An Empirical Comparison Among Four Estimation Methods for The Laplace Distribution And Its Application In Statistical Inference

By

XIAO Yuying,Christina
(1830005037)

A Final Year Project thesis (STAT1001)
submitted in partial fulfillment of the requirements
for the degree of

Bachelor of Science (Honours)
in Statistics

at

BNU-HKBU
UNITED INTERNATIONAL COLLEGE

December,10

DECLARATION

I hereby declare that all the work done in this Project is of my independent effort. I also certify that I have never submitted the idea and product of this Project for academic or employment credits.

____

XIAO Yuying,Christina
(1830005037)

Date:______________________

An Empirical Comparison Among Four Estimation Methods for The Laplace Distribution And Its Application In Statistical Inference

XIAO Yuying, Christina

Science and Technology Division

Abstract

This thesis is applying on comparing four different estimations (The Moment, The Maximum Likelihood, The Fisher Minimum Chi-square method with equiprobable grouping, and the RP Minimum Chi-square) for constructing the chi-square test for goodness of fit for the Laplace distribution.The programme required MATLAB codes to calculate and simulate the results, then analyzed the output. The purpose of this thesis is to carry out extensive simulation (Monte Carlo) study to prove which of the four estimators can give better accuracy by using the Root Mean Square Error (RMSE).Overall, in the last step of project, I also did a simulation study on the performance of the statistic X_{n}^2 in controlling type I error rates for four alternative estimations.

Keywords: Laplace Distribution,The moment Estimation, The Maximum Likelihood Estimation,The Fisher Minimum Chi-square Estimation, Monte Carlo,Type I Error Rates, Root Mean Square Error

Preface

As I learned in my degrees in Statistic, I began to recognize the importance of applying the technology and knowledge for applying in realistic conditions, so I pursued to complete this thesis and project in earnest. After communicating with my supervisor Prof. Jiajuan LIANG, I founded interested in this project, since there is no other scholar research on this topic, we can prove the whole program in our own, the sense of achievement was attractive to me.

In fact, I am not so good at coding, therefore in the beginning, I felt a little bit worried and anxious. With the progressing of continuing learning and trying, I begin to feel confident and know how to apply the theories in practical problems. Proving that without the support of the supervisor and the encouragement of my classmates, I could hardly complete the project perfectly.

Furthermore, the process experienced plenty of issues and troubles, it is actually a tough test for the strong will and patience. Meanwhile, I learn some of the knowledge when solving the bugs which could never discovered in courses, therefore it is a brand new and variable experience for me.

This thesis explain the whole process of calculating and simulating for the Laplace distribution on the basis of Monte Carlo method for four estimation method, the MATLAB programming provide the tech support for the statistical output and inference. Therefore, the thesis is suitable for those who have the basic knowledge in Mathematics, Statistics or Internet Technology(including big data).

Nevertheless, for the individual who ‘ve been diagnosed with Statistics may also find it helpful to know the rough idea in a specific methods. I intended my research to help the reader, approach the field of Statistics from a new angle.

Acknowledgement

Four years of undergraduate study is coming to an end, this is a period of intense exploration. First of all, I'd like to thank Professor Jiajuan LIANG, who is my supervisor of the project. He not only provides me with advanced and superior research environment, but also directly teaches many years of research experience, makes my professional knowledge more stable, and research methods and skills to a higher level. Under the guidance of his patience, this thesis is gradually entering the frontier of discipline research. I am inspired by my supervisor's rigorous academic attitude, extensive knowledge system, positive and optimistic life creed, amiable teacher's demeanor and approachable manner of doing things. Meanwhile, I would like to extend my respect and sincere blessing to our tutor MS Guoqiu ZHANG, who gives me practical suggestions and encouragement during the past half year.

Secondly, I would like to thank my group members, especially Sihong Lin and Siqi Guo, both of them provided me the support of the software skills in MATLAB when I have difficulties in codes. They helped me to overcome the shortcomings of the theory and methods in the course of the thesis, which made me take many detours and finally complete the research. I would like to express my deep gratitude to all of them.

At the end, thanks to my parents and family, they are great, selfless, unrequited love forever in my heart, and will encourage me to love life, love of learning and researching, I will make an effort to live up to your expectations.

Contents

1 Introduction **1**

1.1 Mathematical Symbols 3

2 Methodology **4**

2.1 The Maximum Likelihood Estimation 4

2.2 The Moment Estimation 7

2.3 The Fisher Minimum Chi-square Estimation 8

2.4 The RP Minimum Chi-square Estimation 11

2.5 Calculation Of The Type I Error 11

3 Summary Results **13**

3.0.1 Matlab Output Of MLE Estimation 13

3.0.2 Matlab Output Of Moment Estimation 14

3.0.3 Matlab Output Of Fisher Minimum Chi-square Estimation 14

3.0.4 Matlab Output Of RP Minimum Chi-square Estimation 15

3.0.5 Matlab Output Of Type I Error 17

4 Conclusions **18**

5 Discussion **20**

A Codes in MATLAB **22**

A.1 Codes for generating random variables in Laplace distribution. 22

A.2 Codes for Maximum Likelihood Estimation 22

A.3 Codes for Moment Estimation 23

A.4 Codes for generating Fisher Equiprobable equations 24

A.5 Codes for generating Equiprobable Points and intervals. . . 25

A.6 Codes for calculating the RMSE of Fisher-minimum Chi-square Estimation 27

A.7 Codes for the selection of RP and the functions of RP equations. 28

A.8 Codes for calculating the RMSE of RP methods. 30

A.9 Codes for Calculating the Type I error of MLE and Moment Estimation 31

A.10 Codes for Calculating the Type I error of Fisher Minimum Chi-square and RP Estimations 36

Bibliography **44**

List of Tables

3.1 Matlab Output Of MLE Estimation,N = 1000 . . 13
3.2 Matlab Output Of Moment Estimation,N = 1000 14
3.3 Matlab Output Of Fisher Minimum Chi-square Estimation,m=4,N=1000 14
3.4 Matlab Output Of Fisher Minimum Chi-square Estimation,m=9,N=1000 14
3.5 Matlab Output Of RP Minimum Chi-square Estimation,n=50, N = 1000 15
3.6 Matlab Output Of RP Minimum Chi-square Estimation,n=100, N = 1000 15
3.7 Matlab Output Of RP Minimum Chi-square Estimation,n=200, N = 1000 15
3.8 Matlab Output Of RP Minimum Chi-square Estimation ,n=400,N = 1000 16
3.9 Matlab Output Of Type I Error,N=500 17

List of Figures

5.1 Probability density distribution of Laplace distribution. . . 21

Chapter 1

Introduction

In probability theory and statistics, the Laplace distribution is a continuous probability distribution named after Pierre-Simon Laplace.

It is also sometimes called the double exponential distribution, because it can be thought of as two exponential distributions (with an additional location parameter) spliced together back-to-back, although the term is also sometimes used to refer to the Gumbel distribution.[1]

A random variable has a distribution if its probability density function is

$$f(x_i|\mu, b) = \frac{1}{2b} exp(-\frac{|x_i - \mu|}{b})$$

$$= \left(\frac{1}{2b}\right) \begin{cases} exp(-\frac{\mu - x_i}{b}) & \text{if} \quad x < \mu \\ exp(-\frac{x_i - \mu}{b}) & \text{if} \quad x \geq \mu \end{cases}$$

Here, μ is a location parameter and b, which is sometimes referred to as the diversity, $b$is a scale parameter. The Laplace distribution is easy to integrate (if one distinguishes two symmetric cases) due to the use of the absolute value function. Its cumulative distribution function is as follows:

$$F(x) = \int_{-\infty}^{x} f(u)du$$

$$= \begin{cases} \frac{1}{2}\exp(\frac{x-\mu}{b}) & \text{if} \quad x < \mu \\ 1 - \frac{1}{2}\exp(-\frac{x-\mu}{b}) & \text{if} \quad x \geq \mu \end{cases}$$

There are various methods for estimating the mean and variance from a given i.i.d. (independent identically distributed) sample.The maximum likelihood estimation (MLE) is the most popular method for estimating parameters when the density function of a continuous distribution is known up to some unknown parameters such as the univariate normal $N(\mu, \sigma^2)$.

The moment estimation method for a population parameter σ is to use the sample central moments to estimate the population central moments.

The Fisher minimum Chi-square estimation method is based on minimizing the Pearson Fisher chi-square statisticthen combining the estimated equiprobable points for the location-scale Laplace distribution,which derivatives consist of the estimated equiprobable functions.[5]

The RP (representative point) minimum chi-square is to choose the classification interval as follows. Let$\{\mathrm{R}_i^0 : i =, ..., m\}$be a set of representative points (RPs) from $f_0(\cdot)$, which stands for the standard Laplace distribution, whose RPs (representative points) can be obtained by existing algorithm.

1.1 Mathematical Symbols

x: express the variables in laplace distribution.

f(x): express the probability density function (pdf) of the laplace distribution.

F(x): express the cumulative distribution function (cdf) of the laplace distribution.

μ: location parameter.

σ: scale parameter.

$\hat{\mu}$: express the estimated value of μ.

$\hat{\sigma}$: express the estimated value of σ.

R: a set of representative points from standard laplace distribution.

l: express the log-likelihood function of the laplace distribution.

L: express the maximun likelihood(ML) function of laplace distribution.

n: the sample size of x.

m: the number of equiprobable points.

a: the equiprobable points.

b: express the points of location-scale transformation.

N: express the times of simulation.

p: express the equiproble functions based on different parameters.

i: number of representative points.

I: intervals of the representative points.

j: number of representative points intervals.

Chapter 2

Methodology

2.1 The Maximum Likelihood Estimation

There are various methods for estimating the mean and variance from a given i.i.d. (independent identically distributed) sample. The maximum likelihood estimation (MLE) is the most popular method for estimating parameters when the density function of a continuous distribution is known up to some unknown parameters such as the univariate normal $N(\mu, \sigma^2)$.

In general, if a continuous population has a probability density function (PDF) $f(x; \theta)$ with an unknown parameterθ, which may be a vector. The maximum likelihood (ML) function based on a set of i.i.d. sample $\{x_1, ..., x_n\}$ is given by:

$$L(x_i; \theta) = \prod_{i=1}^{n} f(x_i; \theta) = \prod_{i=1}^{n} \frac{1}{2b} exp(-\frac{|x_i - \mu|}{b})$$

$$= (\frac{1}{2b})^n \exp\{\frac{1}{b}\sum_i^n |x_i - \mu|\}$$

log-ML Function:

$$l(x_i;\theta) = \log L(x_i;\theta) = \sum_{i=1}^{\mathrm{n}} \log f(x_i;\theta)$$

For the Laplace distribution, it's log-ML Function can be shown that,

$$l(x_i;\mu,b) = \log[(\frac{1}{2b})^n \exp\{\frac{1}{b}\sum_i^n |x_i - \mu|\}]$$

$$= -n\log(2b) - \frac{1}{b}\sum_i^n |x_i - \mu|$$

Therefore we can conclude,the log-ML function of Laplace distribution is:

$$-n\log(2b) - \frac{1}{b}\sum_i^n |x_i - \mu|$$

The next step is to find the partial derivative:

$$\frac{dl}{d\mu} = \frac{1}{b} = 0$$

$$\frac{dl}{db} = \frac{-n}{b} + \frac{1}{b^2}\sum_i^n |x_i - \mu| = 0$$

In order to calculate and explain clearly,all the b in the equations are being substituting by σ in the following thesis.

Setting these equal to 0, substituting in the MLE μ by $\hat{\mu}$,and solving gives the MLE for σ as:

$$\hat{\sigma} = \frac{1}{n}\sum_{i}^{n}|x_i - \hat{\mu}|$$

In addition,

$$\hat{\mu} = median\{x_1, x_2, ...x_n\}$$

As we mentioned,we will test these several methods by using the Root Mean Square Error (RMSE) defined by

$$RMSE = \sqrt{\frac{1}{N}\sum\nolimits_{j=1}^{N}(\widehat{\theta}_j - \theta_o)^2}$$

where $\widehat{\theta}_j$ stands for the j-th estimator from N simulations, θ_o stands for the true value of θ in the simulation.

MATLAB is used to realize the program, for the first two methods, after generating random Laplace distribution variables, we choose four sample size(50,100,200,400) to estimate the accuracy differences[4], meanwhile set the original value of μ and σ as 0 and 1, and then calculate its real number, last step would be getting the RMSE for the two parameters in different sample size.

2.2 The Moment Estimation

The moment estimation method for a population parameter σ is to use the sample central moments to estimate the population central moments which are defined by:

$$m_k(\theta) = E(X^k) = \int_{-\propto}^{+\propto} x^k f(x;\theta)dx, k = 1, 2, 3...$$

The sample central moments are defined by:

$$\frac{1}{n}\sum_{i=1}^{n} x_i^k, k = 1, 2, 3...$$

Form the moment function of Laplace Distribution, we can get the expectation of observations: mean:

$$\hat{\mu} = \mu$$

and the variance of observations: variance:

$$\hat{\sigma}^2 = 2b^2$$

$$b = \frac{\sqrt{2}}{2}\hat{\sigma}$$

2.3 The Fisher Minimum Chi-square Estimation

The Fisher minimum Chi-square estimation method is based on minimizing the Pearson-Fisher chi-square statistic.[5] Let

$$-\infty = a_0 < a_1 < a_2 < \ldots < a_m < a_{m+1} = +\infty$$

Be m equiprobable points from the standard Laplace distribution($\mu = 0, b = 1$):

$$f(x; 0, 1) = \frac{1}{2} \exp\left(-|x|\right), \quad -\infty < x < +\infty$$

Besides, the cdf of standard Laplace distribution as followed:

$$\begin{array}{l} F(x; 0, 1) = \frac{1}{2}\mathrm{e}^{x}, \quad -\infty < x < 0 \\ F(x; 0, 1) = 1 - \frac{1}{2}\mathrm{e}^{-x}, \quad 0 < x < +\infty \end{array}$$

We already known that $a_0 = -\infty, a_m = +\infty$, the range is devided into five sections, then we can get the other values by using cdf,that is:

$$\begin{array}{l} \frac{1}{\mathrm{m}} = \int\limits_{-\infty}^{a_1} \frac{1}{2} \exp\left(-|x|\right) dx \\ \frac{1}{\mathrm{m}} = \int\limits_{a_{i-1}}^{a_i} \frac{1}{2} \exp\left(-|x|\right) dx, i = 2, \ldots, m \\ \frac{1}{\mathrm{m}} = \int\limits_{a_m}^{+\infty} \frac{1}{2} \exp\left(-|x|\right) dx \end{array}$$

The location-scale Laplace distribution$Y = \mu + \sigma X$has a density function: $f\left(x;\mu,\sigma\right) = \frac{1}{2\sigma} exp\left\{-\frac{|x-\mu|}{\sigma}\right\}$[3]

Let $\hat{\mu}$ and $\hat{\sigma}$ be the MLE for μ and σ, respectively. Do the location-scale transformation:

$$b_i = \hat{\mu} + \hat{\sigma} a_i, \quad i = 1, \ldots, m$$

When

$$-\infty = b_0 < b_1 < \ldots < b_m < b_{m+1} = +\infty$$

After that, we estimate the number of N, which indicates the number of observations in the interval:

$$N_i = \ \left(b_{i-1}, b_i\right), \ i = 1, \ldots, m+1$$

As the estimated equiprobable points for the location-scale Laplace distribution. Then the estimated equiprobable functions are given by:

$$p_1\left(\mu,\sigma\right) = \int\limits_{-\infty}^{\hat{\mu}+\hat{\sigma}a_1} f\left(x;\mu,\sigma\right)dx$$

$$p_i\left(\mu,\sigma\right) = \int\limits_{\hat{\mu}+\hat{\sigma}a_{i-1}}^{\hat{\mu}+\hat{\sigma}a_i} f\left(x;\mu,\sigma\right)dx, \quad i = 2, \ldots, m_i$$

$$p_{m+1}\left(\mu,\sigma\right) = \int\limits_{\hat{\mu}+\hat{\sigma}a_m}^{+\infty} f\left(x;\mu,\sigma\right)dx$$

We can use the above functions in the equiprobable equations to solve the below function:

$$\sum_{i=1}^{m+1} \frac{N_i}{p_i(\theta)} \frac{\partial p_i(\theta)}{\partial \theta_j} = 0, \quad \theta = (\mu, \sigma), \quad j = 1,\ 2,\ \theta_1 = \mu,\ \theta_2 = \sigma$$

We already got the following derivatives:

$$\frac{\partial}{\partial \mu}(p_i(\mu, \sigma)), \quad i = 1,\ \ldots,\ m+1$$

And

$$\frac{\partial}{\partial \sigma}(p_i(\mu, \sigma)), \quad i = 1,\ \ldots,\ m+1$$

The last step, substituting these derivatives into the equiprobable equations:

$$\sum_{i=1}^{m+1} \frac{N_i}{p_i(\theta)} \frac{\partial p_i(\theta)}{\partial \theta_j} = 0, \quad \theta = (\mu, \sigma), \quad j = 1,\ 2,\ \theta_1 = \mu,\ \theta_2 = \sigma$$

The former two methods are a little complex than the previous ones, we assume the original value of μ and σ as 0 and 1, using the estimated equiprobable points for the location-scale Laplace distribution.
Then the estimated equiprobable functions (mentioned before).Then do two experiments,holding the original value and then set the m=4 and m=9(intervals,equiprobable points is m - 1), repeat the simulation and get the results.

2.4 The RP Minimum Chi-square Estimation

The RP (representative point) minimum chi-square is to choose the classification interval as follows.[2] Let$\{\mathrm{R}_i^0 : i =, ..., m\}$be a set of representative points (RPs) from $f_0(\cdot)$, which stands for the standard Laplace Distribution, whose RPs (representative points) can be obtained by existing algorithm. Then a set of RPs from $f(x; \mu, \sigma)$ can be estimated from:

$$R_i = \hat{\mu} + \widehat{\sigma} R_i^0, i = 1, ..., m$$

Where $\hat{\mu}$ and $\hat{\sigma}$ are MLEs of μ and σ, respectively. Construct the cell intervals by

$$\begin{array}{l} I_1 = \left(- \propto, \frac{R_1+R_2}{2}\right), I_j = \left(\frac{R_{j-1}+R_j}{2}, \frac{R_j+R_{j+1}}{2}\right), j = 2, ..., m-1, \\ I_m = \left(\frac{R_{m-1}+R_m}{2}, + \propto\right) \end{array} \quad [7]$$

Under these classification intervals, the probabilities can be computed by equations. The solution to equation of defining the Fisher's minimum chi-square estimator in here is called the RP minimum chi-square estimator for $\theta = (\mu, \sigma)$.

2.5 Calculation Of The Type I Error

In the project, we will do a simulation study on the performance of the statistic χ^2 in controlling type I error rates.Based on the four types of estimators for θ, we want to construct the Pearson-Fisher chi-square test for the goodness-of-fit problem:

$$H_0 : G(x) = F(x; \theta), versus H_1 : G(x) \neq F(x; \theta)[2],$$

Where $G(x)$ stands for the population distribution function of an i.i.d. sample, $F(x; \theta)$ stands for the distribution function of the location-scale Laplace distribution.

Besides,the Pearson-Fisher chi-square test is constructed by

$$X_n^2 = \sum\nolimits_{i=1}^{m} \frac{(N_i - n\hat{p})^2}{n\hat{p}}[3],$$

where $\hat{p} = p(\hat{\theta})$, $p(\hat{\theta})$ is the estimator of θ by one of the above-mentioned four estimation methods.

Chapter 3

Summary Results

3.0.1 Matlab Output Of MLE Estimation

Table 3.1: Matlab Output Of MLE Estimation,N = 1000

n	μ	σ	$\hat{\mu}$	$\hat{\sigma}$	RMSE1(μ)	RMSE2(σ)
50	0	1	0.0068	0.9928	0.1441	0.1593
100	0	1	-0.0079	0.9910	0.0982	0.1081
200	0	1	0.0011	0.9940	0.0726	0.0755
400	0	1	0.0022	0.9981	0.0495	0.0527

3.0.2 Matlab Output Of Moment Estimation

Table 3.2: Matlab Output Of Moment Estimation,N = 1000

n	μ	σ	$\hat{\mu}$	$\hat{\sigma}$	RMSE1(μ)	RMSE2(σ)
50	0	1	-0.0021	0.9852	0.2020	0.1585
100	0	1	-0.00075	0.9946	0.1312	0.1103
200	0	1	-0.00064	0.9951	0.0993	0.0809
400	0	1	-0.00066	0.9972	0.0702	0.0539

3.0.3 Matlab Output Of Fisher Minimum Chi-square Estimation

Table 3.3: Matlab Output Of Fisher Minimum Chi-square Estimation,m=4,N=1000

n	μ	σ	$\hat{\mu}$	$\hat{\sigma}$	RMSE1(μ)	RMSE2($sigma$)
50	0	1	0.0361	0.9913	0.2250	0.3215
100	0	1	-0.00457	0.8938	0.1942	0.2343
200	0	1	0.0391	0.9734	0.1372	0.1217
400	0	1	0.0483	0.9793	0.0620	0.1112

Table 3.4: Matlab Output Of Fisher Minimum Chi-square Estimation,m=9,N=1000

n	μ	σ	$\hat{\mu}$	$\hat{\sigma}$	RMSE1(μ)	RMSE2($sigma$)
50	0	1	0.0897	0.7955	0.1827	0.3034
100	0	1	0.0170	1.0616	0.1101	0.2258
200	0	1	0.0144	0.9578	0.1648	0.0969
400	0	1	0.0019	1.0174	0.1156	0.0413

3.0.4 Matlab Output Of RP Minimum Chi-square Estimation

Table 3.5: Matlab Output Of RP Minimum Chi-square Estimation,n=50, N = 1000

m	μ	σ	$\hat{\mu}$	$\hat{\sigma}$	RMSE1(μ)	RMSE2(σ)
5	0	1	-0.0335	0.9611	0.1951	0.1316
10	0	1	-0.0014	1.0104	0.0948	0.1195
15	0	1	-0.0325	0.9632	0.1487	0.1459
20	0	1	-0.0054	1.0040	0.1901	0.0853

Table 3.6: Matlab Output Of RP Minimum Chi-square Estimation,n=100, N = 1000

m	μ	σ	$\hat{\mu}$	$\hat{\sigma}$	RMSE1(μ)	RMSE2(σ)
5	0	1	0.0515	0.9785	0.1469	0.0724
10	0	1	-0.0089	0.9854	0.0901	0.0985
15	0	1	0.0232	1.0029	0.1005	0.1031
20	0	1	-0.0165	0.9864	0.1104	0.1045

Table 3.7: Matlab Output Of RP Minimum Chi-square Estimation,n=200, N = 1000

m	μ	σ	$\hat{\mu}$	$\hat{\sigma}$	RMSE1(μ)	RMSE2(σ)
5	0	1	0.0362	0.9903	0.0803	0.0685
10	0	1	0.0064	1.0011	0.0864	0.1459
15	0	1	-0.0301	0.9800	0.0754	0.0391
20	0	1	0.0080	1.0341	0.0634	0.0664

Table 3.8: Matlab Output Of RP Minimum Chi-square Estimation ,n=400,N = 1000

m	μ	σ	$\hat{\mu}$	$\hat{\sigma}$	RMSE1(μ)	RMSE2(σ)
5	0	1	0.0173	1.0057	0.0446	0.0501
10	0	1	0.0137	0.9822	0.0563	0.0517
15	0	1	0.0426	1.0181	0.0510	0.0416
20	0	1	0.0484	1.0048	0.0633	0.0517

3.0.5 Matlab Output Of Type I Error

Table 3.9: Matlab Output Of Type I Error,N=500
Sample size n = 50,100,200,400,
$(\mu = 0, \sigma = 1, alpha_1 = 0.01, alpha_2 = 0.05, alpha_3 = 0.1)$

Estimations		Moment		MLE		Fisher	RP
n	m	Fisher cells	RP cells	Fisher cells	RP cells	Fisher cells	RP cells
50	5	0.0380	0.0440	0.0200	0.0100	0.0140	0.0106
		0.1480	0.1620	0.0940	0.0880	0.1020	0.0690
		0.2440	0.2580	0.1580	0.1560	0.1600	0.1305
	10	0.0280	0.0380	0.0140	0.0100	0.0020	0.0170
		0.1040	0.1100	0.0560	0.0440	0.0260	0.0078
		0.1920	0.2000	0.1240	0.1040	0.0720	0.1130
	20	0.0220	0.0320	0.0100	0.0100	0.0141	0.0110
		0.0720	0.0960	0.0420	0.0440	0.0560	0.0489
		0.1180	0.1600	0.0860	0.1040	0.0900	0.1140
100	5	0.0602	0.0500	0.0260	0.0060	0.0100	0.0120
		0.1980	0.1640	0.0700	0.0580	0.0560	0.0700
		0.3160	0.2920	0.1380	0.1320	0.1440	0.1540
	10	0.0340	0.0180	0.0080	0.0060	0.0060	0.0120
		0.1180	0.1060	0.0520	0.0560	0.0530	0.0500
		0.2080	0.1980	0.0920	0.0980	0.0540	0.0940
	20	0.0100	0.0260	0.0040	0.0160	0.0960	0.0140
		0.0900	0.1020	0.0380	0.0540	0.0598	0.0540
		0.1640	0.1680	0.1080	0.1020	0.1127	0.0960
200	5	0.0504	0.0580	0.0120	0.0240	0.0160	0.0120
		0.1700	0.1940	0.0620	0.0860	0.0780	0.0660
		0.2760	0.3100	0.1520	0.1560	0.1460	0.1200
	10	0.0380	0.0280	0.0120	0.0160	0.0020	0.0108
		0.1420	0.1020	0.0640	0.0500	0.0540	0.0510
		0.2120	0.1700	0.1080	0.0940	0.1040	0.1000
	20	0.0280	0.0140	0.0200	0.0040	0.0220	0.0103
		0.0860	0.0640	0.0620	0.0660	0.0604	0.0479
		0.1360	0.1520	0.1140	0.1160	0.1110	0.1074
400	5	0.0480	0.0560	0.0100	0.0200	0.0340	0.0021
		0.1820	0.1880	0.0600	0.0800	0.0780	0.0550
		0.2820	0.3280	0.1340	0.1480	0.1380	0.1026
	10	0.0460	0.0380	0.0060	0.0120	0.0180	0.0130
		0.1306	0.1020	0.0380	0.0580	0.0700	0.0043
		0.2160	0.1740	0.0920	0.1040	0.1340	0.1220
	20	0.0360	0.0220	0.0120	0.0100	0.0180	0.0240
		0.1000	0.0720	0.0560	0.0540	0.0550	0.0700
		0.1600	0.1540	0.1140	0.1000	0.1020	0.1320

Chapter 4

Conclusions

Since the purpose of this thesis is to carry out extensive simulation (Monte Carlo) study to prove which of the four estimators can give better accuracy by using the Root Mean Square Error (RMSE).Overall, statistic X_{n}^2 in controlling type I error rates in the last step of project, was also a reference to esitimate the simulation study on the performance of the four alternative estimations.

Therefore,after comparing the figures of RMSE for estimating μ and σ in four alternative estimations methods(the moment method, the maximum likelihood method, the minimum chi-square method with equiprobable grouping, and the grouping based on statistical representative points), the table(2.1.1) give evidence to draw the conclusion. The maximum likelihood method (MLE) obviously fit the initial value($\mu = 0$,$\sigma = 1$) the best, and also it has smallest RMSE which indicates that the bias and error are lower than the other three methods. As the sample size growing up, the RMSE decreasing continuous and stable, and when n=400, the RMSE(μ) and RMSE(σ) just equal to 0.00495 and 0.00527, both of them are steady and in a smaller value, getting close to 0.

Meanwhile, when focusing on the type I error rate, the column of using MLE to estimate Fisher cell and RP cell are the ones which get closest to the setting value of $alpha_1 = 0.01, alpha_2 = 0.05, alpha_3 = 0.1$, with a sample size of 400 especially. For the MLE to estimate RP cells, there is no doubt the most accurate value (0.001,0.0540,0.1000).

The conclusion will be summarize simply that the maximum likelihood method carry out extensive simulation (Monte Carlo) study to give the best accuracy by using the Root Mean Square Error (RMSE) and statistic X_{n}^2 in controlling type I error rates among four estimation methods.

Chapter 5

Discussion

After completing this study, actually I thought that there still existed plenty of limitations . For the selection of the initial value and chosen data, I only chose $\mu = 0$ and $\sigma = 1$, then the simulation results maybe too simple to provide strong evidence for conclusion. As shown in the figure of the pdf [6] curve of Laplace distribution, the different initial value will lead to different figures. Providing that I choose another value as the initial value, the results may have some differences.

In addition, the times of simulation are not large enough, which means the accuracy of data cannot represent the best result. Since it took a long time to run the program in MATLAB, I just chose N=1000 or N=500 to do the simulation. The results obviously could give us a result of comparison for the study, but if it is possible in the future, a large sample size will be a better choice.

The next step of the project would be conduct the simulation on power which against some selected alternative distributions,which would be more accurate on testing the goodness of fit for these four methods.Due to the

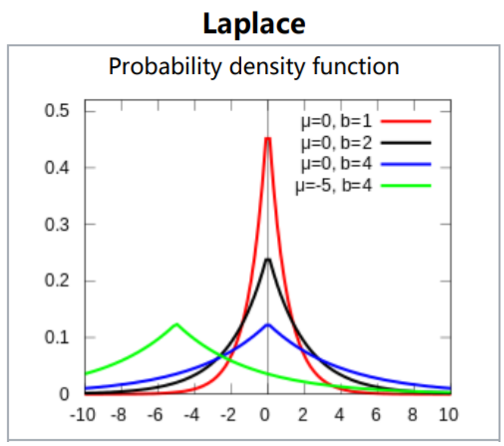

Figure 5.1: Probability density distribution of Laplace distribution.

limitation of the time, i haven't finish this part, while providing that there are opportunities in the future, i would finish these whole process.

Appendix A

Codes in MATLAB

A.1 Codes for generating random variables in Laplace distribution.

```
function y  = laprnd(m, n, mu, sigma)
%LAPRND generate i.i.d. laplacian random number drawn
from laplacian distribution
u = rand(m, n)-0.5;
y = mu - sigma.*sign(u).*log(1- 2* abs(u));
end
```

A.2 Codes for Maximum Likelihood Estimation

```
% Maximum Likelihood Method
```

```
clear
n = 400; % Sample Size
sigma = 1;
mu=0;
N = 1000; % Number of times simulated
SSE1 = [];
SSE2 = [];
new_sigma=[];
new_mu=[];
for i=1:N
x=laprnd(n,1,mu,sigma);
mu_hat = median(x);
b_hat = mean(abs(x-mu_hat)); % ML Estimation
SSE1 = [SSE1,(b_hat - sigma) ^ 2]; % Storing SSE
SSE2 = [SSE2,(mu_hat - mu) ^ 2];
new_sigma(i)=b_hat;
new_mu(i)=mu_hat;
end
mb=mean(new_sigma)
mmu=mean(new_mu)
RMSE1 = sqrt(mean(SSE1))
RMSE2 = sqrt(mean(SSE2))
```

A.3 Codes for Moment Estimation

```
% Moment Method
n = 50; % Sample Size
```

```
b = 1;
mu=0;
N = 1000; % Number of times simulated
SSE1 = [];
SSE2 = [];
new_b=[];
new_mu=[];
for i=1:N
X=laplace(mu,b,n);
new_mu=mean(X);
SSE1 = [SSE1,(new_mu(i) - mu). ^ 2];
new_b=sqrt(1/2*var(X));
SSE2 = [SSE2,(new_b(i) - b). ^ 2];
end
mean(new_b)
mean(new_mu)
RMSE1 = sqrt(mean(SSE1))
RMSE2 = sqrt(mean(SSE2))
```

A.4 Codes for generating Fisher Equiprobable equations

```
function Y = Fisher_Equiprobable_Equations(X)
global n m B w mu_hat sigma_hat
y = mu_hat + sigma_hat .* w; % Generalised laplace rnd
C = 0;
```

```
D = 0;
global diffmu diffsig
syms x
for i = 1 : m
N(i) = length(find(y >= B(i) & y < B(i + 1)));
P = double(vpaintegral(laprnd(x,X(1),X(2)),
x,B(i),B(i + 1)));
DIFF_mu = double(vpaintegral(diffmu,x,B(i),
B(i + 1)));
DIFF_sigma = double(vpaintegral(diffsig,x,B(i),
B(i + 1)));
C = C + N(i) / P * DIFF_mu;
D = D + N(i) / P * DIFF_sigma;
end
Y = [C;D];
end
```

A.5 Codes for generating Equiprobable Points and intervals.

```
global m n mu sigma
A = [-Inf]; % Cutting Points of Laplace
G = @(x1,x2)(cdflaplace(x1,mu,sigma) - cdflaplace
(x2,mu,sigma) - 1 / m);
i = 0;
```

```
while i < m
if i == 0
x1=-10000;
x2=fsolve(@(x2)G(x1,x2),0);
else if i<m-1
x1=x2;
x2=fsolve(@(x2)G(x1,x2),x1);
else
x1=x2;
x2=Inf;
end
A=[A,x2];
i=i+1;
end
end
global mu_hat sigma_hat B
B = mu_hat + sigma_hat .* A;
 % Cutting Points of Generalised Laplace variable
global diffmu diffsigma
syms x Mu1
syms Sigma1 positive
diffmu1 = diff(cdflaplace(x,Mu1,Sigma1),Mu1,1);
diffsigma1 = diff(cdflaplace(x,Mu1,Sigma1),
Sigma1,1);
diffmu = subs(diffmu1,{Mu1,Sigma1},{mu_hat,
sigma_hat});
diffsigma = subs(diffsigma1,{Mu1,Sigma1},
```

```
{mu_hat,sigma_hat});
```

A.6 Codes for calculating the RMSE of Fisher-minimum Chi-square Estimation

```
global m n m1 n1 mu_hat sigma_hat
n1 = 400; % Sample Size
m1 = 5; % # intervals(# equiprobable points is m - 1)
mu = 1;
sigma = 2;
%[mu_hat,sigma_hat] = mle_rmse(mu,sigma);
%mu_hat = 1.0033;
%sigma_hat = 1.9996;
global y
mu_vec_Eq_MCS = [];
sigma_vec_Eq_MCS = [];

for j = 1 : 10
y= laprnd(1,n,mu,sigma); % laplace rnd
% Initial values of the estimators
mu_hat = median(y);
sigma_hat = mean(abs(y-mu_hat));
x00=[mu_hat,sigma_hat];
j
[x,fval] = fsolve('Fisher_Equiprobable_Equations',x00,
optimoptions('fsolve','Display','iter'))
```

```
mu_vec_Eq_MCS = [mu_vec_Eq_MCS,x(1)];
sigma_vec_Eq_MCS = [sigma_vec_Eq_MCS,x(2)];
end
mean(mu_vec_Eq_MCS)
mean(sigma_vec_Eq_MCS)
MSE_mu_Eq_MCS = sqrt(mean((mu_vec_Eq_MCS - mu)
 .^ 2))
MSE_sigma_Eq_MCS = sqrt(mean((sigma_vec_Eq_MCS -
 sigma) .^ 2))
```

A.7 Codes for the selection of RP and the functions of RP equations.

```
function MSE_RP=X_MSE_l(m)
if m == 5
MSE_RP = [ -3.1872, -1.1872, 0, 1.1872, 3.1872];
end
if m == 10
MSE_RP = [-4.9657, -2.9657, -1.7784,  -0.9305,
-0.2704, 0.2704 ,0.9305 ,1.7784 ,2.9657,  4.9657];
end
if m == 15
MSE_RP=[ -6.0911,-4.0911,-2.9039 ,-2.0560 ,
 -1.3958, -0.8551 , -0.3972,  0 ,0.3972 ,0.8551
   1.3958, 2.0560, 2.9039 ,4.0911, 6.0911];
end
```

```
if m == 20
MSE_RP =[ -6.8979,-4.8979, -3.7106 ,-2.8627,
 -2.2025 ,-1.6618,  -1.2039 ,-0.8067, -0.4560 ,
  -0.1421 ,0.1421, 0.4560 ,0.8067 ,1.2039 ,
   1.6618 ,2.2025 ,2.8627,  3.7106 ,4.8979 ,6.8979];
end
end

function Y = RP_Equations(X)
global n x m B y diffmu diffsig
C = 0;
D = 0;
syms x
for i = 1 : m
N(i) = length(find(y >= B(i) & y < B(i + 1)));
syms x Mu1
syms Sigma1 positive
pdflaplace = @(x,Mu1,Sigma1)exp(-abs(x - Mu1)
/ Sigma1) / (2 * Sigma1);
P = double(int(pdflaplace(x,X(1),X(2)),x,B(i),
B(i + 1)));
DIFF_mu = double(vpaintegral(diffmu,x,B(i),
B(i + 1)));
DIFF_sigma = double(vpaintegral(diffsig,x,B(i),
B(i + 1)));
C = C + N(i) / P * DIFF_mu;
D = D + N(i) / P * DIFF_sigma;
```

```
end
Y = [C;D];
end
```

A.8 Codes for calculating the RMSE of RP methods.

```
global n m mu sigma mu_hat sigma_hat y R
n = 400; % Sample Size
m = 10; % # intervals
mu = 0;
sigma = 1;
mu_vec_RP_MCS = [];
sigma_vec_RP_MCS = [];
for j = 1 : 10
u = rand(1, n)- 0.5;
w = - sign(u).*log(1- 2* abs(u));
y = mu + sigma .* w;
% Initial values of the estimators
mu_hat = median(y);
sigma_hat = mean(abs(y-mu_hat));
x0 = [mu_hat,sigma_hat];
j
[x,fval] = fsolve('RP_Equations', x0,
optimoptions('fsolve','Display', 'iter'))
mu_vec_RP_MCS = [mu_vec_RP_MCS,x(1)];
```

```
sigma_vec_RP_MCS = [sigma_vec_RP_MCS,x(2)];
Chisq = 0;
end
for i = 1 : m
N = length(find(y >= R(i) & y < R(i + 1)));
syms r;
pdflaplace = @(x,Mu1,Sigma1)exp(-abs(x - Mu1)
 / Sigma1) / (2 * Sigma1);
P = double(int(pdflaplace(r,x(1),x(2)),r,R(i)
,R(i + 1)));
Chisq = Chisq + (N - n * P) ^ 2 / (n * P);
end
mean(mu_vec_RP_MCS)
mean(sigma_vec_RP_MCS)
MSE_mu_RP_MCS = sqrt(mean((mu_vec_RP_MCS - mu)
 .^ 2))
MSE_sigma_RP_MCS = sqrt(mean((sigma_vec_RP_MCS
- sigma) .^ 2))
```

A.9 Codes for Calculating the Type I error of MLE and Moment Estimation

```
global x n m y
global new_mu new_sigma
global mu_hat sigma_hat
n = 400; %
```

```
m = 10; %
mu = 0;%
sigma = 1;%
%%%%%%%%%%%
alpha1 = 0.01;
alpha2 = 0.05;
alpha3 = 0.1;
% Liang's talk ppt slide 24
critical1 = chi2inv(1 - alpha1,m - 3);
critical2 = chi2inv(1 - alpha2,m - 3);
critical3 = chi2inv(1 - alpha3,m - 3);
%choice_cell = input("Please input a number
 to choose a way of classifying cells. " + ...
% "Default is RP, and press 1 for
 Fisher Equiprobable. ");
choice_cell = 1;
if choice_cell == 1
A = [-Inf]; % Cutting Points of laplace
cdflaplace = @(x,mu,sigma) 0.5*(1 + sign(x-mu)
.*(1 - exp( -abs(x-mu)/(sigma/sqrt(2)) )));;
G = @(x1,x2)(cdflaplace(x2,0,1) - cdflaplace(x1,0,1)
 - 1 / m);
i = 0;
while i < m
if i == 0
x1 = -Inf;
x2 = fsolve(@(x2)G(x1,x2),0);
```

```
elseif i < m - 1
x1 = x2;
x2 = fsolve(@(x2)G(x1,x2),x1);
else
x1 = x2;
x2 = Inf;
end
A = [A,x2];
i = i + 1;
end
else
A = [-Inf,X_MSE_l(m),Inf];
end
syms x
Reject1 = 0;
Reject2 = 0;
Reject3 = 0;
%choice_estimation = input("Please input
a number to choose a method of estimation. " + ...
%       "Default is MLE, and press 1 for moment
estimation. ");
choice_estimation = 0;
for k = 1 : 500
k
if choice_estimation == 1
[mu_hat,sigma_hat] = moment(mu,sigma);
else
```

```
[mu_hat,sigma_hat] = f_mle(mu,sigma);
end
X = [mu_hat,sigma_hat];
if choice_cell == 1
B = X(1) + X(2) .* A; % Cutting Points of laplace
else
B = X(1) + X(2) .* A; % Cutting Points of laplace
for i = 2 : m
B(i) = (B(i) + B(i + 1)) / 2;
end
B(m + 1) = Inf;
B(m + 2) = [];
end
Chisq = 0;
for i = 1 : m
N = length(find(y >= B(i) & y < B(i + 1)));
syms r;
pdflaplace = @(x,Mu1,Sigma1)exp(-abs(x - Mu1)
/ Sigma1) / (2 * Sigma1);
P = double(vpaintegral(pdflaplace(x,X(1),X(2)),
x,B(i),B(i + 1)));
Chisq = Chisq + (N - n * P) ^ 2 / (n * P);
end
% Alpha = 0.01
if Chisq > critical1
reject1 = 1;
else
```

```
reject1 = 0;
end
Reject1 = Reject1 + reject1
% Alpha = 0.05
if Chisq > critical2
reject2 = 1;
else
reject2 = 0;
end
Reject2 = Reject2 + reject2
% Alpha = 0.1
if Chisq > critical3
reject3 = 1;
else
reject3 = 0;
end
Reject3 = Reject3 + reject3
end
n
m
choice_cell
choice_estimation
Rejection_rate1 = Reject1 / 500
Rejection_rate2 = Reject2 / 500
Rejection_rate3 = Reject3 / 500
```

A.10 Codes for Calculating the Type I error of Fisher Minimum Chi-square and RP Estimations

```
global n m mu sigma mu_hat sigma_hat y B X
n = 50; % Sample Size
m = 10; % # intervals(# equiprobable points is m - 1)
mu = 0;
sigma = 1;
mu_vec_Eq_MCS = [];
sigma_vec_Eq_MCS = [];
alpha1 = 0.01;
alpha2 = 0.05;
alpha3 = 0.1;
% Liang's talk ppt slide 24
critical1 = chi2inv(1 - alpha1,m - 3);
critical2 = chi2inv(1 - alpha2,m - 3);
critical3 = chi2inv(1 - alpha3,m - 3);
Reject1 = 0;
Reject2 = 0;
Reject3 = 0;
%choice = input("Please input a number to choose
 a method of estimation. " + ...
%"Default is MLE, and press 1 for moment estimation. ");
choice = 0;
A = [-Inf]; % Cutting Points of laplace
```

```
cdflaplace = @(x,mu,sigma) 0.5*(1 + sign(x-mu)
.*(1 - exp( -abs(x-mu)/(sigma/sqrt(2)) )));;
G = @(x1,x2)(cdflaplace(x2,0,1) - cdflaplace
(x1,0,1) - 1 / m);
i = 0;
while i < m
if i == 0
x1 = -Inf;
x2 = fsolve(@(x2)G(x1,x2),0);
elseif i < m - 1
x1 = x2;
x2 = fsolve(@(x2)G(x1,x2),x1);
else
x1 = x2;
x2 = Inf;
end
A = [A,x2];
i = i + 1;
end

for j = 1 : 500
% Initial values of the estimators
if choice == 1
[mu_hat,sigma_hat] = moment(mu,sigma);
else
[mu_hat,sigma_hat] = f_mle(mu,sigma);
end
```

```
X = [mu_hat,sigma_hat];
B = mu_hat + sigma_hat .* A;
x0 = [mu_hat,sigma_hat];
j
[x,fval] =
fsolve('Fisher_Equiprobable_Equations', x0, optimoptions('fsolve','Display', 'ite
mu_vec_Eq_MCS = [mu_vec_Eq_MCS,x(1)];
sigma_vec_Eq_MCS = [sigma_vec_Eq_MCS,x(2)];

Chisq = 0;
for i = 1 : m
N = length(find(y >= B(i) & y < B(i + 1)));
syms r;
pdflaplace = @(x,Mu1,Sigma1)exp(-abs(x - Mu1) /
 Sigma1) / (2 * Sigma1);
P = double(vpaintegral(pdflaplace(r,X(1),X(2)),
r,B(i),B(i + 1)));
Chisq = Chisq + (N - n * P) ^ 2 / (n * P);
end
% Alpha = 0.01
if Chisq > critical1
reject1 = 1;
else
reject1 = 0;
end
Reject1 = Reject1 + reject1
% Alpha = 0.05
```

```
if Chisq > critical2
reject2 = 1;
else
reject2 = 0;
end
Reject2 = Reject2 + reject2
% Alpha = 0.1
if Chisq > critical3
reject3 = 1;
else
reject3 = 0;
end
Reject3 = Reject3 + reject3
end
n
m
mean(mu_vec_Eq_MCS)
mean(sigma_vec_Eq_MCS)
MSE_mu_Eq_MCS = sqrt(mean((mu_vec_Eq_MCS - mu)
 .^ 2))
MSE_sigma_Eq_MCS = sqrt(mean((sigma_vec_Eq_MCS
- sigma) .^ 2))
Rejection_rate1 = Reject1 / 500
Rejection_rate2 = Reject2 / 500
Rejection_rate3 = Reject3 / 500

global n m mu sigma mu_hat sigma_hat y B X
```

```
n = 400; % Sample Size
m = 10; % # intervals
mu = 0;
sigma = 1;
mu_vec_RP_MCS = [];
sigma_vec_RP_MCS = [];
alpha1 = 0.01;
alpha2 = 0.05;
alpha3 = 0.1;
% Liang's talk ppt slide 24
critical1 = chi2inv(1 - alpha1,m - 3);
critical2 = chi2inv(1 - alpha2,m - 3);
critical3 = chi2inv(1 - alpha3,m - 3);
Reject1 = 0;
Reject2 = 0;
Reject3 = 0;
%choice = input("Please input a number to
choose a method of estimation. " + ...
% "Default is MLE, and press 1 for moment
 estimation. ");
choice = 0;
A = [-Inf,(X_MSE_l(m)),Inf];
for j = 1 : 500
% Initial values of the estimators
if choice == 1
[mu_hat,sigma_hat] = moment(mu,sigma);
else
```

```
[mu_hat,sigma_hat] = f_mle(mu,sigma);
end
X = [mu_hat,sigma_hat];
B = mu_hat + sigma_hat .* A; % Cutting Points
 of Generalising laplace distribution
for i = 2 : m
B(i) = (B(i) + B(i + 1)) / 2;
end
B(m + 1) = Inf;
B(m + 2) = [];
x0 = [mu_hat,sigma_hat];
j
[x,fval] = fsolve('RP_Equations', x0,
optimoptions('fsolve','Display', 'iter'))
mu_vec_RP_MCS = [mu_vec_RP_MCS,x(1)];
sigma_vec_RP_MCS = [sigma_vec_RP_MCS,x(2)];

Chisq = 0;
for i = 1 : m
N = length(find(y >= B(i) & y < B(i + 1)));
syms r;
pdflaplace = @(x,Mu1,Sigma1)exp(-abs(x - Mu1)
 / Sigma1) / (2 * Sigma1);
P = double(int(pdflaplace(r,X(1),X(2)),r,B(i),
B(i + 1)));
Chisq = Chisq + (N - n * P) ^ 2 / (n * P);
end
```

```
% Alpha = 0.01
if Chisq > critical1
reject1 = 1;
else
reject1 = 0;
end
Reject1 = Reject1 + reject1
% Alpha = 0.05
if Chisq > critical2
reject2 = 1;
else
reject2 = 0;
end
Reject2 = Reject2 + reject2
% Alpha = 0.1
if Chisq > critical3
reject3 = 1;
else
reject3 = 0;
end
Reject3 = Reject3 + reject3
end
%n
%m
%mean(mu_vec_RP_MCS)
%mean(sigma_vec_RP_MCS)
%MSE_mu_RP_MCS = sqrt(mean((mu_vec_RP_MCS - mu)
```

```
.^ 2))
%MSE_sigma_RP_MCS = sqrt(mean((sigma_vec_RP_MCS
- sigma) .^ 2))
Rejection_rate1 = Reject1 / 500
Rejection_rate2 = Reject2 / 500
Rejection_rate3 = Reject3 / 500
```

Bibliography

[1] D'Agostino, R. B. and Stephens, M. A. (1986). *Goodness-of-Fit Techniques.* ,Marcel Dekker,Inc.

[2] Fang, K. T. and He, S. (1982). The problem of selecting a given number of representative points in a normal population and a generalized mill's ratio. *Technical Report No 5, Department of Statistics, Stanford University, USA.*

[3] Rinne, H. (unpublished manuscript). *Location-Scale Distributions: Linear Estimation and Probability Plotting Using MATLAB*?,

[4] Puig, P. and Stephens, M. A. (2000). *Tests of Fit for the Laplace Distribution, with Applications. Technometrics, 42 (4), 417-424*

[5] Voinov, V., Nikulin, M., and Balakrishnan, N. (2013). *Chi-Squared Goodness of Fit Tests with Applications. Elsevier.*

[6] Gragh of probability density distribution of Laplace distribution (2020). Wikipedia:Laplce distribution

[7] Fang, K. T. and He Ping, Yang Jun (2020). Representative point set of statistical distribution and its application. *Chinese Science: Mathematics, 50, 1149-1168*