel density estimator in (3.4) has the corresponding dis-

tribution function

$$\hat{F}_X(x) = \frac{1}{n}\sum_{j=1}^{n} W\left(\frac{x - Y_j}{b}\right)$$

where

$$W(u) = \int_{-\infty}^{u} K(t)dt.$$

3.2.2 Distribution derivative deconvolution

A similar approach to subsection 3.1.4 is based on deriving an inversion formula for F_X (instead of f_X) in terms of the derivatives of F_Y; the derivatives are then replaced by the difference quotients of the empirical distribution function F_n. The analogue of the inverse transform in (3.9) can be expressed as

$$F_X(y) = \sum_{k=0}^{n} \frac{\sigma^{2k} t}{2^k k!}(-1)^k F_Y^{(2k)}(y) \tag{3.15}$$

(Gaffey, 1959), and from (1.3) the $2k$-th convolution integral of F_Y is

$$F_Y^{(2k)}(y) = \frac{1}{\sqrt{\pi}(\sqrt{2\sigma})^{2k+1}} \int_{-\infty}^{\infty} H_{2k}\left(\frac{y-x}{\sqrt{2}\sigma}\right) \exp\left\{-\frac{(y-x)^2}{2\sigma^2}\right\} F_X(x)dx,$$

where $H_k(y)$ is the kth Hermite polynomial, given as

$$H_k(y) = (-1)^k \exp\left\{y^2\right\}\left(\frac{d}{dy}\right)^k \exp\left\{-y^2\right\}$$

(Pollard, 1953). The estimator of F_X is obtained by replacing $F_Y^{(2k)}$ in (3.15) by the $2k$-th difference quotients $F_n^{(2k)}$ of the empirical distribution . The estimator of F_X, for a sample of size n, will then be

$$\hat{F}_X(y) \approx \sum_{k=0}^{n} \frac{\sigma^2 t^k}{2^k k!}(-1)^k F_n^{(2k)}(y). \tag{3.16}$$

The sum can be further approximated by truncating at a small value of k such as $k = 1$. See Gaffey (1959) for further discussion about the proposed estimator.

3.2.3 Mixture modelling method

The mixture modelling method concerns modelling a statistical distribution by a mixture (or weighted sum) of other distributions. The idea is to describe the variable of interest X in terms of flexible distribution functions, which cover a wide range of possibilities including the normal distribution (Carroll, 1997).

Cordy and Thomas (1997) modelled F_X as a mixture of a finite number of known distributions and estimated the unknown proportions using the EM algorithm. Represent F_X as a mixture

$$F_X = \sum_{k=1}^{m} p_k F_k$$

where the F_k are known distribution functions (also called component distributions); p_k are unknown nonnegative constants satisfing $\sum_{k=1}^{m} p_k = 1$; and $m \geq 2$ is the number of components.

Under the mixture model, the distribution of Y can be expressed as

$$F_Y = \sum_{k=1}^{m} p_k \left(F_k * f_\epsilon\right) \tag{3.17}$$

where f_ϵ is the normal density with mean zero and known variance σ^2 and $*$ denotes convolution. The log-likelihood function for the mixing proportion $\mathbf{p} = (p_1, \ldots, p_m)^T$ based on the data $\mathbf{y}$ is given by

$$\mathcal{L}(\mathbf{y}; \mathbf{p}) = \sum_{j=1}^{n} \log \left(\sum_{k=1}^{m} p_k (f_k * f_\epsilon)(y_j) \right) \tag{3.18}$$

where f_k is the corresponding density function of F_k. The EM algorithm is quite simple to apply for the estimation of mixing proportions (Cordy and Thomas, 1997). The sufficient statistics for $\mathbf{p}$ from the complete data are the counts

$$C_k = \text{the number of sample values from population } k.$$

Given a prior estimate of $\mathbf{p}$, say $\mathbf{p}^{\text{old}}$, an updated estimate is obtained through the following step:

E step: For $k = 1, \ldots, m$ calculate the conditional expectations

$$\hat{C}_k = \mathrm{E}(C_k|\mathbf{y}, \mathbf{p}^{\mathrm{old}})$$

M step: Calculate the maximum likelihood estimate of $\mathbf{p}$, $\mathbf{p}^{\mathrm{new}}$, from the estimates $\hat{C}_1, \ldots, \hat{C}_m$:

$$p_k^{\mathrm{new}} = \hat{C}_k/n.$$

Once the final estimate, say $\widehat{\mathbf{p}} = (\widehat{p}_1, \ldots, \widehat{p}_m)^T$, of $\mathbf{p}$ is obtained, the corresponding estimate of F_X is given by

$$\hat{F}_X(t) = \sum_{k=1}^{m} \widehat{p}_k F_k(t).$$

Remarks

A simple choice for the component distributions is to take them to be normal with common variance, σ_c^2.

The choice of the number, m, of components and the value of σ_c^2 can have a large impact on the performance of the resulting estimator.

It was also noted by Cordy and Thomas (1997) that the variance of $\hat{F}_X$ usually increases as m increases.

On the other hand, when m is too small, the mixture model may provide only a poor approximation to the true distribution, in which case the bias of $\hat{F}_X$ will be unacceptably large. Refer to Cordy and Thomas (1997) regarding the choice of F_k and m for this problem.

Chapter 4

Fitting Pareto distributions to Data with Errors

4.1 Introduction

In many practical cases, information from a sample is used to draw conclusions about unknown population parameters. This is one of the most important tasks of statisticians. Usually researchers draw a random sample from a population and make some assumptions about the sample observations. For example, the assumption that the sample observations are drawn from one of a known parametric family of distributions (e.g., the normal density function), is common in statistics. The density function $f_Y(\cdot)$ underlying the sample could then be estimated by finding estimates of its mean μ and variance σ^2 from the set of observations $\{y_1, y_2, \ldots, y_n\}$ and substituting these estimates into the formula for the normal density. This is called a parametric approach. However, if the sample observations $\{y_1, y_2, \ldots, y_n\}$ are subject to measurement errors, the estimation of μ and σ^2 by fitting a normal density to contaminated observations y_j is not entirely appropriate.

Let the measured values be $y_j = x_j + e_j$ $(j = 1, \ldots, n)$, where x_j are the true (i.e., error-free) values of the quantity of interest, and e_j are errors. The distribution of $y_j = x_j + e_j$ is given by the convolution integral in (1.2). That is, the measurements y_j have now a convolved PDF. Consider the case where X is normal, but is observed subject to Gaussian measurement errors with mean zero. The convolution of the normal densities $f_X(\cdot)$ and $f_\epsilon(\cdot)$ with means μ_X and $\mu_\epsilon = 0$ and variances σ_X^2 and

σ_ϵ^2 is again a normal density, with mean $\mu = \mu_X$ and variance $\sigma^2 = \sigma_X^2 + \sigma_\epsilon^2$.

4.1.1 Pareto distribution

A Pareto distribution is a simple model for positive data with a power-law probability tail. Some examples from http://en.wikipedia.org/wiki/Pareto_distribution include the distribution of income and wealth among individuals, the sizes of human settlements, file size distribution of internet traffic which uses the TCP protocol, the values of oil reserves in oil fields, the length distribution in jobs assigned to supercomputers, sizes of sand particles and sizes of meteorites. The examples from the field of astronomy are the masses of molecular clouds, stellar initial mass functions, etc. Refer to Koen and Kondlo (2009) for more examples in this field. The truncated version of the Pareto distribution has a wide range of applications in several fields in data analysis (Zaninetti and Ferraro, 2008). In astronomy and many physical and social sciences, the parameters of the truncated Pareto are estimated to draw inference about the processes underlying the phenomena − that is, to test theoretical models and to scale up the local observations to global patterns (White *et al.*, 2008). Therefore, it is very important that these parameters be estimated accurately (White *et al.*, 2008 and Zaninetti and Ferraro, 2008). Often power-law indices, and other distributional parameters, are estimated by fitting the truncated Pareto form to a set of observations. This is also not appropriate if the measurements are contaminated by substantial measurement errors. A methodology for error-contaminated observations is derived and developed in the following sections.

4.2 Convolved Pareto distribution

The PDF of the truncated Pareto distribution is given by

$$f_X(x) = \frac{ax^{-a-1}}{L^{-a} - U^{-a}} \quad a > 0, \text{and} \quad 0 < L \leq x \leq U \tag{4.1}$$

and the zero mean Gaussian error density as

$$f_\epsilon(e) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{e^2}{2\sigma^2}\right] \quad -\infty < e < \infty;$$

where σ^2 is not necessarily known. Also of interest is the CDF corresponding to (4.1):

$$F_X(x) = \begin{cases} 0 & x < L \\ \frac{1-(L/x)^a}{1-(L/U)^a} & L \leq x \leq U \\ 1 & x > U \end{cases} . \tag{4.2}$$

Since the values of x are bounded in the interval $[L, U]$, the probability density function in (4.1) is known as the truncated Pareto density. The somewhat non-standard term "finite-support Pareto distribution" (FSPD) will be used in this thesis to designate a Pareto distribution defined on $[L, U]$, as opposed to the standard Pareto form with support $[L, \infty)$. In reality, the support does not range from L to ∞ but over a finite range. Accelerated life testing with samples censored is a good example (Nadarajah and Kotz 2006). For this reason, the focus is on the FSPD version of the Pareto distribution. The term "truncated Pareto distribution" will be reserved for data subject to truncation, i.e., subject to physical restrictions which constrain observations to lie in an interval $[l, u]$. Truncated distributions arise in many practical situations.

The convolved PDF of the FSPD and Gaussian error distributions is

$$\begin{aligned} f_Y(y) &= \int_{-\infty}^{\infty} f_X(x) f_\epsilon(y-x)\,\mathrm{d}x \\ &= \frac{a}{L^{-a}-U^{-a}} \frac{1}{\sigma\sqrt{2\pi}} \int_L^U x^{-a-1} \exp\left[-\frac{1}{2}\left(\frac{y-x}{\sigma}\right)^2\right] \mathrm{d}x \end{aligned} \tag{4.3}$$

and the corresponding convolved CDF is

$$\begin{aligned} F_Y(y) &= \int_{-\infty}^{y} f_Y(t)\,dt \\ &= \frac{a}{L^{-a}-U^{-a}} \int_L^U x^{-a-1} \left[\Phi\left(\frac{y-x}{\sigma}\right)\right] \mathrm{d}x \end{aligned} \tag{4.4}$$

where $\Phi(\cdot)$ is the CDF of the standard normal distribution. Alternatively, the convolved CDF can be derived from (1.3), i.e.

$$\begin{aligned} F_Y(y) &= \int_{-\infty}^{\infty} F_X(x) f_\epsilon(y-x)\,\mathrm{d}x \\ &= \frac{1}{1-(L/U)^a} \frac{1}{\sigma\sqrt{2\pi}} \int_L^U [1-(L/x)^a] \exp\left[-\frac{1}{2}\left(\frac{y-x}{\sigma}\right)^2\right] \mathrm{d}x + \Phi\left(\frac{y-U}{\sigma}\right). \end{aligned}$$

Also of interest is the standard Pareto distribution, i.e., x is in the interval $[L, \infty)$ and its PDF is given by

$$f_X(x) = aL^a x^{-a-1} \qquad a > 0, \quad \text{and} \quad L > 0 \tag{4.5}$$

and the corresponding CDF by

$$F_X(x) = 1 - L^a x^{-a} \qquad x \geq L.$$

The convolved PDF and CDF of a standard Pareto and Gaussian error distributions are

$$\begin{aligned} f_Y(y) &= \frac{aL^a}{\sigma\sqrt{2\pi}} \int_L^\infty x^{-a-1} \exp\left[-\frac{1}{2}\left(\frac{y-x}{\sigma}\right)^2\right] \mathrm{d}x \\ F_Y(y) &= \Phi\left(\frac{y-L}{\sigma}\right) - \frac{L^a}{\sigma\sqrt{2\pi}} \int_L^\infty x^{-a} \exp\left[-\frac{1}{2}\left(\frac{y-x}{\sigma}\right)^2\right] \mathrm{d}x. \end{aligned} \tag{4.6}$$

For computational purposes, $\Phi(\cdot)$ can be calculated by a special function called the *error function*:

$$\Phi\left(\frac{y-x}{\sigma}\right) = \frac{1}{2}\left[1 + \mathrm{erf}\left(\frac{y-x}{\sigma\sqrt{2}}\right)\right]$$

which is convenient for numerical work.

4.3 Unknown parameters

In practical application, the unknowns are the specific parameter values. These are the lower limit L, upper limit U, power-law index a (also known as the *exponent*) and the Gaussian error variance σ^2.

There are several parameter estimation techniques. However, some of the techniques could lead to bias and inaccurate estimates. From a statistical point of view, the method of maximum likelihood is more robust and yields estimators with good statistical properties. Recent relevant references are Goldstein *et al.* (2004), Zaninetti and Ferraro (2008), White, *et al.* (2008). The standard errors for the estimated parameters are derived from both the inverse Fisher information matrix and the jackknife methods.

The fit of the derived distributions is then assessed using different methods: graphical assessment (probability-probability plot) and goodness-of-fit tests (Chi-squared and Kolmogorov-Smirnov). A bootstrapping method is used to calculate the critical values of the Kolmogorov-Smirnov test statistic, which are not available for real data.

4.4 Estimation method

4.4.1 Maximum likelihood estimation

Maximum likelihood estimation (MLE) is one of the preferred methods for estimating parameter values. MLE can only be used if the form of the underlying distribution is known. The method is based on maximising the likelihood of the observed sample given the statistical model. Specifically, MLE finds the parameter values that maximise the product of the PDFs at each of the observed values (assuming the observations are independent). The method was pioneered by Sir R. A. Fisher between the years 1912 and 1922.

Let $\theta = [L, U, a, \sigma]'$ represent the vector of parameters and $\widehat{\theta} = [\hat{L}, \hat{U}, \hat{a}, \hat{\sigma}]'$ the vector of estimates. Then, for independent observations $\{y_1, y_2, \ldots, y_n\}$, the likelihood for the density $f_Y(y; \theta)$ is of the form

$$\ell(y_1, y_2, \cdots, y_n; \theta) = \prod_{j=1}^{n} f_Y(y_j; \theta). \tag{4.7}$$

The likelihood function of the PDF given in (4.3) is

$$\ell(y_j; \theta) = \left[\frac{a}{L^{-a} - U^{-a}} \frac{1}{\sqrt{2\pi\sigma^2}}\right]^n \prod_{j=1}^{n} \int_L^U x^{-a-1} \exp\left[-\frac{1}{2}\left(\frac{y_j - x}{\sigma}\right)^2\right] \mathrm{d}x. \tag{4.8}$$

It is equivalent but often mathematically easier to maximise the log-likelihood function instead of the likelihood function itself. The log-likelihood function correspond-

ing to (4.8) is given by

$$\begin{aligned}\mathcal{L} = \log(\ell) &= n\left[\log(a) - \log(L^{-a} - U^{-a}) - \log(\sigma) - \frac{1}{2}\log(2\pi)\right] \\ &+ \sum_{j=1}^{n} \log \int_L^U x^{-a-1} \exp\left[-\frac{1}{2}\left(\frac{y_j - x}{\sigma}\right)^2\right] \mathrm{d}x. \end{aligned} \tag{4.9}$$

According to the method of maximum likelihood the estimates of L, U, a and σ are chosen so as to maximize the observed likelihood function in (4.8), or equivalently, the log-likelihood function in (4.9). The maximum likelihood estimators of θ are the simultaneous solutions of the m equations:

$$\frac{\partial \mathcal{L}}{\partial \theta_k} = 0, \quad k = 1, 2, \ldots, m. \tag{4.10}$$

The derivatives are given in Appendix A. Solving for L, U, a and σ from the first partial derivatives in Appendix A is difficult. Alternatively, the best values of the parameters can be obtained by direct numerical maximisation of the log-likelihood function in (4.9). This is relatively easy using a mathematical and engineering calculation computer language such as MATLAB, MAPLE or MATHEMATICA. The procedure was implemented in MATLAB. The program is in Appendix C.

The maximum likelihood estimates of the parameters for the standard Pareto distribution in (4.5) and the FSPD in (4.1) are given by:

$$\begin{aligned}\hat{L} &= \min(x_1, x_2, \ldots, x_n) \\ \hat{a} &= n\left[\sum_{j=1}^{n}\left(\log x_{(j)} - \log \hat{L}\right)\right]^{-1}\end{aligned}$$

for $U \leftrightarrow \infty$ and

$$\begin{aligned}\hat{L} &= \min(x_1, x_2, \ldots, x_n) \\ \hat{U} &= \max(x_1, x_2, \ldots, x_n) \quad \text{and} \quad \hat{a} \quad \text{solves} \\ \frac{n}{\hat{a}} + \frac{n r^{\hat{a}} \log r}{1 - r^{\hat{a}}} &= \sum_{j=1}^{n}\left[\log x_{(j)} - \log \hat{L}\right]\end{aligned} \tag{4.11}$$

where $r = \hat{L}/\hat{U}$, for finite U. The last part of equation (4.11), for $\hat{a}$, is implicit (Aban, *et al.* 2006). The maximum likelihood estimates of the parameters of convolved standard Pareto and Gaussian error distributions are not provided in this study. The interested reader is referred to Koen and Kondlo (2009) for the latter.

We note in a passing that estimates for L and U of form (4.10) do not apply to the standard Pareto distribution since derivatives with respect to L and U are not defined in that case.

4.5 Measures of statistical accuracy

Standard errors may be used to provide an indication of the size of the uncertainty and to provide approximate confidence intervals. The standard errors of the parameter estimates are calculated by two different methods: from the Fisher information matrix and by a jackknife procedure. The bootstrap method can also be used to calculate the standard errors of the parameter estimates.

4.5.1 Fisher information matrix

The Fisher information matrix of the estimates has elements

$$\mathtt{F}_{ij} = -\mathrm{E}\left[\frac{\partial^2 \mathcal{L}}{\partial\theta_i\theta_j}\right] \qquad i,j = 1,\ldots,m, \tag{4.12}$$

where m is the dimension of the parameter vector θ. Evaluating the expectations in (4.12) is tedious. Alternatively, an approximation in the form of the empirical information follows by substituting directly the observed data into the second derivatives. Then the Fisher information matrix has elements

$$\mathcal{F}_{ij} = -\left[\frac{\partial^2 \mathcal{L}}{\partial\vartheta_i\vartheta_j}\right] \qquad i,j = 1,\ldots,m.$$

The standard errors are the square roots of the diagonal elements of $[\mathcal{F}_{ij}]^{-1}$. The second derivatives for this Fisher information matrix are in Appendix A. The procedure was implemented in MATLAB - see the program in Appendix C.

4.5.2 The jackknife method

The jackknife and bootstrap are nonparametric computer-intensive techniques for estimating (e.g.) standard errors of the estimated parameters. The jackknife procedure consists of taking subsamples of the original sample of n independent observations by omitting a single observation at a time. Thus, each subsample consists of n - 1 observations formed by deleting a different observation from the sample. Parameter estimates are then calculated from these subsamples. Standard errors are determined from the variability across the n sets of parameter estimates. A more detailed description of the jackknife method proceeds as follows:

Let $\widehat{\theta}$ be the vector of parameter estimates obtained by MLE from the sample observations $\{y_1, \ldots, y_n\}$. Divide the sample into g subgroups (at random if $g < n$) of size k. Then from each subgroup, re-estimate θ from the remaining $(g-1)k$ observations. This provides the g partial estimates $\widehat{\theta}^{(-j)}$, $j = 1, \ldots, g$. Form the pseudo-values (the jackknife replications)

$$\theta_{*j} = g\widehat{\theta} - (g-1)\widehat{\theta}^{(-j)}.$$

The jackknife estimate of θ is the average of the jackknife replications θ_{*j}, that is

$$\theta_{(J)} = \frac{1}{g}\sum_{j=1}^{g} \theta_{*j} = g\widehat{\theta} - (g-1)\bar{\widehat{\theta}};$$

where

$$\bar{\widehat{\theta}} = \frac{1}{g}\sum_{j=1}^{g} \widehat{\theta}^{(-j)}.$$

The corresponding estimated covariance matrix is

$$\widehat{C}_J(\theta) = \frac{g-1}{g}\sum_{j=1}^{g} \left(\theta_{*j} - \theta_{(J)}\right)\left(\theta_{*j} - \theta_{(J)}\right)'. \tag{4.13}$$

The procedure was implemented in MATLAB with $k = 1$; i.e. $g = n$. See the MATLAB program in Appendix C.

4.5.3 The bootstrap method

In case of bootstrapping, pseudo-samples of size n are produced by drawing with *replacement* from the original n data elements. The number of pseudo-samples would typically be of the order of a thousand or more, hence computing time would be exorbitant for the problem under discussion. The procedure is as follows:
Draw B independent bootstrap samples y_b^* of size n from the sample observations $\{y_1, \ldots, y_n\}$. Estimate the parameters θ for each y_b^*. This provides B bootstrap replicates of $\widehat{\theta}$ i.e., $\widehat{\theta}_b^*$ for $b = 1, \ldots, B$. For each component θ_i of the vector θ, order the bootstrap replicates such that $\widehat{\theta}^*_{i(1)} \leq \cdots \leq \widehat{\theta}^*_{i(B)}$. The lower and upper confidence bounds are $B \cdot \alpha$ and $B \cdot (1-\alpha)$ ordered elements. The estimated confidence interval of $\widehat{\theta}_i$ is

$$\left[\widehat{\theta}^*{}_{i(B\cdot\alpha)}; \widehat{\theta}^*{}_{i(B\cdot(1-\alpha))}\right].$$

The estimated covariance matrix is

$$\widehat{C}_B(\theta) = \frac{1}{B-1} \sum_{b=1}^{B} \left(\widehat{\theta}_b^* - \bar{\widehat{\theta}}^*\right)\left(\widehat{\theta}_b^* - \bar{\widehat{\theta}}^*\right)' \tag{4.14}$$

where $\bar{\widehat{\theta}}^* = B^{-1} \sum_{b=1}^{B} \widehat{\theta}_b^*$. Efron and Tibshirani (1993) is a readable introduction to both the bootstrap and jackknife methods.

4.6 Simulation study

A simple experiment is used to validate the proposed methodology. Datasets of sizes $n = 50$ and $n = 300$ are drawn from a FSPD and normal deviates of fixed dispersion are added. Assumed parameter values $L = 3$, $U = 6$, $a = 1.5$ and $\sigma = 0.4$ are used.

4.6.1 The effects of measurement errors

Histograms in Figure 4.1 show some of the pitfalls caused by the presence of measurement errors. The top panel shows a histogram of a sample from a FSPD. A histogram of the same data, with added measurement errors, is given in the bottom panel. Two effects are clearly visible: the contaminated data extend beyond the interval $[L, U]$ over which the pure data occur, and the shape of the distribution is changed. The first effect will clearly bias estimates of the lower and upper limits,

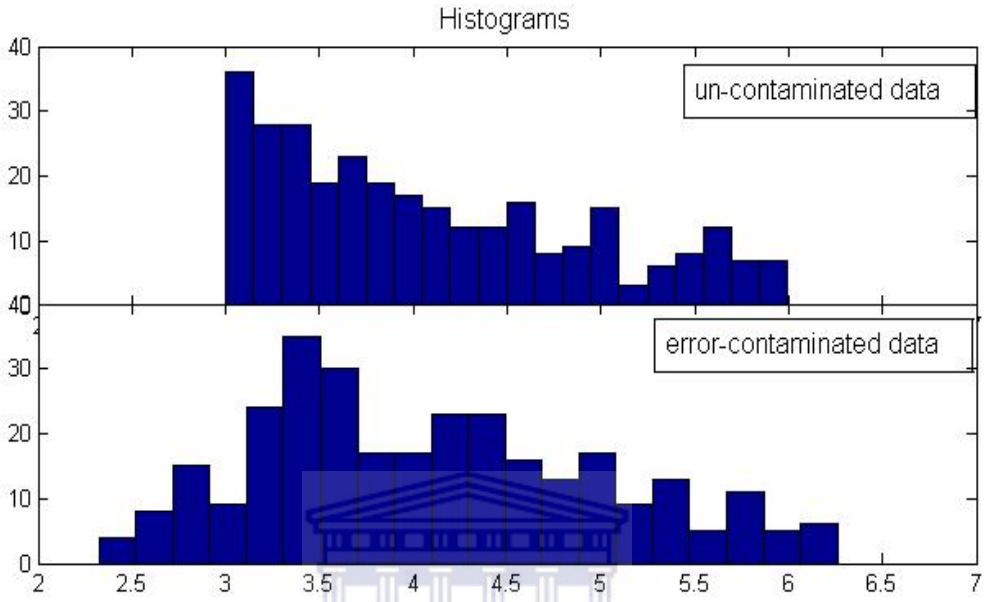

Figure 4.1: The top histogram is for 300 simulated data elements from a FSPD with $L = 3$, $U = 6$ and $a = 1.5$. Zero mean Gaussian measurement errors with $\sigma = 0.4$ were added to give the convolved distribution in the bottom panel. The error-contaminated data "spill" out of the interval $[L, U]$.

while the second will lead to biased estimates of the power law exponent: in particular, since the data are spread over a wider interval, the value of index a estimated from contaminated data may generally be too small.

The convolved PDF in (4.3) could differ substantially from the FSPD in (4.1). The probability-probability plots based on these forms can be compared to distinguish the difference between the distributions (4.3) and (4.1).

4.6.2 Assessing quality of fit of the model

Without some assessment of goodness-of-fit of the model the parameter estimates have very limited meaning. We illustrate the importance of verifying that the statistical model fitted to the data is appropriate − making use of the goodness-of-fit tests described in section 2.3 of Chapter 2.

An informal goodness-of-fit test, the P-P plot, is used to demonstrate the importance of verifying that the statistical model fitted to the data is appropriate. The theoretical distribution values are obtained by first estimating the parameters occuring in the theoretical CDFs (i.e., L, U, a and σ); $F_Y(\cdot)$ can then be explicity evaluated in each observation value y_j. See Chapter 2 for extensive discussion. The data used in the histrogram above were used to create Figure 4.2. The nonlinear form in the top panel, corresponding to the FSPD, i.e., assuming zero measurement errors, convincingly demonstrates that the distribution (4.2) does not describe the data. On the other hand, the bottom panel shows that (4.4) is a good model for the data. Note that for plots values of the parameters estimated from (4.11) and (4.9) have been substituted in order to calculate the CDFs in (4.2) and (4.4).

MLEs and the corresponding 95% confidence intervals for L, U, a and σ from the data with measurement errors are given in Table 4.1. Note that the convolved PDF in (4.3) fitted to the data gives favourable results with true parameter values $L = 3$, $U = 6$, a $= 1.5$, and $\sigma = 0.4$, in particular for larger n.

The Kolmogorov-Smirnov and other goodness-of-fit tests discussed in Chapter 2 are not distribution-free in this context, therefore percentage points need to be found either by a Monte-Carlo method or bootstrapping. The Kolmogorov-Smirnov test is selected because it is widely used in practice. The percentage points for the Kolmogorov-Smirnov test statistic are obtained by bootstrapping.

MLE	$\hat{L}$	$\hat{U}$	$\hat{a}$	$\hat{\sigma}$	K-S p-values
$n = 50$	3.0919	6.4370	1.6971	0.4455	0.980
CI	[2.67; 3.38]	[5.73; 7.25]	[1.3E-7; 3.99]	[4.6E-8; 0.59]	
$n = 300$	3.0258	5.9989	1.6048	0.3967	1.00
CI	[2.85; 3.19]	[5.79; 6.22]	[0.60; 2.78]	[0.29; 0.48]	

Table 4.1: MLEs for two different sample sizes ($n = 50$ and $n = 300$) are provided. Confidence intervals at 95% level calculated from $B = 1000$ bootstrap samples. Percentage points of the K-S statistic (obtained from bootstrapping) are given in the last column.

P-values of the K-S statistic given in Table 4.1 indicate lack of statistical significance. They are larger than expected: for $n = 50$ and $n = 300$, the p-values are respectively

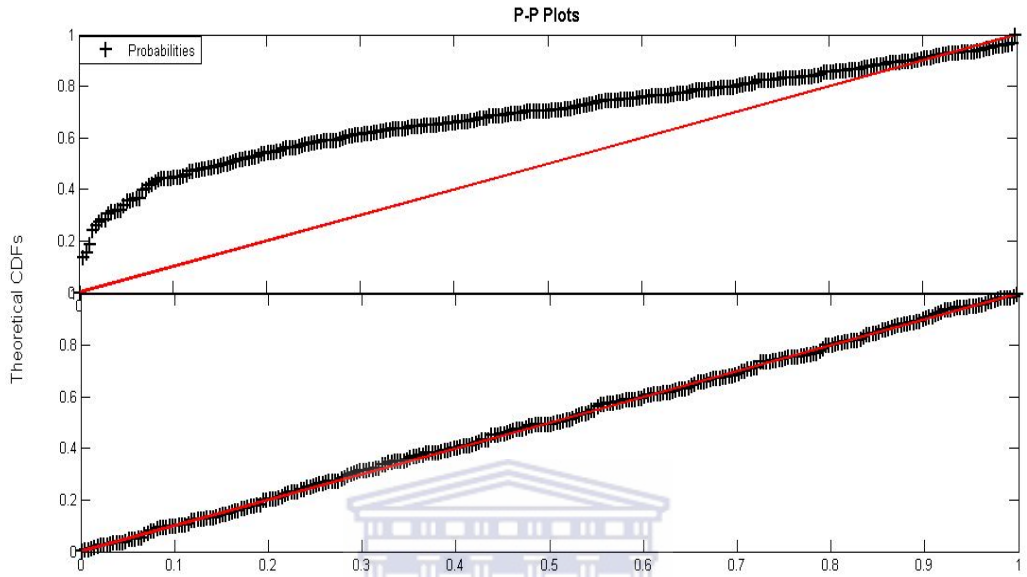

Figure 4.2: Probability-probability plots for a simulated dataset, consisting of 300 values distributed according to (4.3). The plot in the top panel is based on the (incorrect) assumption that there is no measurement error; the plot in the bottom panel incorporates Gaussian measurement errors.

0.98 and 1.00. This means that 98% or 100% of the percentage points calculated from $B = 1000$ bootstrap samples are larger than the actual K-S test statistic.

4.6.3 Bootstrapping - sample size

Figure 4.3 shows the histrogram of the bootstrap estimates for $n = 50$. The results have been confirmed by 200 Monte Carlo simulations. Figure 4.4 shows the histogram of the bootstrap replications for $n = 300$.

An excessive amount of computer time is needed for large sample sizes (n a few hundred or more) - primarily required for the maximisation of the log-likelihood function in (4.9).

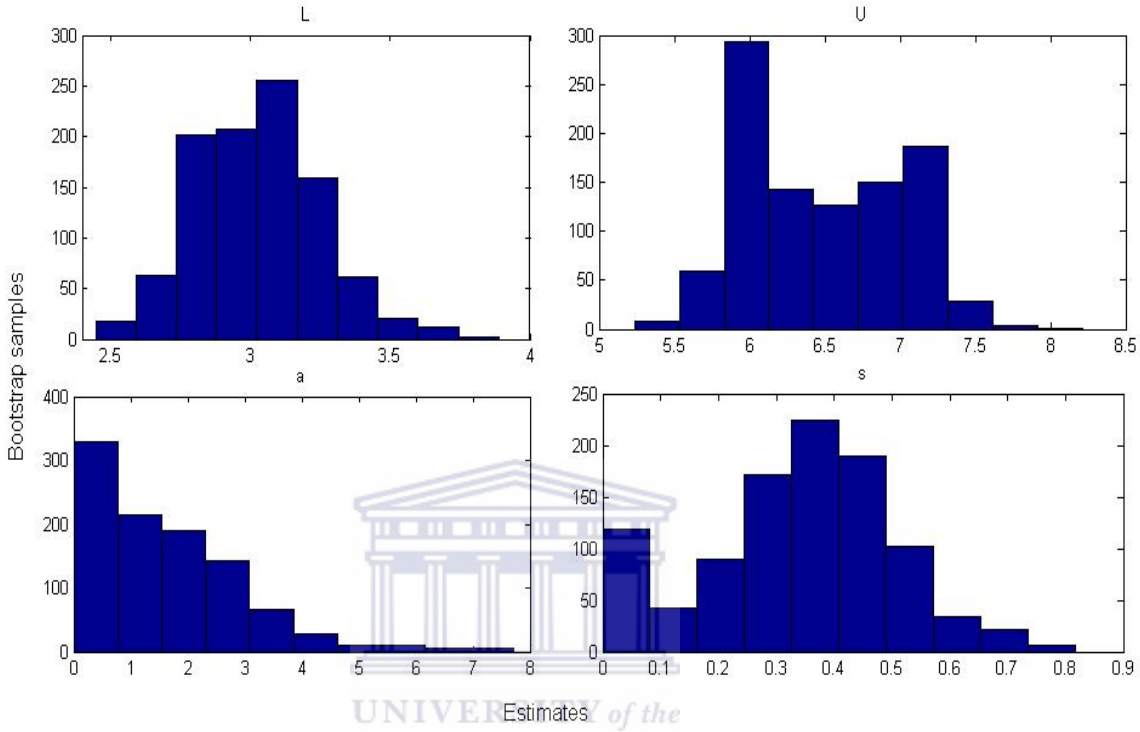

Figure 4.3: Distributions of the estimated parameters for the $n = 50$ sample, from 1000 bootstrap samples.

4.6.4 Comparison of covariance matrices

We also compare the covariance matrix results calculated from the inverse Fisher information matrix, bootstrapping and the jackknife method for the $n = 50$ sample. For the parameter vector $[\hat{L}\ \hat{U}\ \hat{a}\ \hat{\sigma}]$, the respective covariance matrices are

$$C_F = \begin{pmatrix} 0.0419 & 0.0084 & 0.1974 & 0.0150 \\ 0.0084 & 0.2319 & 0.3020 & -0.0201 \\ 0.1974 & 0.3020 & 1.8221 & 0.0490 \\ 0.0150 & -0.0201 & 0.0490 & 0.0181 \end{pmatrix}$$

$$C_B = \begin{pmatrix} 0.0495 & 0.1127 & 0.2982 & 0.0372 \\ 0.1127 & 0.2712 & 0.6806 & 0.0845 \\ 0.2982 & 0.6806 & 1.9091 & 0.2202 \\ 0.0372 & 0.0845 & 0.2202 & 0.0297 \end{pmatrix}$$

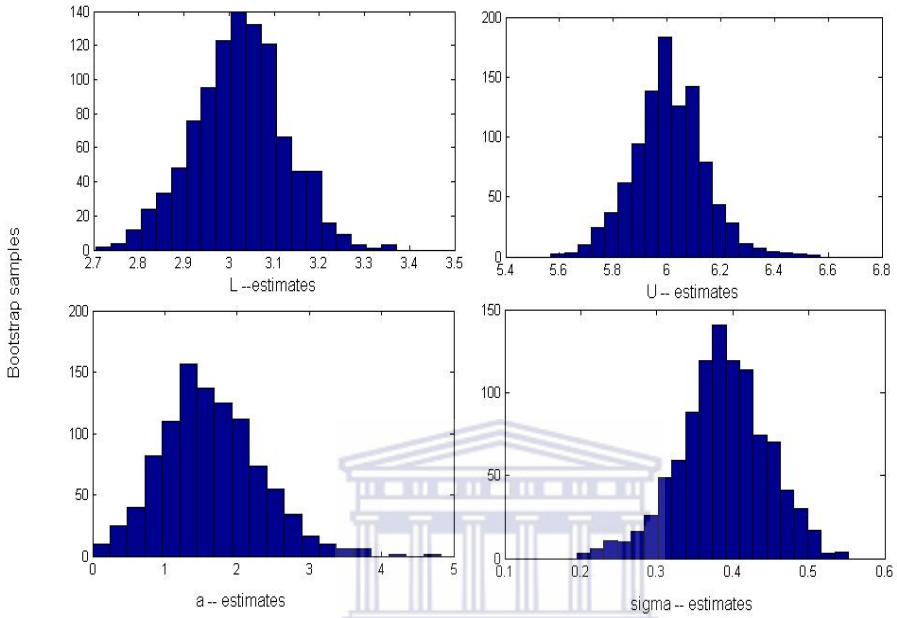

Figure 4.4: As for Figure 4.3 , but for a sample size $n = 300$. The distributions are much closer to normal than for $n = 50$.

$$C_J = \begin{pmatrix} 0.0684 & 0.0945 & 0.3910 & 0.0351 \\ 0.0945 & 0.3293 & 0.8853 & 0.0417 \\ 0.3910 & 0.8853 & 3.3338 & 0.1824 \\ 0.0351 & 0.0417 & 0.1824 & 0.0338 \end{pmatrix}$$

The agreement is reasonable, especially between the matrices calculated from the nonparametric computer-intensive techniques (Jackknife and Bootstrap). The standard errors from the matrices are compared in Table 4.2.

	L	U	a	σ
$\widehat{s.e}(Fish)$	0.205	0.482	1.350	0.135
$\widehat{s.e}(Boot)$	0.223	0.521	1.382	0.172
$\widehat{s.e}(Jack)$	0.262	0.574	1.826	0.184

Table 4.2: The standard errors calculated from the Fisher information, the bootstrap and jackknife matrices

4.6.5 Deconvolving density functions

The aim in deconvolution is to recover the unknown density functions from contaminated observations. An example when X is a normal random variable can be seen in Figure 4.5. The estimated density is not far from the true density of X. The method used is the nonparametric penalised contrast estimator for adaptive density deconvolution proposed by Comte, *et al.* (2006). The method is based on model selection, more precisely by minimisation of a penalised contrast function. The method requires the error variance σ to be chosen. The error density, which was taken to be $N(0, 1)$, is assumed to be fully known.

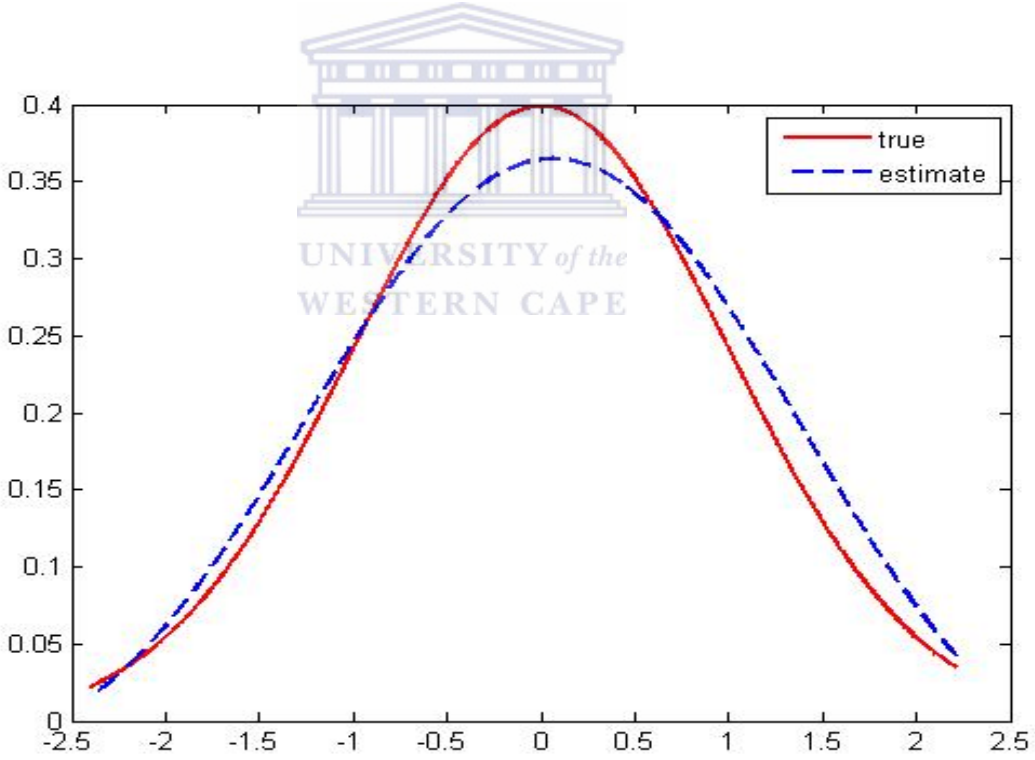

Figure 4.5: A simulated example with X (the true density) a standard normal random variable. The Gaussian error distribution is taken to have mean zero and variance equal to one. The sample size is $n = 500$. A nonparametric penalised adaptive method is used.

Consider the case when X is of Pareto form. Unlike the case when X is normal, the resulting estimator could differ very significantly from the FSPD. As an illustration, the nonparametric deconvolution and parametric methods were applied to simulated data of the same form as discussed above (FSPD plus noise), with sample size

$n = 300$. The results are shown in Figure 4.6.

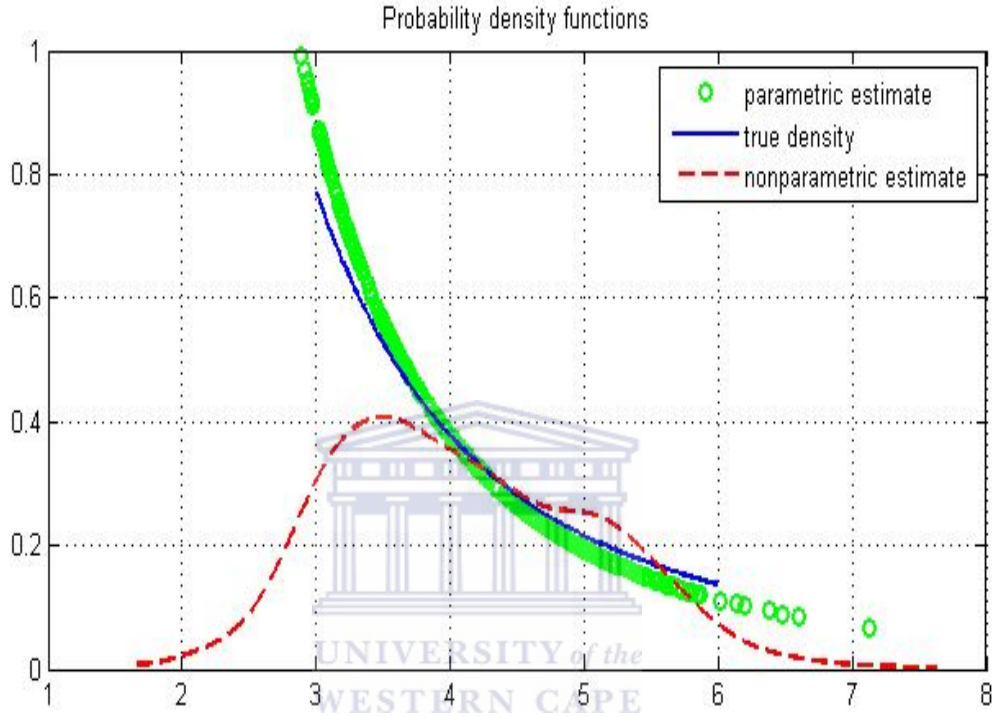

Figure 4.6: The nonparametrically deconvolved distribution is compared to the true underlying FSPD and the parametrically deconvolved distributions. The true parameter values are $L = 3$, $U = 6$, $a = 1.5$ and $\sigma = 0.4$.

Notes

The shape of the nonparametric density estimate in Figure 4.6 is not appropriate. Neither is the method designed to estimate the finite interval over which the FSPD is defined. Note also that a small difference in the specified error variance values could lead to completely different deconvolved density estimates. Therefore, the choice of the error variance is very critical in this problem.

Application of nonparametric methods is much less restricted than parametric methods, due to fewer assumptions made. However, if the assumptions made in parametric methods are correct, parametric methods can produce more accurate and precise estimates as they did in the above examples. They are said to have more statistical power. But, if those assumptions are incorrect, parametric methods can

be very misleading.

The power of a statistical test is the probability that the test will reject a false null hypothesis (that is will not make a Type II error). As power increases, the chances of a Type II error decreases.

4.7 Testing for specific power-law distributional forms

Some specific distributional forms of (4.3) are often of interest. Several types of null hypotheses could also be of interest. Some are

(i) There may be theoretical models, or other datasets, which suggest particular parameter values, and one may be interested in testing for conformity with these. A common example is to test whether the power-law exponent $a = c$, for specified c.

(ii) The other type of null hypothesis involves whether a simplified distributional form, such as an unrestricted Pareto or FSPD in (4.5) or (4.1), rather than the distributional form of (4.3), provides an adequate description of the data. Three example are: $U \to \infty$, for which (4.3) reduces to (4.6); $\sigma = 0$ for which (4.3) reduces to FSPD in (4.1); and $U \to \infty$, $\sigma = 0$ which provides the standard power-law distributional form in (4.5).

Hypotheses can often be tested by likelihood ratio test statistics

$$2\left[\max \mathcal{L}\left(H_1\right) - \max \mathcal{L}\left(H_0\right)\right] \sim \chi^2_d$$

where H_0 and H_1 denote the null and alternative hypotheses respectively. The maxima of the log-likelihood are determined under both hypotheses. The statistic has an asymptotic χ^2_d distribution, with degrees of freedom d equal to the number of constraints imposed by H_0. Hypotheses of the form (i) can be tested by this procedure.

The same procedure cannot be used for hypotheses like those in (ii), since required regularity conditions are not satisfied (Koen and Kondlo, 2009). Model selection is

a simple way of viewing the problem for these hypotheses. In that case information criteria such as Akaike and Bayes for the competing models can be compared. The Akaike and Bayes Information Criterion are

$$\begin{aligned} AIC &= -2\mathcal{L} + 2K + \frac{2K(K+1)}{n-K-1} \\ BIC &= -2\mathcal{L} + K\log n \end{aligned}$$

where K is the number of model parameters. The likelihood term in these criteria measures how well the model fits the data; since it appears as the negative of the likelihood, the term is small for good fit. The remaining terms are a measure of the model complexity − simple models (i.e., small values of K) are preferred. It is therefore desirable to have both terms as small as possible, i.e. the "best" model is that which minimises the information criterion.

In practice, the AIC is usually best when increased model complexity leads to incrementally better fits, while the BIC performs best for data which can be modelled very well with simple models. The model probabilities are

$$p_i = \frac{\Delta_i}{\sum \Delta_i}$$

where

$$\Delta_i = \exp\left[-\frac{1}{2}\left(IC_i - IC_{min}\right)\right]$$

and i indexes the model, and IC_{min} is the minimum value of the information criterion (either AIC or BIC). The model with the largest probability is then selected. For extensive discussion and application to the problem under consideration consult Koen and Kondlo (2009).

4.8 Truncation of the Pareto distribution

Restriction of the distributional form of (4.3) to the interval $[u, l]$ where $-\infty < l \leq L < U \leq u < \infty$ may arise. The PDF of the truncated density f_Y is given as

$$t_r(y) = \frac{f_Y(y)}{F_Y(u) - F_Y(l)} \quad l < y \leq u \tag{4.15}$$

and its corresponding CDF by

$$T_r(y) = \frac{F_Y(y) - F_y(l)}{F_Y(u) - F_Y(l)}$$

Truncated distributions can also be seen as conditional distributions.

The likelihood for the truncated convolved density $t_r(y;\theta)$ in (4.15) is of the form:

$$\ell_r(y_1, y_2, \cdots, y_n; \theta) = \left[\frac{1}{F_Y(u) - F_Y(l)}\right]^n \prod_{j=1}^{n} f_Y(y_j;\theta)$$

and the log-likelihood

$$\mathcal{L}_r = \sum_{j=1}^{n} \log f_Y(y_j;\theta) - n \log\left[F_Y(u) - F_Y(l)\right] \tag{4.16}$$

where

$$\begin{aligned}\sum_{j=1}^{n} \log\left[f_Y(y_j;\theta)\right] &= n\left[\log(\beta) - \log(\sigma) - \frac{1}{2}\log(2\pi)\right] \\ &+ \sum_{j=1}^{n} \log \int_L^U x^{-a-1} \exp\left\{-\frac{(y_j - x)^2}{2\sigma^2}\right\} dx\end{aligned}$$

and

$$\log\left[F_Y(u) - F_Y(l)\right] = \log(\beta) + \log \int_L^U x^{-a-1} \left\{\Phi\left(\frac{u-x}{\sigma}\right) - \Phi\left(\frac{l-x}{\sigma}\right)\right\} dx$$

where $\beta = a/\left(L^{-a} - U^{-a}\right)$. Therefore the log-likelihood becomes

$$\begin{aligned}\mathcal{L}_r &= -n\log(\sigma) - \frac{n}{2}\log(2\pi) \\ &- n\log \int_L^U x^{-a-1} \left\{\Phi\left(\frac{u-x}{\sigma}\right) - \Phi\left(\frac{l-x}{\sigma}\right)\right\} dx \\ &+ \sum_{j=1}^{n} \log \int_L^U x^{-a-1} \exp\left\{-\frac{(y_j - x)^2}{2\sigma^2}\right\} dx\end{aligned} \tag{4.17}$$

The first and second partial derivatives of the log-likelihood in (4.17) are given in appendix B.

4.9 A computational detail

Evaluation of the log-likelihood is computationally expensive, since n convolution integrals need to be calculated. A problem arises when integrating a concentrated function such as $\exp[-0.5(y-x)^2]$ with respect to x over a wide interval. The Matlab built-in functions "quad" or "quadl" do not subdivide the integration interval finely enough and misses where the integrand is non-zero. This can easily be seen by integrating the function above over the interval $[-50, 50]$ (assume $y = 0$). The answer should of course be $\sqrt{2\pi}$, but "quad" gives 7.9E-19 and "quadl" gives 2.3E-22. The solution is to identify the sub-interval over which the integrand is non-negligible, and to restrict the integration to this sub-interval only.

Limiting the integration domain is particularly pertinent in the case of the unrestricted Pareto distribution, for which the upper limit $U \to \infty$. Fortunately, numerical determination of the interval over which the integrand is non-negligible is straightforward.

Chapter 5

Analysis of real data

5.1 Introduction

The developed methodology is applied to the estimation of the distribution of the masses of giant molecular clouds (GMCs) in the galaxy M33 and in the Large Magellanic Cloud (LMC), and to the mass distribution of H I clouds in the LMC. GMCs and H I clouds are massive clouds of interstellar gas and dust observable by radio telescope. The mass of the cloud is determined from the intensity of its radio radiation at a wavelength of 21-cm. The method used to measure cloud masses is subject to measurement errors (Rosolowsky, 2005).

The intention is to demonstrate the application of the theory, rather than to derive definitive results. Therefore, for example, questions regarding the quality of published data are not addressed (Koen and Kondlo, 2009).

Satisfactory results of fitting convolved distributions to GMC masses in the two galaxies are demonstrated below.

5.2 GMCs in M33

The data analysed in this section were drawn from Engargiola *et al.* (2003). There are 148 GMCs in their catalogue. The results of fitting the convolved distributions to the masses of GMCs in the galaxy M33 are shown in Table 5.1.

The estimated exponent $\hat{a} = 1.33$ is in good agreement with the Engargiola *et al.* (2003) estimate of 1.6 ± 0.3, but their lower mass limit $L = 4$ is rather different from the value $\hat{L} = 6.9382$ obtained here. The estimated measurement error $\hat{\sigma} = 3.48$ is bound to be a severe underestimate of the errors in the largest clouds. Since there are many more small clouds than large, it seems safe to assume that $\hat{\sigma}$ will be determined primarily by masses close to L.

The probability-probability plot in Figure 5.1 is approximately linear, indicating that the data come from the convolved distribution.

The estimated covariance matrices obtained from Fisher information and the jackknife are

$$C_F = \begin{pmatrix} 0.4200 & 0.4655 & 0.1343 & 0.2259 \\ 0.4655 & 24.7046 & 0.2617 & 0.3783 \\ 0.1343 & 0.2617 & 0.0687 & 0.0803 \\ 0.2259 & 0.3783 & 0.0803 & 0.3197 \end{pmatrix}$$

and

$$C_J = \begin{pmatrix} 0.5480 & 0.0638 & 0.1670 & 0.3181 \\ 0.0638 & 6.3346 & -0.0332 & 0.0625 \\ 0.1670 & -0.0332 & 0.0749 & 0.1030 \\ 0.3181 & 0.0625 & 0.1030 & 0.3133 \end{pmatrix}$$

There appears to be reasonable agreement, except for covariances involving $\hat{U}$. We speculate that this is due to the structure of the Pareto fitting problem: much more data are required for accurate estimation of U, since only the largest (and therefore most scanty) data determine the value of $\hat{U}$. This means that large sample approximations (such as Fisher information) of covariances of $\hat{U}$ will be poorer than

	$\hat{L}$	$\hat{U}$	$\hat{a}$	$\hat{\sigma}$
MLE	6.9382	77.7202	1.3336	3.4755
$\widehat{s.e}(Fish)$	0.6480	4.9704	0.2622	0.5654
$\widehat{s.e}(Jack)$	0.7403	2.5169	0.2737	0.5598

Table 5.1: The results of fitting the convolved distribution to the masses of GMCs in the galaxy M33. The estimated parameters with associated standard errors are provided. The unit of mass is 10^4 solar masses.

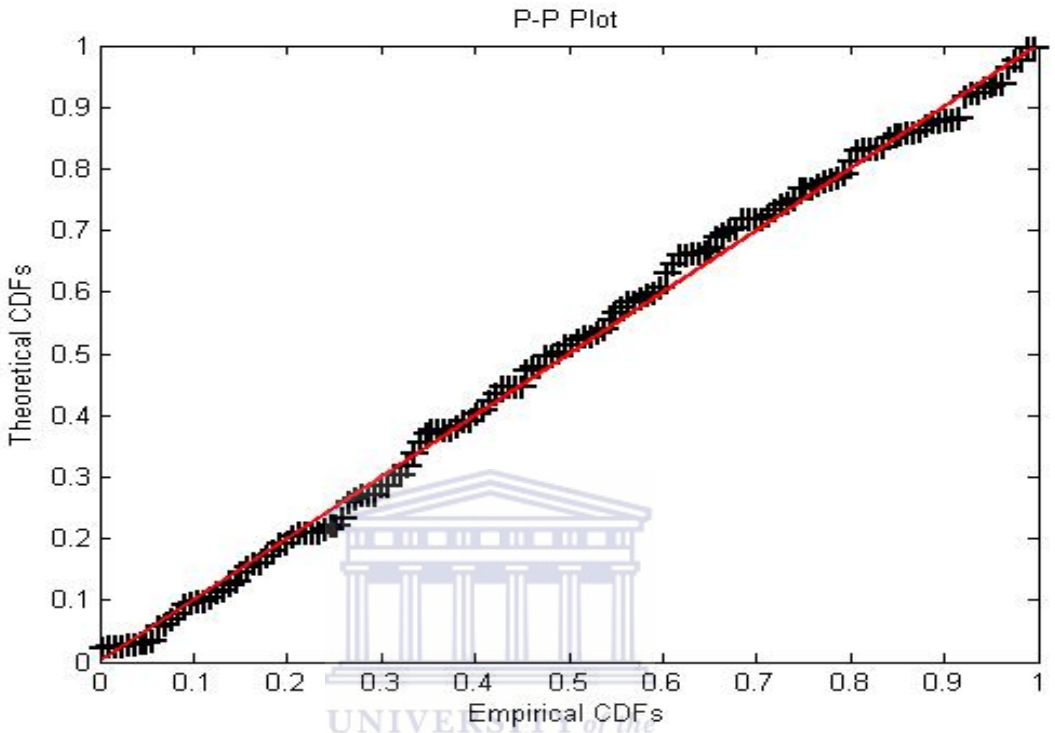

Figure 5.1: Probability-probability plot for the galaxy M33 data.

covariances involving other parameters. If this is correct, then the jackknife covariance matrix is preferred for datasets of this order.

The significance levels of χ^2 goodness-of-fit statistics for various numbers of binning intervals are given in Table 5.2.

number of bins	χ^2 statistic p-values
$B = 10$	0.50
$B = 15$	0.66
$B = 20$	0.28

Table 5.2: The significance levels of χ^2 goodness-of-fit statistics for various binning intervals - Engargiola *et al.* (2003) data.

5.3 H I clouds in the LMC

There are 195 H I cloud masses in the catalogue of Kim *et al.* (2007). The modelling results are presented in Table 5.3.

	$\hat{L}$	$\hat{U}$	$\hat{a}$	$\hat{\sigma}$
MLE	3.47	2915.9	0.56	0.82
$\widehat{s.e}(Fish)$	0.195	14.846	0.054	0.178
$\widehat{s.e}(Jack)$	0.183	1409.4	0.051	0.169

Table 5.3: The results of fitting the convolved distribution to the masses of H I clouds in the LMC. The estimated parameters with associated standard errors are provided. The unit of mass is 10^3 solar masses.

$M_{(191)}$	$M_{(192)}$	$M_{(193)}$	$M_{(194)}$	$M_{(195)}$
866.9	1019.6	1167.5	1495.9	2913.0

Table 5.4: Five largest masses of the LMC H I clouds. The unit of mass is 10^3 solar masses.

Interestingly, there is a larger difference between the asymptotic and the jackknife standard error estimates than we saw in Table 5.1, despite the dataset being larger. A contributory factor is the extent of the high mass tail of the LMC H I cloud distribution (see Table 5.4): the 190 lowest masses are in the interval $2.2 - 550.4 \times 10^3$ solar masses.

This isolation of the large mass value (the second largest mass is 1496×10^3 solar masses) accounts for the very large standard errors of $\hat{U}$ in Table 5.3. Removing the largest mass, $M_{(195)}$, does not affect the estimates for $\hat{L}$ and $\hat{a}$ by much ($\hat{L} = 3.66$ and $\hat{a} = 0.60$), but the estimated measurement error is increased to $\hat{\sigma} = 0.98$, and $\hat{U}$ is dramatically reduced to 1501.

The estimated exponent $\hat{a} = 0.56$ is in reasonable agreement with the value 0.68 found by Kim *et al.* (2007) for the same dataset. The linear form of the probability-probability plot in Figure 5.2 indicates that the estimated distribution fits the data very well.

The significance levels of χ^2 goodness-of-fit statistics are given in Table 5.5.

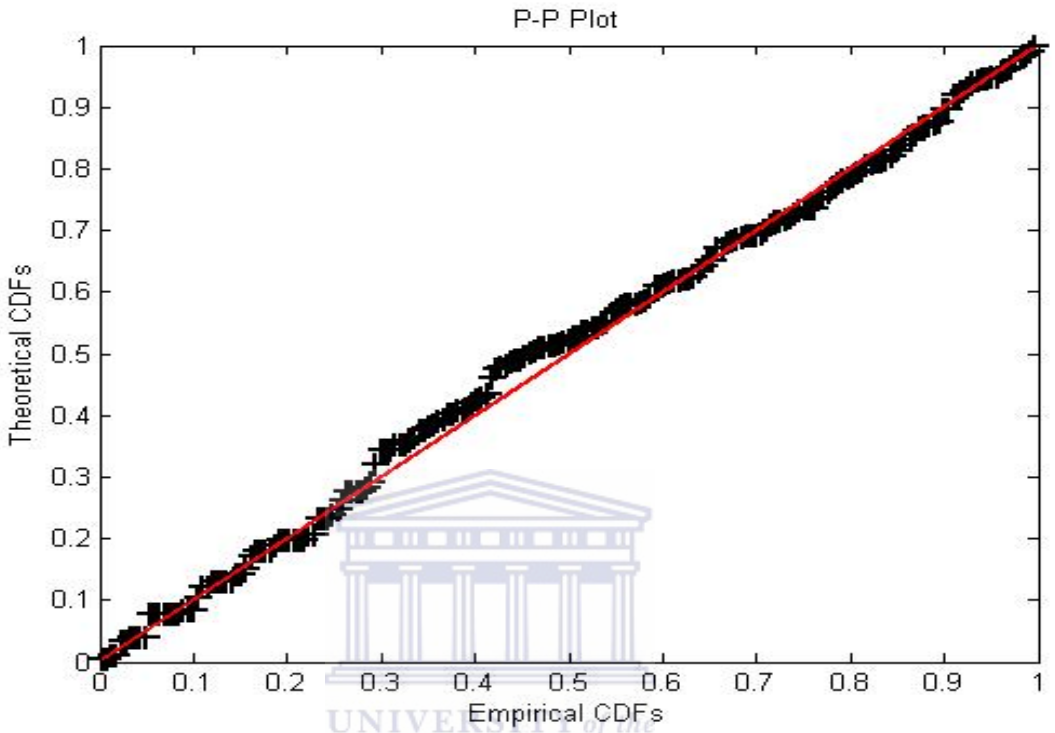

Figure 5.2: Probability-probability plot for the LMC data.

5.4 GMCs in the LMC

A feature of the catalogue of GMC masses in the LMC (Fukui *et al.*, 2008) is that values are only given to one significant digit. There are only 21 distinct values among the 230 masses in the catalogue, ranging from 0.1 to 100 $\times 10^5$ solar masses. It is therefore not possible to compare the theoretical and the empirical distributions using statistics such as goodness-of-fit tests. The MLEs are presented in Table 5.6. Figure 5.3 contains a P-P plot for these data.

number of bins	χ^2 statistic p-values
$B = 10$	0.30
$B = 15$	0.60
$B = 20$	0.63

Table 5.5: The significance levels of χ^2 goodness-of-fit statistics for various binning intervals - Kim *et al.* (2007) data.

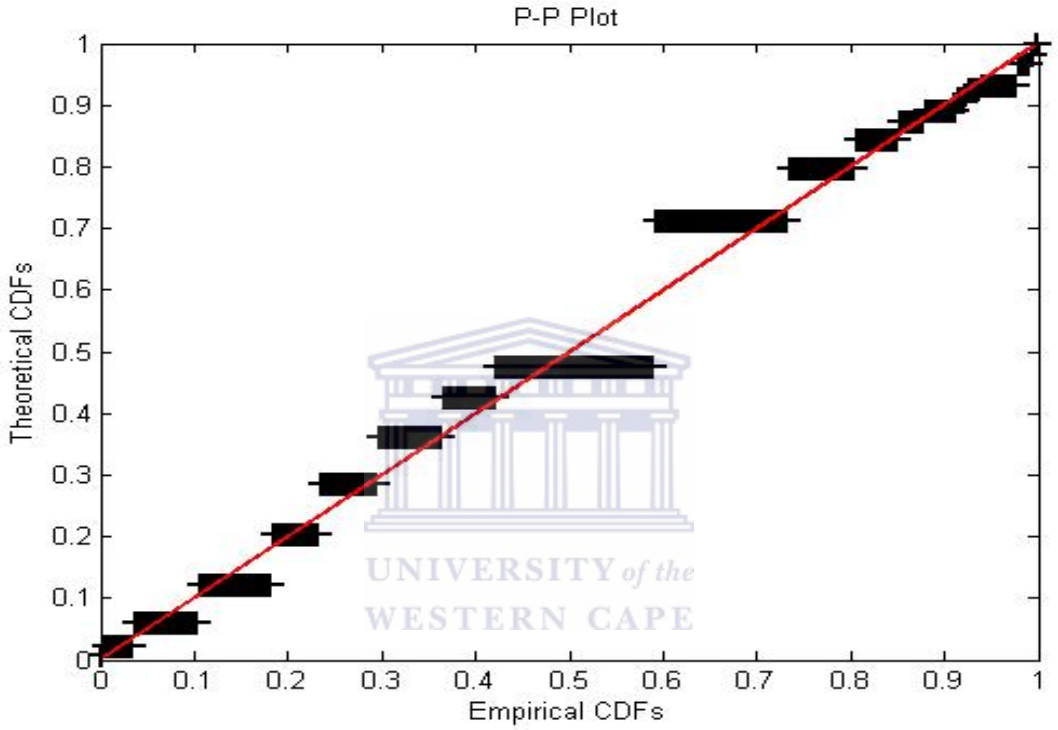

Figure 5.3: Probability-probability plot for the LMC data.

	$\hat{L}$	$\hat{U}$	$\hat{a}$	$\hat{\sigma}$
MLE	0.4491	100.4950	0.8196	0.1616

Table 5.6: The results of fitting the convolved distribution to the masses of GMCs in the LMC. The estimated parameters are provided. The unit of mass is 10^5 solar masses.

Chapter 6

Conclusion

Deconvolution is a useful statistical method for estimating an unknown distribution in the presence of measurement error. The most widely used nonparametric deconvolution estimator in the literature is the deconvoluting kernel density estimator. The assumption that the measurement errors have a Gaussian distribution is common. Few studies deal with the case of unknown measurement errors properties.

Researchers from other fields of study, such as astronomy, econometrics, etc. have not always fully realised the usefulness of deconvolution methods. Astronomers often rely on work by Lucy (1974), performed more than 30 years ago to tackle this problem.

The assumption of power-law probability distributions is common in the astronomical literature. The methodology for deconvolution when the underlying distribution is known to be of power-law form is developed in this thesis. Satisfactory results were found by MLE.

The model for the measurement errors is obviously very restrictive. It seems very likely that in most settings measurement errors will depend on the true values of the variable. For example, in the case of GMCs a model such as $\sigma = a + bx$ with a and b constant, seems reasonable (Rosolowsky, 2005). This complicates the analysis, since σ can no longer be treated as a constant − it must be included in the integrand. Also, measurement errors cause the data to "spill" from the interval $[L, U]$, particularly near the lower limit L where the probability density $f_X(\cdot)$ is largest.

This cause the tailing off of data with decreasing $x(x < L)$, giving the impression of data incompleteness. The point is illustrated in Figure 4.1, which shows histograms for simulated data with and without, measurement errors. The data in the bottom panel appears to be complete over [3,6.5], and incomplete for smaller y. In actual fact there is no incompleteness. Determination of completeness limits is therefore not entirely straightforward. A brute force way of dealing with this is to select a conservative completeness interval, and to ignore all data outside the interval. In this regard, the likelihood function given in section 4.8 (and appendix B) would be useful. The price paid is that the analysis is more complicated − furthermore, if the completeness interval is too small, it may no longer be possible to determine a lower limit L and an upper limit U.

Comparison of the computation times of the covariance matrices C_F and C_J is also of interest. The time taken to compute the covariance matrices C_F and C_J for the Engargiola *et al.* (2003) dataset with $n = 148$ were, respectively, 4.7 seconds and 6.2 hours (Acer 3273 WXMi, clock speed 1.66 GHz, 80 GB HDD). The computation time of C_J could be speeded up considerably by relaxing the convergence criteria invoked when maximising likelihoods, but it would still be orders of magnitude larger than the time required to calculate C_F. For larger n, the computational time of C_F rises slowly (most of it expended on the calculation of integrals in Appendix A), while the computation of C_J becomes prohibitive for datasets of the order of a few hundreds. Therefore, the Fisher information matrix (constituted of the second partial derivatives in Appendix A) would be very useful in such instances.

A simulation based method of inference for parametric measurement error models called Simulation-Extrapolation (SIMEX) first proposed by Cook and Stefanski (1994) and elaborated upon by Stefanski and Bay (1996), could also be useful. SIMEX estimates are obtained by adding additional measurement error to the data in a resampling stage, establishing a trend of measurement error induced bias versus the variance of the added measurement error, and extrapolating this trend back to the case of no measurement error. This methodology has not been exploited for parametric density deconvolution.

Other possible extensions will be to look at (i) other distributions; (ii) truncated samples and (iii) the case where σ is no longer treated as constant, that is, where it

depends on the true values of x.

Bibliography

[1] Aban, I. B., Meerschaert, M. M. and Panorska, A. K. (2006). Parameter estimation for the truncated Pareto distribution. *Journal of the American Statistical Association* **101**, 270-277

[2] Billingsly, P. (1994). *Probability and Measure (Third edition)* Wiley, New York.

[3] Bissell, A. F. and Ferguson, R. A. (1975). The jackknife - toy, tool or two-edged weapon? *The Statistician* **6**, 79-100.

[4] Britt, H. I. and Luecke, R. H. (1976). The estimation of parameters in nonlinear, implicit models. *Journal of the American Statistical Association* **15** 233-247.

[5] Carroll, R. J. (1997) Measurement error in epidemiologic studies. Texas A&M University. [Online]. http://www.stat.tamu.edu/ftp/pub/rjcarroll/gail03.pdf

[6] Carroll, R. J. and Hall, P. (1988). Optimal rates of convergence for deconvolving a density. *Journal of the American Statistical Association* **83**, 1184-1186.

[7] Carroll, R. J. and Hall, P. (2004). Low order approximations in deconvolution and regression with errors in variables. *Journal of the Royal Statistical Society B* **66**, 31-46.

[8] Carroll, R. J. Ruppert, D. and Stefanski, L. A. (1995). *Measurement Error in Nonlinear Models*. Chapman and Hall, New York.

[9] Chen, C., Fuller, W. A. and Breidt, F. J. (2003). Spline estimator of the density function of a variable measured with error. *Communications in Statistics* **32**, 73-86.

[10] Comte, F., Rozenholc, Y. and Taupin, M-L. (2006). Penalized contrast estimator for adaptive density deconvolution. *The Canadian Journal of Statistics* **34**, 431-452.

[11] Cook, J. R. and Stefanski, L. A. (1994). Simulation extrapolation estimation in parametric measurement error models. *Journal of the American Statistical Association* **89**, 1314-1324.

[12] Cordy, C. B. and Thomas, D. R. (1997). Deconvolution of a distribution function. *Journal of the American Statistical Association* **92**, 1459-1465.

[13] Delaigle, A. (2008). An alternative view of the deconvolution problem. *Statistica Sinica* **18**, 1025-1045.

[14] Delaigle, A. and Gijbels, I. (2007). Frequent problems in calculating integrals and optimizing objective functions: a case study in density deconvolution. *Statistics and Computing* **17**, 349-355.

[15] Devroye, L. (1989). Consistent deconvolution in density estimation. *The Canadian Journal of Statistics* **17**, 235-239.

[16] Diggle, P. J. and Hall, P. (1993). A Fourier approach to nonparametric deconvolution of a density estimate. *Journal of the Royal Statistical Society B* **55**, 523-531.

[17] Efromovich, S. (1997). Density estimation for the case of supersmooth measurement error. *Journal of the American Statistical Association* **56**, 526-535.

[18] Efromovich, S. (1985). Nonparametric estimation of a density. *Theory of probability and its applications* **29**, 2473-2491.

[19] Efron, B. and Tibshirani, R. J. (1993). *An Introduction to the Bootstrap*. Chapman and Hall, London.

[20] Eltinge, J. L. (1999). Accounting for non-Gaussian measurement error in complex survey estimators of distribution functions and quantiles. *Statistica Sinica* **9**, 425-449.

[21] Engargiola, G., Plambeck, R. L., Rosolowsky, E. and Blitz, L. (2003). Giant molecular clouds in M33. I. Bima All-Disk survey. *The Astrophysical Journal Supplement Series* **149**, 343-363.

[22] Fan, J. (1992). Deconvolution with supersmooth distribution. *The Canadian Journal of Statistics* **20**, 155-169.

[23] Fukui *et al.* (2008). The Second survey of molecular clouds in the Large Magellanic Cloud for NANTEN I: catalog of molecular clouds. *The Astrophysical Journal Supplement Series* **178**, 56-70.

[24] Fuller, W. A. (1995). Estimation in the presence of measurement error. *International statistical review* **63**, 121-141.

[25] Gaffey, W. R. (1959). A consistent estimator of a component of a convolution. *The Annals of Mathematical Statistics* **30**, 198-205.

[26] Goldstein, M. L., Steven A. M. and Yen G. G. (2004). Problems with fitting to the power-law distribution. *European Physical Journal B* **41**, 255-258.

[27] Hahn, B. (2002). *Essential MATLAB for Scientists and Engineers*. A Maskew Miller Longman Company, Cape Town.

[28] Kim, S., Rosolowsky E., Lee, Y., Kim Y., Jung, Y.C., Dopita, M.A., Elmegreen, B. G., Freeman, K.C., Sault, R.J., Kesteven M., McConnell D., and Chu Y.-H. (2007). A catalog of H I clouds in the Large Magellanic Cloud. *The Astrophysical Journal Supplement Series* **171**, 419-446.

[29] Koen, C. (2006). On the upper limit on stellar masses in the Large Magellanic Cloud cluster R136. *Monthly Notices of the Royal Astronomical Society* **365**, 590-594.

[30] Koen, C. and Kondlo, L. (2009). Fitting power law distributions to data with measurement errors. *Monthly Notices of the Royal Astronomical Society* **397**, 495-505.

[31] Lucy, L. B. (1974). An iterative technique for the rectifivation of observed distributions. *The Astronomical Journal* **79**, 745-754.

[32] Liu, M. C. and Taylor, R. L. (1989). A consistent nonparametric density estimator for the deconvolution problem. *The Canadian Journal of Statistics* **17**, 399-410.

[33] Masry, E. and Rice, J. A. (1992). Gaussian deconvolution via differentiation. *The Canadian Journal of Statistics* **20**, 9-21.

[34] Massey, F. J. (1951). The Kolmogorov-Smirnov test for goodness-of-fit. *Journal of the American Statistical Association* **46**, 68-78.

[35] Meister, A. (2006). Density estimation with normal measurement error with unknown variance. *Statista Sinica* **16**, 195-211.

[36] Meister, A. (2009). *Deconvolution Problems in Nonparametric Statistics* Springer-Verlag, Berlin.

[37] Mendelsohn, J. and Rice, J. (1982). Deconvolution of microflourometric histograms with B splines. *Journal of the American Statistical Association* **77**, 748-753.

[38] Mood, A. M. Graybill, F. A. and Boes, D. C. (1974). *Introduction to the theory of statistics (3rd edition)*. McGraw-Hill, Auckland.

[39] Myung, I. J. (2003). Tutorial on maximum likelihood estimation. *Journal of Mathematical Psychology* **47**, 90-100.

[40] Nadarajah, S. and Kotz, S. (2006). **R** Programs for computing truncated distributions. *Journal of Statistical Software* **16**, *Code Snippet 2*.

[41] Neumann, M. H. (2007). Deconvolution from panel data with unknown error distribution. *Journal of Multivariate Analysis* **98**, 1955-1968.

[42] Pollard, H. (1953). Distribution functions containing a Gaussian factor. *Proceedings of the American Mathematical Society* **4**, 578-582.

[43] Proença, I. (2003). A simple deconvolving kernel density estimator when noise is Gaussian. Universidade Tecnica de Lisboa. [Online]. http://129.3.20.41/eps/em/papers/0508/0508006.pdf

[44] Quenouille, M. (1956). Notes on bias and estimation. *Biometrika* **43**, 353-360.

[45] Rosolowsky, E. (2005). The mass spectra of Giant Molecular Clouds in the Local Group. *Astronomical Society of the Pacific* **117**, 1403-1410.

[46] Silverman, B. W. (1986). *Density Estimation for Statistics and data Analysis*. Chapman and Hall, London.

[47] Stefanski, L. (2000). Measurement error models. *Journal of the American Statistical Association* **95**, 1353-1358.

[48] Stefanski, L. A. and Bay, J. M. (1996). Simulation extrapolation deconvolution of finite population cumulative distribution function estimators. *Biometrika* **83**, 407-417.

[49] Stefanski, L. A. and Carroll, J. R. (1990). Deconvoluting kernel density estimators. *Statistics* **21**, 169-184.

[50] Turlach, B. A. and Hazelton, M. L. (2008). Nonparametric density deconvolution by weighted kernel estimators. National University of Singapore. *Unpublished manuscript*.

[51] van Es, B., Jongbloed, G. and van Zuijlen, M. (1998). Isotonic inverse estimators for nonparametric deconvolution. *The Annals of Statistics* **26**, 2395-2406.

[52] Watson, G. S. (1958). On Chi-Square goodness-of-fit tests for continuous distributions. *Journal of the Royal Statistical Society B* **20** 44-72.

[53] White, E. P., Enquist, B. J. and Green, J. L. (2008). On estimating the exponent of power-law frequency distributions. *Ecology* **89**, 905-912.

[54] Zaninetti, L. and Ferraro, M. (2008). On the truncated Pareto distribution with applications. *Central European Journal of Physics* **6**, 1-6.

[55] Zhang, C. H. (1990). Fourier methods for estimating mixing density and distributions. *The Annals of Statistics* **18**, 806-831.

[56] Zhang, S. and Karunamuni R. J. (2000). Boundary bias correction for nonparametric deconvolution. *Annals of the Institute of Statistal Mathematics* **5**, 612-629.

Appendices

Appendix A: The derivatives of the log-likelihood function in (4.9).

The following defnitions are useful in both Appendices A and B below:

$$
\begin{aligned}
E(y_j, x) &= \exp\left\{-\frac{(y_j - x)^2}{2\sigma^2}\right\} \\
K(x) &= \Phi\left(\frac{u - x}{\sigma}\right) - \Phi\left(\frac{l - x}{\sigma}\right) \\
I_0(y_j) &= \int_L^U x^{-a-1} E(y_j, x)\, dx \\
I_1(y_j) &= \int_L^U x^{-a-1} (\log x)\, E(y_j, x)\, dx \\
I_2(y_j) &= \int_L^U x^{-a-1} (\log x)^2\, E(y_j, x)\, dx \\
I_3(y_j) &= \int_L^U x^{-a-1} (y_j - x)^2\, E(y_j, x)\, dx \\
I_4(y_j) &= \int_L^U x^{-a-1} (y_j - x)^2 (\log x)\, E(y_j, x)\, dx \\
I_5(y_j) &= \int_L^U x^{-a-1} (y_j - x)^4\, E(y_j, x)\, dx \\
I_6 &= \int_L^U x^{-a-1} K(x)\, dx
\end{aligned}
$$

The first partial derivatives of the log-likelihood function $\mathcal{L}$ in (4.9) are:

$$
\begin{aligned}
\frac{\partial \mathcal{L}}{\partial L} &= n\left(\frac{aL^{-a-1}}{L^{-a} - U^{-a}}\right) - L^{-a-1} \sum_{j=1}^{n} \frac{E(y_j, L)}{I_0(y_i)} \\
\frac{\partial \mathcal{L}}{\partial U} &= U^{-a-1} \sum_{j=1}^{n} \frac{E(y_j, U)}{I_0(y_j)} - n\left(\frac{aU^{-a-1}}{L^{-a} - U^{-a}}\right) \\
\frac{\partial \mathcal{L}}{\partial a} &= \frac{n}{a} - \frac{n}{L^{-a} - U^{-a}}\left[U^{-a}(\log U) - L^{-a}(\log L)\right] - \sum_{j=1}^{n} \frac{I_1(y_i)}{I_0(y_j)} \\
\frac{\partial \mathcal{L}}{\partial \sigma} &= -\frac{n}{\sigma} + \frac{1}{\sigma^3} \sum_{j=1}^{n} \frac{I_3(y_j)}{I_0(y_j)}
\end{aligned}
$$

The second partial derivatives are

$$
\begin{aligned}
F_{11} = \frac{\partial^2 \mathcal{L}}{\partial L^2} &= n\left[\frac{(a^2+a)\,L^{-a-2}U^{-a}}{(L^{-a}-U^{-a})^2}\right] - n\left[\frac{aL^{-2a-2}}{(L^{-a}-U^{-a})^2}\right] \\
&+ (a+1)\,L^{-a-2}\sum_{j=1}^{n}\frac{E(y_j,L)}{I_0(y_j)} - \frac{L^{-a-1}}{\sigma^2}\sum_{j=1}^{n}\frac{(y_j-L)\,E(y_j,L)}{I_0(y_j)} \\
&- L^{-2a-2}\sum_{j=1}^{n}\left[\frac{E(y_j,L)}{I_0(y_j)}\right]^2 \\
F_{22} = \frac{\partial^2 \mathcal{L}}{\partial U^2} &= n\left[\frac{(a^2+a)\,U^{-a-2}L^{-a}}{(L^{-a}-U^{-a})^2}\right] - n\left[\frac{aU^{-2a-2}}{(L^{-a}-U^{-a})^2}\right] \\
&- (a+1)\,U^{-a-2}\sum_{j=1}^{n}\frac{E(y_j,U)}{I_0(y_j)} + \frac{U^{-a-1}}{\sigma^2}\sum_{j=1}^{n}\frac{(y_j-U)\,E(y_j,U)}{I_0(y_j)} \\
&- U^{-2a-2}\sum_{j=1}^{n}\left[\frac{E(y_j,U)}{I_0(y_j)}\right]^2 \\
F_{33} = \frac{\partial^2 \mathcal{L}}{\partial a^2} &= -\frac{n}{a^2} + \frac{nL^{-a}U^{-a}}{(L^{-a}-U^{-a})^2}\,(\log L - \log U)^2 \\
&+ \sum_{j=1}^{n}\left\{\frac{I_2(y_j)}{I_0(y_j)} - \left[\frac{I_1(y_j)}{I_0(y_j)}\right]^2\right\} \\
F_{44} = \frac{\partial^2 \mathcal{L}}{\partial \sigma^2} &= \frac{n}{\sigma^2} - \frac{3}{\sigma^4}\sum_{j=1}^{n}\frac{I_3(y_j)}{I_0(y_j)} + \frac{1}{\sigma^6}\sum_{j=1}^{n}\left\{\frac{I_5(y_j)}{I_0(y_j)} - \left[\frac{I_3(y_j)}{I_0(y_j)}\right]^2\right\} \\
F_{12} = \frac{\partial^2 \mathcal{L}}{\partial L \partial U} &= -n\left[\frac{a^2L^{-a-1}U^{-a-1}}{(L^{-a}-U^{-a})^2}\right] \\
&+ L^{-a-1}U^{-a-1}\sum_{j=1}^{n}\frac{E(y_j,L)E(y_j,U)}{\left[I_0(y_j)\right]^2} \\
F_{14} = \frac{\partial^2 \mathcal{L}}{\partial L \partial \sigma} &= -\frac{L^{-a-1}}{\sigma^3}\sum_{j=1}^{n}\left[\frac{E(y_j,L)\,(y_j-L)^2}{I_0(y_j} - \frac{E(y_j,L)I_3(y_j)}{\left[I_0(y_j)\right]^2}\right] \\
F_{24} = \frac{\partial^2 \mathcal{L}}{\partial U \partial \sigma} &= \frac{U^{-a-1}}{\sigma^3}\sum_{j=1}^{n}\left\{\frac{E(y_j,U)\,(y_j-U)^2}{I_0(y_j)} - \frac{E(y_j,U)\cdot I_3(y_i)}{\left[I_0(y_j)\right]^2}\right\} \\
F_{34} = \frac{\partial^2 \mathcal{L}}{\partial a \partial \sigma} &= \frac{1}{\sigma^3}\sum_{j=1}^{n}\left\{\frac{I_1(y_j)I_3(y_i)}{\left[I_0(y_j)\right]^2} - \frac{I_4(y_i)}{I_0(y_j)}\right\}
\end{aligned}
$$

$$
\begin{aligned}
F_{31} = \frac{\partial^2 \mathcal{L}}{\partial a \partial L} &= \frac{nL^{-a-1}\left(1 - a\log L\right)}{L^{-a} - U^{-a}} - \frac{naL^{-a-1}\left(U^{-a}\log U - L^{-a}\log L\right)}{\left[L^{-a} - U^{-a}\right]^2} \\
&- L^{-a-1}\left\{\sum_{j=1}^{n}\frac{E(y_j, L)I_1(y_j)}{\left[I_0(y_j)\right]^2} - \sum_{j=1}^{n}\log L\frac{E(y_j, L)}{I_0(y_j)}\right\} \\
F_{32} = \frac{\partial^2 \mathcal{L}}{\partial a \partial U} &= U^{-a-1}\left\{\sum_{j=1}^{n}\frac{E(y_j, U)I_3(y_j)}{\left[I_0(y_j)\right]^2} - \log U\sum_{j=1}^{n}\frac{E(y_j, U)}{I_0(y_j)}\right\} \\
&+ \frac{nU^{-a-1}\left(1 - a\log U\right)}{L^{-a} - U^{-a}} + \frac{naU^{-a-1}\left(U^{-a}\log U - L^{-a}\log L\right)}{\left[L^{-a} - U^{-a}\right]^2}
\end{aligned}
$$

Note that the second partial derivatives are used in the following information matrix

$$
\begin{pmatrix}
F_{11} & F_{12} & F_{13} & F_{14} \\
F_{21} & F_{22} & F_{23} & F_{24} \\
F_{31} & F_{32} & F_{33} & F_{34} \\
F_{41} & F_{42} & F_{43} & F_{44}
\end{pmatrix}^{-1}
$$

and $F_{12} = F_{21}$; $F_{13} = F_{31}$; $F_{14} = F_{41}$; $F_{42} = F_{24}$; $F_{32} = F_{23}$; $F_{34} = F_{43}$.

Appendix B: The derivatives of the log-likelihood function $\mathcal{L}_r$ in (4.17)

The first partial derivatives of the log-likelihood function $\mathcal{L}_r$ are

$$
\begin{aligned}
\frac{\partial \mathcal{L}_r}{\partial L} &= L^{-a-1}\left[\frac{nK(L)}{I_6} - \sum_{i=1}^{n}\frac{E(y_j, L)}{I_0(y_j)}\right] \\
\frac{\partial \mathcal{L}_r}{\partial U} &= U^{-a-1}\left[\sum_{j=1}^{n}\frac{E(y_j, U)}{I_0(y_i)} - \frac{nK(U)}{I_6}\right] \\
\frac{\partial \mathcal{L}_r}{\partial a} &= \frac{n\int_L^U x^{-a-1}\left(\log x\right)K(x)\,dx}{I_6} - \sum_{j=1}^{n}\frac{I_1(y_j)}{I_0(y_j)} \\
\frac{\partial \mathcal{L}_r}{\partial \sigma} &= -\frac{n}{\sigma} + \frac{1}{\sigma^3}\sum_{j=1}^{n}\frac{I_3(y_j)}{I_0(y_j)} - \frac{n\int_L^U x^{-a-1}K_\sigma(x)\,dx}{I_6}
\end{aligned}
$$

The second partial derivatives are

$$\begin{aligned}
\frac{\partial^2 \mathcal{L}_r}{\partial L^2} &= (-a-1)L^{-a-2}\left[\frac{nK(L)}{I_6} - \sum_{j=1}^{n}\frac{E(y_j,L)}{I_0(y_j)}\right] \\
&+ nL^{-a-1}\left[\frac{K_L(L)}{I_6} + \frac{L^{-a-1}\left[K(L)\right]^2}{\left[I_6\right]^2}\right] \\
&- L^{-a-1}\sum_{j=1}^{n}\left\{\frac{\sigma^{-2}E(y_j,L)(y_j-L)}{I_0(y_j)} + \frac{L^{-a-1}\left[E(y_j,L)\right]^2}{\left[I_0(y_j)\right]^2}\right\} \\
\frac{\partial^2 \mathcal{L}_r}{\partial U^2} &= (-a-1)U^{-a-2}\left[\sum_{j=1}^{n}\frac{E(y_j,U)}{I_0(y_j)} - \frac{nK(U)}{I_6}\right] \\
&- nU^{-a-1}\left[\frac{K_U(U)}{I_6} - \frac{U^{-a-1}\left[K(U)\right]^2}{\left[I_6\right]^2}\right] \\
&+ U^{-a-1}\sum_{j=1}^{n}\left\{\frac{\sigma^{-2}E(y_j,U)(y_j-U)}{I_0(y_j)} - \frac{U^{-a-1}\left[E(y_j,U)\right]^2}{\left[I_6\right]^2}\right\} \\
\frac{\partial^2 \mathcal{L}_r}{\partial a^2} &= \sum_{j=1}^{n}\left\{\frac{I_2(y_j)}{I_0(y_j)} + \frac{\left[I_1(y_j)\right]^2}{\left[I_0(y_j)\right]^2}\right\} \\
&- n\left\{\frac{\int_L^U x^{-a-1}\left(\log x\right)^2 K(x)\,dx}{I_6} + \frac{\left[\int_L^U x^{-a-1}\left(\log x\right)K(x)\,dx\right]^2}{\left[I_6\right]^2}\right\} \\
\frac{\partial^2 \mathcal{L}_r}{\partial \sigma^2} &= \frac{n}{\sigma^2} - \frac{3}{\sigma^4}\sum_{j=1}^{n}\frac{I_3(y_j)}{I_0(y_j)} + \frac{1}{\sigma^6}\sum_{j=1}^{n}\left\{\frac{I_5(y_j)}{I_0(y_j)} - \frac{\left[I_3(y_j)\right]^2}{\left[I_0(y_j)\right]^2}\right\} \\
&- \left\{\frac{n\int_L^U x^{-a-1}K_{\sigma^2}(x)\,dx}{I_6} + \frac{n\left[\int_L^U x^{-a-1}K_{\sigma}(x)\,dx\right]^2}{\left[I_6\right]^2}\right\} \\
\frac{\partial^2 \mathcal{L}_r}{\partial L \partial U} &= L^{-a-1}U^{-a-1}\left\{\sum_{j=1}^{n}\frac{E(y_j,L)E(y_j,U)}{\left[I_0(y_j)\right]^2} - \frac{nK(L)K(U)}{\left[I_6\right]^2}\right\} \\
\frac{\partial^2 \mathcal{L}_r}{\partial a \partial U} &= U^{-a-1}\left(\log U\right)\left[\sum_{j=1}^{n}\frac{E(y_j,U)}{I_0(y_j)} - \frac{nK(U)}{I_6}\right] + U^{-a-1} \\
&\left\{\sum_{j=1}^{n}\frac{E(y_j,U)I_1(y_j)}{\left[I_0(y_j)\right]^2} - \frac{nK(U)\int_L^U x^{-a-1}\left(\log x\right)^2 K(x)\,dx}{\left[I_6\right]^2}\right\}
\end{aligned}$$

$$
\begin{aligned}
\frac{\partial^2 \mathcal{L}_r}{\partial U \partial \sigma} &= -U^{-a-1}\left\{\frac{nK_\sigma(U)}{I_6} - \frac{nK(U)\int_L^U x^{-a-1}K_\sigma(x)\,dx}{[I_6]^2}\right\} + \\
&\quad \frac{U^{-a-1}}{\sigma^3}\sum_{j=1}^{n}\left\{\frac{(y_j-U)^2 E(y_j,U)}{I_0(y_j)} - \frac{I_3(y_j)E(y_j,U)}{[I_0(y_j)]^2}\right\} \\
\frac{\partial^2 \mathcal{L}_r}{\partial a \partial L} &= L^{-a-1}(\log L)\left[\frac{nK(L)}{I_6} - \sum_{j=1}^{n}\frac{E(y_j,L)}{I_0(y_j)}\right] + \\
&\quad L^{-a-1}\left\{\frac{nK(L)\int_L^U x^{-a-1}(\log x)^2 K(x)\,dx}{[I_6]^2} - \sum_{j=1}^{n}\frac{E(y_j,L)I_1(y_j)}{[I_0(y_j)]^2}\right\} \\
\frac{\partial^2 \mathcal{L}_r}{\partial L \partial \sigma} &= L^{-a-1}\left\{\frac{nK_\sigma(L)}{I_6} - \frac{nK(L)\int_L^U x^{-a-1}K_\sigma(x)\,dx}{[I_6]^2}\right\} - \\
&\quad \frac{L^{-a-1}}{\sigma^3}\sum_{j=1}^{n}\left\{\frac{(y_j-L)^2 E(y_j,L)}{I_0(y_j)} - \frac{I_3(i)E(y_j,L)}{[I_0(y_j)]^2}\right\} \\
\frac{\partial^2 \mathcal{L}_r}{\partial \sigma \partial a} &= \frac{n\int_L^U x^{-a-1}(\log x)^2 K(x)\,dx}{I_6} - \frac{n\left[\int_L^U x^{-a-1}K(x)\,dx\right]^2}{[I_6]^2} + \\
&\quad \frac{1}{\sigma^3}\sum_{j=1}^{n}\left\{\frac{I_4(y_j)}{I_0(y_j)} - \frac{I_3(y_j)I_1(y_j)}{[I_0(y_j)]^2}\right\}
\end{aligned}
$$

where the derivatives of $K(x)$ with respect to the parameters: σ, L and U are:

$$
\begin{aligned}
K_\sigma(x) &= \frac{1}{\sigma^4\sqrt{2\pi}}\left\{\int_{-\infty}^{u} E(t,x)(t-x)^2\,dt - \int_{-\infty}^{l} E(t,x)(t-x)^2\,dt\right\} - \frac{1}{\sigma}K(x) \\
K_L(L) &= \frac{1}{\sigma^3\sqrt{2\pi}}\left\{\int_{-\infty}^{u} E(t,L)(t-L)\,dt - \int_{-\infty}^{l} E(t,L)(t-L)\,dt\right\} \\
K_U(U) &= \frac{1}{\sigma^3\sqrt{2\pi}}\left\{\int_{-\infty}^{u} E(t,U)(t-U)\,dt - \int_{-\infty}^{l} E(t,U)(t-U)\,dt\right\} \\
K_{\sigma^2}(x) &= \frac{2}{\sigma}K(x) + \frac{1}{\sigma^7\sqrt{2\pi}}\left\{\int_{-\infty}^{u} E(t,x)(t-x)^4\,dt - \int_{-\infty}^{l} E(t,x)(t-x)^4\,dt\right\} \\
&- \frac{5}{\sigma^5\sqrt{2\pi}}\left\{\int_{-\infty}^{u} E(t,x)(t-x)^2\,dt - \int_{-\infty}^{l} E(t,x)(t-x)^2\,dt\right\}
\end{aligned}
$$

Appendix C: MATLAB programs

Algorithm 1: Simulations

1. For a given model $Y = X + \epsilon$, where the variable X follows a Pareto PDF given in (4.1) and $\epsilon \sim N(0, \sigma^2)$.
2. Let $z = F_X(x)$, then $x = (L^{-a} - z(L^{-a} - U^{-a}))^{-1/a}$, refer to (4.2). Given the parameter vector values $[L, U, a]$, desired sample size n and $z \sim U(0, 1)$; x in (2) above returns the un-contaminated simulated data.
3. Adding the Gaussian error with mean zero and variance σ^2 to x. The error-contaminated data $Y = x + \epsilon$ is produced.

The Matlab code of Algorithm 1


```
function [Y] = SimuPareto(n,params)
%************************************************************************
% inputs: desired sample size n and the parameter vector values [L, U, asigma]
L = params(1);          % lower limit
U = params(2);          % upper limit
a = params(3);          % exponent or power indices
sigma = params(4);      % error standard deviation
%************************************************************************
z = rand(n,1);      % vector of uniformly distributed pseudo-random numbers.
x =((L.^(-a) - z.*(L.^(-a) - U.^(-a))).^(-1./a));
e = sigma*randn(n,1);
Y = x + e;
%************************************************************************
% The histograms
subplot(2,1,1),hist(x)
title('UN-CONTAMINATED DATA')
subplot(2,1,2),hist(Y)
title('ERROR-CONTAMINATED DATA')
return
```

Algorithm 2: MLEs of the log-likelihood function given in (4.9)

The Matlab codes below minimises the negative log-likelihood function with respect to unknown parameters of the convolved PDF. It accepts initial estimates (params) and returns the estimated parameter vector (paramsEst) obtained by optimization (minimization or maximization) procedure.

The Matlab code of Algorithm 2

```
function [paramsEst,Funval,exitflag,output] = MAXlikelihood(data,params)
%*************************************************************************

% fminserach = minimize the scalar function loglikfnct, starting at initial (params).
% Funval is the value of the function loglikfnct at the solution paramsEst.
% exitflag = describe the exit condition of fminsearch:
%            1     fminsearch converge to a solution paramsEst
%            0     Maximum number of function evaluations or iterations was reached
%           -1     Algorithm was terminated by the output function
% output = returns structure output that contains information about the optimization
%*************************************************************************

[paramsEst,Funval,exitflag,output] = fminsearch(@loglikfnct,params,[ ],data);
return
%*************************************************************************

function [loglikhod] = loglikfnct(params,data)

% loglikfnct(params,data) function returns the negative log-likelihood value
%*************************************************************************
z = sort(data);              %       assigning data to variable z
n = length(z);               %       sample size
%*************************************************************************
% initial estimates

L = params(1);               %       Lower limit
```

```
U = params(2);              %       Upper limit
a = params(3);              %       exponent
sigma = params(4);          %        error standard deviation
%*************************************************************************
% Terminate if the conditions are not met

if (a < 0||L < 0||U < L|| sigma < 0)
loglikhod = 1.0e+20;
return
end
%*************************************************************************
% First find the limits of the function only where the integrand is non zero.

re = 1.e-8;         %       relative error
tol = 1.e-8;

for i = 1: n
x = z(i);
y = integrand2(x,z(i),a,sigma);

   while y > re
x = x - 0.5*sigma;
y = integrand2(x,z(i),a,sigma);
end
L1 = x;                 % lower limits
%*************************************************************************
% Note that lower and upper limits must be in the interval [L, U].
%*************************************************************************

if (L1 < L)

       Llim = L;
else
Llim = L1;
```

```
end
%*************************************************************************
x = z(i);
y = integrand2(x,z(i),a,sigma);
while y > re
x = x + 0.5*sigma;
y = integrand2(x, z(i), a,sigma);
end
U1 = x;          % upper limits
%*************************************************************************
if (U1 > U)
Ulim = U;
else
Ulim = U1; end
%*************************************************************************
D(i) = quadl(@integrand2,Llim,Ulim,tol,[],z(i),a,sigma);
end
%*************************************************************************
% function body

B = n*log(a) - n*log(L^(-a) - U^(-a));
A = -n*log(sigma) - (n/2)*log(2*pi);
C = sum(log(D));
loglike = (A + B) + C;
loglikhod = -loglike;          % negative log-likelihood
return
%*************************************************************************

function y = integrand2(x,z,a,sigma)
% integral part of the log-likelihood function.

y = (x.^(-a - 1)).*exp((-1/2).*((z - x)./sigma).^2);
return
```

Algorithm 3: Information matrix and standard errors of the estimates

The Matlab codes below calculates information matrix and standard errors of the estimated parameter vector $[L, U, a, \text{sigma}]$ obtained by algorithm 2.

The Matlab code of Algorithm 3a

```
function [covmatrix,stderrors] = infomatrix(data,paramsEst)

% function [covmatrix,stderrors] = infomatrix(data,paramsEst)
% input: data and estimated parameters
% output: Covariance matrix and standard errors of the parameter estimates
%************************************************************************

L = paramsEst(1);
U = paramsEst(2);
a = paramsEst(3);
sigma = paramsEst(4);

z = sort(data);
n = length(data);
%************************************************************************
for i = 1:n
alpf(i) = quad(@integrand2,L,U, [],[],z(i),a,sigma);

pbl = (a.^2*L.^(-a-2)*U.^(-a) - a*L.^(-2*a-2) + a*L.^(-a-2)*U.^(-a));
bc = L.^(-a) - U.^(-a);
pL = n*(pbl/bc.^2);

palpL(i) = -L.^(-a-2).*(-a-1).*exp(-1/2*(z(i)-L).^2 ./sigma.^2) - ...
L.^(-a-1).*((z(i)-L)./sigma.^2).*exp(-1/2*(z(i)-L).^2./sigma.^2);

alpL(i) = (-L.^(-a-1).*exp(-1/2*(z(i)-L).^2./sigma.^2)).^2;

pL2 = (palpL.*alpf - alpL)./(alpf.^2);
%************************************************************************
pbu = (a^2*U^(-a-2)*L^(-a) - a*U^(-2*a-2) + a*U^(-a-2)*L^(-a));
```

```
pU = n*(pbu/bc^2);

palpU(i) = U^(-a-2).*(-a-1).*exp(-1/2*(z(i)-U).^2./sigma.^2) + ...
U^(-a-1).*((z(i)-U)./sigma.^2).*exp(-1/2*(z(i)-U).^2./sigma.^2);

alpU(i) = (U^(-a-1).*exp(-1/2*(z(i)-U).^2./sigma.^2)).^2;

pU2 = (palpU.*alpf - alpU)./(alpf.^2);
%*************************************************************************
pba = (L^(-a)*U^(-a))*(log(L) - log(U))^2;
pA = n*(-a^(-2) + pba/(bc^2));

alpa2(i) = quad(@apart21,L,U, [],[],z(i),a,sigma);

alpa1(i) = (quad(@apart1,L,U, [],[],z(i),a,sigma)).^2;

pA2 = (alpa2.*alpf - alpa1)./(alpf.^2);
%*************************************************************************
pS = n*sigma^(-2);

alps1(i) = quad(@sigpart11,L,U, [],[],z(i),a,sigma);

alps2(i) = quad(@sigpart21,L,U, [],[],z(i),a,sigma);

alps3(i) = (quad(@sigpart31,L,U, [],[],z(i),a,sigma)).^2;

pS2 = ((alps1 + alps2).*alpf - alps3)./(alpf.^2);
%*************************************************************************
pblu = -a^2*L^(-a-1)*U^(-a-1);

pLU = n*(pblu/bc^2);

palpLU(i) = L^(-a-1).*U^(-a-1).*exp(-1/2.*((z(i)-L).^2./sigma.^2 + ...
(z(i)- U).^2./sigma.^2));
```

```
pLU2 = palpLU./(alpf. ^ 2);
%*************************************************************************
pbla = L ^ (-2*a-1) + (L ^ (-a-1)*U ^ (-a))*((a*log(L) - a*log(U)) - 1);
pLA = n*(pbla/bc ^ 2);

palpLA1(i) = L ^ (-a-1).*log(L).*exp(-1/2.*(z(i)-L). ^ 2./sigma. ^ 2);

alpLA2(i) = -L ^ (-a-1).*exp(-1/2*(z(i)-L). ^ 2./sigma. ^ 2);

alpLA3(i) = quad(@apart1,L,U, [],[],z(i),a,sigma);

pLA2 = ((palpLA1.*alpf) - (alpLA2.*alpLA3))./(alpf. ^ 2);
%*************************************************************************
palpLS1(i) = -L ^ (-a-1).*((z(i) -L). ^ 2./sigma ^ 3).*exp(-1/2.*(z(i)-L). ^ 2./sigma. ^ 2);

alpLS2(i) = -L ^ (-a-1).*exp(-1/2*(z(i)-L). ^ 2./sigma. ^ 2);

alpLS3(i) = quad(@sigpart31,L,U, [],[],z(i),a,sigma);

pLS2 = ((palpLS1.*alpf) - (alpLS2.*alpLS3))./(alpf. ^ 2);
%*************************************************************************
pbua = U ^ (-2*a-1) + (U ^ (-a-1)*L ^ (-a))*(a*log(U) - a*log(L) - 1);
pUA = n*(pbua/bc ^ 2);

palpUA1(i) = -U ^ (-a-1).*log(U).*exp(-1/2.*(z(i)-U). ^ 2./sigma. ^ 2);

alpUA2(i) = U ^ (-a-1).*exp(-1/2*(z(i)-U). ^ 2./sigma. ^ 2);

alpUA3(i) = quad(@apart1,L,U, [],[],z(i),a,sigma);

pUA2 = ((palpUA1.*alpf) - (alpUA2.*alpUA3))./(alpf. ^ 2);
%*************************************************************************
palpUS1(i) = U ^ (-a-1).*((z(i) -U). ^ 2./sigma ^ 3).*exp(-1/2.*(z(i)-U). ^ 2./sigma. ^ 2);

alpUS2(i) = U ^ (-a-1).*exp(-1/2*(z(i)-U). ^ 2./sigma. ^ 2);
```

```
alpUS3(i) = quad(@sigpart31,L,U, [],[],z(i),a,sigma);

pUS2 = ((palpUS1.*alpf) - (alpUS2.*alpUS3))./(alpf.^2);
%*************************************************************************
alpAS1(i) = quad(@sigpart41,L,U, [],[],z(i),a,sigma);

alpAS2(i) = quad(@sigpart31,L,U, [],[],z(i),a,sigma);

alpAS3(i) = quad(@apart1,L,U, [],[],z(i),a,sigma);

pAS2 = ((alpAS1.*alpf) - (alpAS2.*alpAS3))./(alpf.^2);
%*************************************************************************
end
xL = (-1).*(pL + sum(pL2));
xU = (-1).*(pU + sum(pU2));
xA = (-1).*(pA + sum(pA2));
xS = (-1).*(pS + sum(pS2));
xLU = (-1).*(pLU + sum(pLU2));
xLA = (-1).*(pLA + sum(pLA2));
xLS = (-1).*sum(pLS2);
xUa = (-1).*(pUA + sum(pUA2));
xUs = (-1).*sum(pUS2);
xAS = (-1).*sum(pAS2);

covmatrix = inv([xL xLU xLA xLS;xLU xU xUa xUs;xLA xUa xA xAS;xLS xUs
xAS xS]);

stderrors = sqrt(diag(invmatrix));
return
%*************************************************************************

function y = integrand2(x,z,a,sigma)
y = (x.^(-a - 1)).*exp((-1/2).*((z - x)./sigma).^2);
return
```

```
%*************************************************************************

function y1 = apart1(x,z,a,sigma)
y1 = (-x. ^(-a - 1)).*(log(x)).*exp((-1/2).*((z -x)./sigma). ^2);
return
%*************************************************************************

function y3 = apart21(x,z,a,sigma)
y3 = (x. ^(-a - 1)).*((log(x)). ^2).*exp((-1/2).*((z - x)./sigma). ^2);
return
%*************************************************************************
function y4 = sigpart11(x,z,a,sigma)
y4 = (-3.*x. ^(-a - 1).*(((z-x). ^2)./sigma. ^4)...
.*exp((-1/2).*((z - x)./sigma). ^2));
return
%*************************************************************************

function y5 = sigpart21(x,z,a,sigma)
y5 = x. ^(-a - 1).*((z-x). ^4)./(sigma. ^6).*(exp((-1/2).*((z -x)./sigma). ^2));
return
%*************************************************************************

function y6 = sigpart31(x,z,a,sigma)
y6 = (x. ^(-a - 1)).*(((z-x). ^2)./(sigma). ^3).*exp((-1/2).*((z -x)./sigma). ^2);
return
%*************************************************************************

function y7 = sigpart41(x,z,a,sigma)
y7 = (-x. ^(-a - 1)).*(log(x)).*(((z-x). ^2)./(sigma). ^3).*exp((-1/2).*((z -x)./sigma).
^2);
return
```


The Matlab code of Algorithm 3b

```
function [sehatJack] = jackfnct(data,paramsEst)
```

```
%*************************************************************************

% function [sehat] = jack(data,paramsEst) estimates the standard errors
% of the paramsEst [L,U,a,sigma]
% using jackknife method
% input - data and estimated parameter vector of [L,U,a,sigma]
% Outputs: Standard errros of the estimates

% Created by Lwando Kondlo
% March 2008
%
% References:
% Efron, B. Bootstrap Methods. Another Look at the Jackknife.
% The Annals of Statistics, Vol. 7, pp. 1-26, 1979.
%*************************************************************************
z = sort(data);
n = length(z);

reps = zeros(n,1); % initialize jackknife replications
reps2 = zeros(n,1);
reps3 = zeros(n,1);
reps4 = zeros(n,1);
%*************************************************************************

for i = 1 : n
zt = z; % store temporary vector
zt(i) = []; % leave i-th point out (jackknife sample)
Ls = MAXlikelihood(zt,paramsEst);
reps(i) = Ls(1,1); %jackknife replicates of paramsEst
reps2(i) = Ls(1,2);
reps3(i) = Ls(1,3);
reps4(i) = Ls(1,4);
end
%*************************************************************************
% jackknife estimate
```

```
mureps = mean(reps);
mureps2 = mean(reps2);
mureps3 = mean(reps3);
mureps4 = mean(reps4);
%*************************************************************************
% get the estimate of the standard error

sehat1 = sqrt((n-1)/n *sum((reps - mureps). ^ 2));
sehat2 = sqrt((n-1)/n *sum((reps2 - mureps2). ^ 2));
sehat3 = sqrt((n-1)/n *sum((reps3 - mureps3). ^ 2));
sehat4 = sqrt((n-1)/n *sum((reps4 - mureps4). ^ 2));
%*************************************************************************

sehatJack = [sehat1 sehat2 sehat3 sehat4];
return
```


Algorithm 4: Probability-probability plots

The Matlab code below plots the Probability-probability plots of the convolved CDF given the estimated parameter vector $[L, U, a, \text{sigma}]$ obtained by algorithm 2 and dataset.

The Matlab code of Algorithm 4

```
function ppplots(paramsEst,data)

% ppplots(paramsEst,data) returns the Probability-probability plots

z = sort(data);
n = length(z);

L = paramsEst(1);
U = paramsEst(2);
a = paramsEst(3);
```

```
sigma = paramsEst(4);
%*************************************************************************

for i = 1 : n
intepart(i) = quad(@CDFfuntrail,L,U,[],[],z(i),a,sigma);
constant = a/(L^(-a)- U^(-a));
emp(i) = (i./n);
end
%*************************************************************************
k = 0:0.5:1;
G2 = emp';
G1 = (constant.*intepart)';
plot(G2,G1,'+',k,k,'r','LineWidth',2)
legend('probabilities','straight line');
xlabel('Empirical CDFs')
ylabel('Theoretical CDFs')
return
%*************************************************************************

function y = CDFfuntrail(P,z,a,sigma)

% integral part of the CDF function
%*************************************************************************
N = length(P);
for j = 1:N;
x = P(j);
k = (z - x)./(sqrt(2).*sigma);
CDFnorm = 0.5.*(1 + erf(k));
v = (x.^(-a-1)).*CDFnorm;
u(j) = x;
y(j) = v;
end
return
```

Algorithm 5: convolved CDF graph and K-S test statistic

The Matlab code below calculates the K-S test statistic and plots the graph of the convolved CDF, given the estimated parameter vector $[L, U, a, \text{sigma}]$ obtained by algorithm 2 and dataset.

The Matlab code of Algorithm 5

```
function [F1] = CDFParetoNorm(paramsEst,data);
%************************************************************************
z = sort(data(:));
n = length(z);
%************************************************************************
% estimates
L = paramsEst(1);
U = paramsEst(2);
a = paramsEst(3);
sigma = paramsEst(4);
%************************************************************************
% Tolerance parameter for quadrature algorithm (quad or lobbato-quadl)
tolr = 1.e-6;
%************************************************************************
% for loop
for i = 1 : n
D(i) =(quad(@CDFfuntrail,L,U,tolr,[],z(i),a,sigma));
jj = find(z <= z(i));
empir2(i) = length (jj)./n;
end
%************************************************************************
constant = a/(L^(-a)- U^(-a));
G = (constant.*D);
%************************************************************************
F = G';
emp2 = empir2';
%************************************************************************
plot(z,F,'b-',z,emp2,'-r','LineWidth',2)
```

```
legend('Theoretical CDF','Empirical CDF');
xlabel('observe random variables')
ylabel('Probabilities')
%**************************************************************************
F1 = [F emp2];
kol = max(abs(emp2 - F))        % kolmogorov-smirnov value
return
```

Algorithm 6: Bootstrapping critical values of the K-S test statistic

The Matlab code below returns the critical values of the K-S test statistic given the estimated parameter vector $[L, U, a, \text{sigma}]$ obtained by algorithm 2, dataset and number of bootstrap replicates, nboot.

The Matlab code of Algorithm 6


```
function [KSvaleus paramsEst1] = KSbootscriticalV(data,nboot,paramsEst)

z = sort(data)'; % input data
n = length(z);
%**************************************************************************
for i=1:nboot,
a = ceil(n*rand(n,1));
zt = z(a);

paramsEst0 = Maxlikelihood(zt,paramsEst);
ztt = CDFParetoNorm(paramsEst0,zt);

D = max(abs(ztt(:,2) - ztt(:,1)));
Dx = abs(ztt(:,2) - ztt(:,1));
jj = find(Dx == D);
xx = length(jj);
%**************************************************************************

parL(i) = paramsEst0(:,1);
```

```
parU(i) = paramsEst0(:,2);
para(i) = paramsEst0(:,3);
pars(i) = paramsEst0(:,4);
end;
%*************************************************************************

KSvaleus = sort(mx11)';
paramsEstL = sort(parL)';
paramsEstU = sort(parU)';
paramsEsta = sort(para)';
paramsEsts = sort(pars)';
paramsEst1 = [paramsEstL paramsEstU paramsEsta paramsEsts];
return
```