GENERALIZED METHOD OF MOMENTS IN EXPONENTIAL DISTRIBUTION FAMILY

Yanzhao Lai

A Thesis Submitted to the
University of North Carolina Wilmington in Partial Fulfillment
of the Requirements for the Degree of
Master of Science

Department of Mathematics and Statistics

University of North Carolina Wilmington

2009

Approved by

Advisory Committee

Dr.Xin Lu

Dr.Susan Simmons

Dr.Yishi Wang

Chair

Accepted by

DN: cn=Robert D. Roer, o=UNCW, ou=Dean of the Graduate School & Research, email=roer@uncw.edu, c=US
Date: 2009.12.16 10:30:02 -05'00'

Dean, Graduate School

TABLE OF CONTENTS

ABSTRACT iv

DEDICATION v

ACKNOWLEDGMENTS vi

LIST OF TABLES vii

LIST OF FIGURES viii

1 INTRODUCTION 1

2 BACKGROUND 2

2.1 STATISTICAL DISTRIBUTIONS 2

2.2 MOMENTS IN STATISTICS 5

2.3 MATRIX IN STATISTICS 7

2.4 NEWTON-RAPHSON METHOD IN TWO-DIMENSION . . 10

3 THE METHOD OF MOMENTS 13

3.1 INTRODUCTION OF THE METHOD OF MOMENTS . . . 13

3.2 EXAMPLES OF THE METHOD OF MOMENTS 15

4 THE GENERALIZED METHOD OF MOMENTS (GMM) ESTIMATOR 16

4.1 MINIMUM DISTANCE ESTIMATION 16

4.2 INTRODUCTION OF GMM 17

4.3 PROPERTIES OF THE GMM ESTIMATORS 25

4.4 MOMENTS CONDITION IN GMM 31

5 SUMMARY AND CONCLUSION 35

REFERENCES 36

APPENDIX 37

A. R CODE FOR FOUR MOMENTS IN NORMAL DISTRIBUTION 37

B. R CODE FOR THREE MOMENTS IN GAMMA DISTRIBUTION 42

B. R CODE FOR THREE MOMENTS IN NORM-123 47
B. R CODE FOR THREE MOMENTS IN NORM-124 52
B. R CODE FOR THREE MOMENTS IN NORM-134 57

ABSTRACT

This paper applys the generalized method of moments (GMM) to the exponential distribution family. We present the way to find the weighting matrix W to minimize the quadratic form $f = \bar{G}'(X, \Theta)W\bar{G}(X, \Theta)$ and show two methods to prove the S^{-1} is the optimal weight matrix where $S = G(X, \hat{\Theta}_1)G'(X, \hat{\Theta}_1)$. This paper also discusses the advantages and disadvantages in GMM computation with newton-raphson method. Some simulations are reported.

DEDICATION

I would like to dedicate this thesis to my family, whose love and support make me able to do more than I thought.

ACKNOWLEDGMENTS

I would like to deeply thank my advisor Dr. Yishi Wang for bringing me the interest to study Statistics and the courage to conquer all the obstacles in the study and research. Without his great insight, guidance, patience and encouragement all the way through my study and research, this can never be possible.

I would also like to thank all the faculty members, staff and fellow students in the Mathematics and Statistics department for all kinds of help in my studying life here.

LIST OF TABLES

1 Normal(2,9) with Moments(1,2) 21

2 Normal(2,9) with Moments(1,2,3) 21

3 Normal(2,9) with Moments(1,2,4) 21

4 Normal(2,9) with Moments(1,3,4) 21

5 Normal(2,9) with Moments(1,2,3,4) 22

6 Gamma(3,0.1) with Moment(1,2) 22

7 Gamma(3,0.1) with Moment(-1,1,2) 22

8 Normal(2,9) with Sample 2000 23

9 Normal(2,9) with Sample 1500 23

10 Normal(2,9) with Sample 500 23

11 Normal(2,9) with Sample 100 24

12 Normal(2,9) with Moments(1,2,3,4,5) 32

13 Normal(2,9) with Moments(1,2,3,4,5,6) 33

LIST OF FIGURES

1 Normal Distribution PDF 3
2 Gamma Distribution PDF 4
3 Normal Distribution 5 moments 32
4 Normal Distribution 6 moments 34

1 INTRODUCTION

Generalized Method of Moments (GMM) has become one of the main statistical tools for the analysis of economic and financial data. Accompanying this empirical interest, there is a growing literature in econometrics on GMM-based inference techniques. In fact, in many ways, GMM is becoming the common language of econometric dialogue because the framework subsumes many other statistical methods of interest, such as least squares, maximum likelihood and instrumental variables.

The exponential distribution family has a density function that can take on many possible forms commonly encountered in economical applications. This fact has led many people to study the properties of the exponential distribution family and to propose various estimation techniques (method of moments, mixed moments, maximum likelihood etc.). We discuss some of the most important properties of this flexible family of distribution and present a flexible method of parameter estimation, called the *generalized method of moments*.

In random sampling, under generally benign assumptions, a sample statistic will converge in probability to some constant. This constant will in turn, be a function of the unknown parameters of the distribution. To estimate p parameters, $\theta_1, \ldots, \theta_p$, we can compute K such statistics, $\bar{m}_1, \ldots, \bar{m}_k$, whose probability limits are known functions of the parameters. These K moments are equated to the K functions, and the functions are inverted to express the parameters as functions of the moments. But there are cases in which there are more moment equations than parameters, so the system is overdetermined. Here we use minimum distance estimation to find the *generalized method of moments (GMM) estimator* which minimizes the loss function of data and parameters :

$$L(\Theta) = \bar{G}^T(X, \Theta) W \bar{G}(X, \Theta).$$

2 BACKGROUND

2.1 STATISTICAL DISTRIBUTIONS

In probability theory, a probability distribution identifies either the probability of each value of an unidentified random variable (when the variable is discrete), or the probability of the value falling within a particular interval (when the variable is continuous).[1]

- **Probability Density Function**

A **probability density function (PDF)**, or density, of a random variable is a function which describes the density of probability at each point in the sample space. The probability of a random variable falling within a given set is given by the integral of its density over the set.

A random variable Z has density f, where f is a non-negative Lebesgue-integrable function, if:

$$P(a < Z \leq b) = \int_a^b f(x)dx.$$

- **Cumulative Distribution Function**

The **cumulative distribution function (CDF)** or distribution function, completely describes the probability distribution of a real-valued random variable Z. The distribution function can be defined in terms of the probability density function f as follows:

$$F(x) = P(Z \leq x) = \int_{-\infty}^{x} f(t)dt.$$

- **Exponential family**

The general exponential family includes all the distributions, whether continuous, discrete or of mixed type, whose probability function or density can be written as follows:

The multi-parameter definition can be extended to a vector $\theta = (\theta_1, \theta_2, \ldots, \theta_p)^T$.

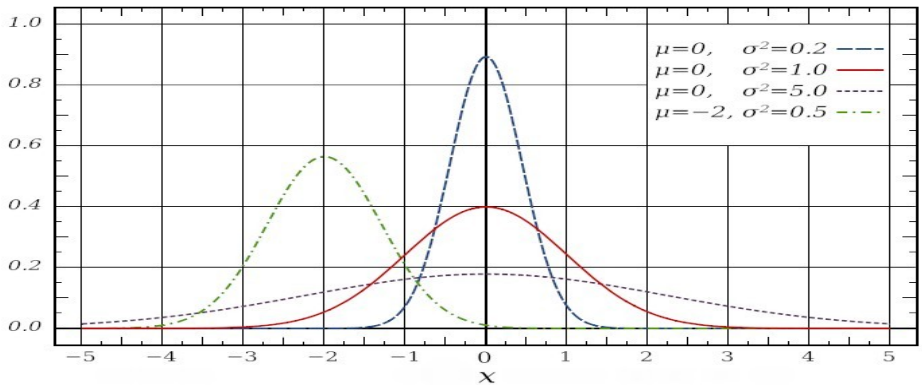

Figure 1: Normal Distribution PDF

A family of distributions is said to belong to a vector exponential family if the probability density function (or probability mass function, for discrete distributions) can be written as

$$f_X(x;\theta) = h(x)\exp(\sum_{i=1}^{p}\eta_i(\theta)T_i(x) - A(\theta)).$$

- **Normal distribution and Gamma distribution**

The normal, exponential, gamma, chi-square, beta, weibull (if the shape parameter is known), dirichlet, bernoulli, binomial, multinormal, poisson, negative binomial, and geometric distributions are all exponential families.

Normal distribution: The normal distribution or gaussian distribution is a continuous probability distribution that describes data that clusters around a mean or average.The probability density function for a normal distribution is given by the formula

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}}\exp(-\frac{(x-\mu)^2}{2\sigma^2}),$$

where μ is the mean, σ is the standard deviation (a measure of the width of the bell), and exp denotes the exponential function.

Gamma distribution: A two-parameter distribution that belongs to the expo-

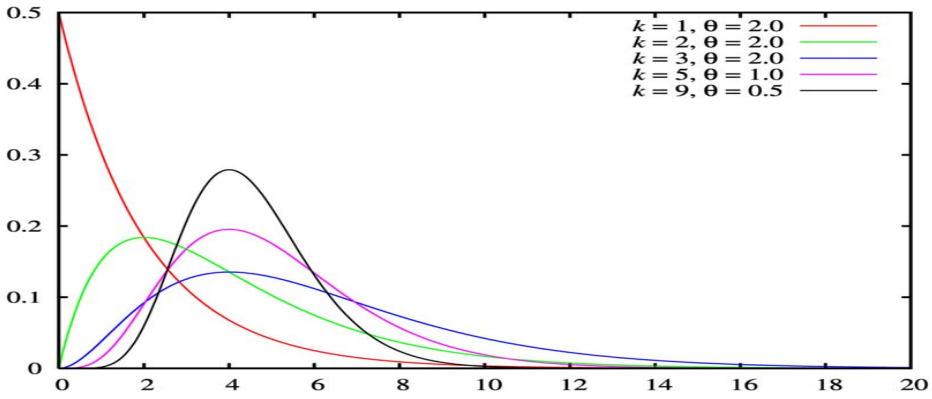

Figure 2: Gamma Distribution PDF

nential family is the gamma distribution. It has a scale parameter θ and a shape parameter α. The equation defining the probability density function of a gamma-distributed random variable Z is

$$f(x;\alpha,\theta) = x^{\alpha-1}\frac{e^{-\frac{x}{\theta}}}{\theta^{\alpha}\Gamma(\alpha)},$$

where $x > 0$, $\alpha, \theta > 0$ and,

$$\Gamma(x) = \int_0^{\infty} t^{x-1}e^{-t}dt,$$

is the gamma function, with the property that, if n is an integer then $\Gamma(n+1) = n!$ note that by definition $0! = 1$.

2.2 MOMENTS IN STATISTICS

The concept of moment in mathematics evolved from the concept of moment in physics. The n^{th} moment of a real-valued function $f(x)$ of a random variable Z about a value c is

$$\mu_n = \int_{-\infty}^{+\infty} (x-c)^n f(x) dx.$$

- **Definition**

In this paper, we define the k^{th} moments of a random variable as follows:

Let $\theta = (\theta_1, \theta_2, \ldots, \theta_p)^T$ be a vector of parameters. The k^{th} moment about 0 (if it exists) is defined as:

$$\mu_k(\theta) = E(x^k) = \int_{-\infty}^{+\infty} x^k f(x;\theta) dx,$$

where $f(x, \theta)$ is the density function of Z.

- **Moments in normal distribution and gamma distribution**

Moments in normal distribution

The density function of normal distribution is $f(x) = \frac{1}{\sigma\sqrt{2\pi}} exp(-\frac{(x-\mu)^2}{2\sigma^2})$, based on the definition of k^{th} moments of a distribution. The moments of normal distribution are as below:

$$\begin{aligned}
E(x) &= \mu, \\
E(x^2) &= \mu^2 + \sigma^2, \\
E(x^3) &= \mu^3 + 3\mu\sigma^2, \\
E(x^4) &= \mu^4 + 6\mu^2\sigma^2 + 3\sigma^4, \\
E(x^5) &= \mu^5 + 10\mu^3\sigma^2 + 15\sigma^4, \\
E(x^6) &= \mu^6 + 15\mu^4\sigma^2 + 45\mu^2\sigma^4 + 15\sigma^6.
\end{aligned}$$

Moments in gamma distribution

The density function of gamma distribution is $f(x;\alpha,\theta) = x^{\alpha-1}\frac{e^{-\frac{x}{\theta}}}{\theta^{\alpha}\Gamma(\alpha)}$, based on the definition of k^{th} moments of a distribution. The moments of gamma distribution are as below:

$$\begin{aligned} E(x^1) &= \alpha\theta, \\ E(x^2) &= \theta^2\alpha(\alpha+1), \\ E(x^{-1}) &= \frac{1}{\theta(\alpha-1)}. \end{aligned}$$

• **Vector calculus**

Let $f = (f_1, f_2, \cdots, f_m)^T$ be an $m \times 1$ vector whose elements are differentiable functions of the elements of an $n \times 1$ vector $X = (x_1, x_2, \cdots, x_n)^T$. We write $f = f(X)$ and say that f is a vector function of X. Then we have the follow definitions.

Definition 1. The derivative of f with respect to X, denoted by $\frac{\partial f}{\partial X}$, is the $n \times m$ matrix given by

$$\frac{\partial f}{\partial X} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \cdots & \ddots & \cdots \\ \frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix}.$$

Note that under this notation if $f(X)$ is a scalar function of X, the derivative $\frac{\partial f}{\partial X}$ is then $n \times 1$ vector given by

$$\frac{\partial f}{\partial X} = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \cdots \\ \frac{\partial f}{\partial x_n} \end{bmatrix}.$$

Similarly, if X is a scalar and f is an $m \times 1$ vector, then the derivative $\frac{\partial f}{\partial X}$ is the $m \times 1$ vector

$$\frac{\partial f}{\partial X} = \left[\frac{\partial f_1}{\partial X}, \frac{\partial f_2}{\partial X} \cdots \frac{\partial f_m}{\partial X}\right]^T.$$

• **Matrix calculus**

Let f be a scalar function that is differentiable functions of the elements of an $n \times m$ matrix $X = (x_{ij})$. Then the derivative $\frac{\partial f}{\partial X}$ is the $n \times m$ matrix

$$\frac{\partial f}{\partial X} = \begin{bmatrix} \frac{\partial f}{\partial x_{11}} & \cdots & \frac{\partial f}{\partial x_{1m}} \\ \cdots & \ddots & \cdots \\ \frac{\partial f}{\partial x_{n1}} & \cdots & \frac{\partial f}{\partial x_{nm}} \end{bmatrix}$$

• **Some simple matrix calculus results**

Theorem 2.1. Let X be an $n \times 1$ vector and let A be an $n \times n$ matrix of constants(i.e., the elements of A are not functions of X.) Then

$$\begin{aligned} \frac{\partial AX}{\partial X} &= A^T, \\ \frac{\partial X^T A}{\partial X} &= A, \\ \frac{\partial X^T AX}{\partial X} &= (A^T + A)X. \end{aligned}$$

Proof. The j^{th} element of AX is $\sum_k a_{jk}X_k$, and so the j^{th} column of $\frac{\partial AX}{\partial X}$ is $A'^{j\cdot}$, where $A^{j\cdot}$ is the j^{th} row of A^T and $\frac{\partial AX}{\partial X} = A'$. Under our notation, $\frac{\partial X^T A}{\partial X} = \frac{\partial A^T X}{\partial X} = A$. The j^{th} element of $\frac{\partial X^T AX}{\partial X}$ is $\sum_i a_{ij}X_i + \sum_k a_{jk}X_k$ so $\frac{\partial X^T AX}{\partial X} = (A^T + A)X$ [2].

Theorem 2.2. Let X be a scalar and A is a matrix of function of X Then

$$\frac{\partial A^{-1}}{\partial X} = -\frac{\partial A}{\partial X}A^{-1}(A^T)^{-1}.$$

Proof. We have $AA^{-1} = I_n$.Differentiating both sides with respect to X we have,

$$\frac{\partial A}{\partial X}A^{-1} + \frac{\partial A^{-1}}{\partial X}A^T = 0.$$

Solving, we obtain

$$\frac{\partial A^{-1}}{\partial X} = -\frac{\partial A}{\partial X}A^{-1}(A^T)^{-1}.$$

- **Positive-definite matrix**

An $n \times n$ real symmetric matrix M is positive definite if $Y^T MY > 0$ for all non-zero vectors Y with real entries ($Y \in R^n$). All the eigenvalues λ_i of M are positive.

Theorem 2.3. There exists a unique lower triangular matrix U, with strictly positive diagonal elements, that allows the factorization of M into $M = UU^T$, where U^T is the conjugate transpose of U.

- **Trace in linear algebra**

In linear algebra, the trace of an $n \times n$ square matrix A is defined to be the sum of the elements on the main diagonal (the diagonal from the upper left to the lower right) of A:

$$tr(A) = a_{11} + a_{22} + \ldots + a_{nn} = \sum_{i=1}^{n} a_{ii},$$

where a_{ii} represents the entry on the i^{th} row and i^{th} column of A.[3] If A is a square $n \times n$ matrix with real entries and if $\lambda_1, \cdots, \lambda_n$ are the eigenvalues of A, then

$$\begin{aligned} tr(A) &= \sum \lambda_i \\ det(A) &= \prod \lambda_i \end{aligned}$$

2.4 NEWTON-RAPHSON METHOD IN TWO-DIMENSION

Based on the Vieta's cubic Formulas[4], there is no algebra way to find the roots for the polynomial function with 4^{th} degrees or higher. We will use newton-raphson method in numerical analysis to find the root of the function.

Basic definition

Suppose there is a two dimensional vector of a function $f(x, y)$ and the partial derivatives of $f(x, y)$ are accessible.

$$\begin{aligned} Gradient \quad &: \quad \nabla f(x,y) = \left(\frac{\partial f(x,y)}{\partial x}, \frac{\partial f(x,y)}{\partial y} \right)^T \\ HessianMatrix \quad &: \quad H(\bar{x}) = H(x,y) = \begin{pmatrix} f_{xx}(x,y) & f_{xy}(x,y) \\ f_{yx}(x,y) & f_{yy}(x,y) \end{pmatrix} \end{aligned}$$

Introduction of Newton-Raphson method

Newton's method, also called the newton-raphson method, is a root-finding algorithm that uses the first few terms of the Taylor series of a function in the vicinity of a suspected root. Assume we have a polynomial function $f(x, y)$, the Taylor series of $f(x, y)$ about the point $(x + \Delta x, y + \Delta y)$ is given by

$$f(x + \Delta x, y + \Delta y) = f(x, y) + f_x(x, y)\Delta x + f_y(x, y)\Delta y +$$

$$\frac{1}{2!}(f_{xx}(x, y)\Delta x^2 + 2f_{xy}(x, y)\Delta x\Delta y + f_{yy}(x, y)\Delta y^2 + \cdots ,$$

Keeping terms only to second order,

$$f(x + \Delta x, y + \Delta y) \approx f(x, y) + (f_x(x, y), f_y(x, y)) \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix}$$

$$+\frac{1}{2!}(\Delta x, \Delta y)\begin{pmatrix} f_{xx}(x,y) & f_{xy}(x,y) \\ f_{yx}(x,y) & f_{yy}(x,y) \end{pmatrix}\begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix},$$

$$f(x+\Delta x, y+\Delta y) \approx f(x,y) + (f_x(x,y), f_y(x,y))\begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix}$$

$$+\frac{1}{2!}(\Delta x, \Delta y)H(x,y)\begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix}.$$

Looking for a local minimum point with the initial $\bar{p} = (p_o, q_o)$, we have

$$\nabla f(x+\Delta x, y+\Delta y) = f_x(p_o, q_o) f_y(p_o, q_o) +$$

$$(x-p_o, y-q_o)\begin{pmatrix} f_{xx}(p_o,q_o) & f_{xy}(p_o,q_o) \\ f_{yx}(p_o,q_o) & f_{yy}(p_o,q_o) \end{pmatrix},$$

The equation above is the equation of the tangent line to the curve at $(\bar{p}_o, f(\bar{p}_o))$, so $(\bar{p}_1, 0)$ is the place where that tangent line intersects the x-axis. A graph can therefore give a good intuitive idea of why newton's method works at a well-chosen starting point and why it might diverge with a poorly-chosen starting point.

This expression above can be used to estimate the amount of offset $(\Delta x, \Delta y)$ needed to land closer to the root starting from an initial guess $\bar{p}_o$. Setting $\nabla f(\bar{p}_o) + (\bar{p_1} - \bar{p}_o)H(\bar{p}_o) = 0$ and solving for $\bar{p}_1$,gives

$$\bar{p}_1 = \bar{p}_o - \nabla f(\bar{p}_o)(H(\bar{p}_o))^{-1}.$$

This procedure can be unstable near a horizontal asymptote or a local extremum. However, with a good initial choice of the root's position, the algorithm can be

applied iteratively to obtain

$$\bar{p}_{n+1} = \bar{p}_n - \nabla f(\bar{p}_n)(H(\bar{p}_n))^{-1}.$$

When $|\bar{p}_{n+1} - \bar{p}_n| < \varepsilon$, we may get close to the local minimum point.

3 THE METHOD OF MOMENTS

3.1 INTRODUCTION OF THE METHOD OF MOMENTS

The basis of the method of moments is as follows: In random sampling, under generally benign assumptions, a sample statistic will converge in probability to some constant. This constant will, in turn, be a function of the unknown parameters of the distribution. To estimate p parameters, $\theta_1, \theta_2, \ldots, \theta_p$, we can compute p such statistics, $\bar{m_1}, \bar{m_2}, \ldots, \bar{m_p}$, whose probability limits are known functions of the parameters. These p moments are equated to the p functions, and the functions are inverted to express the parameters as functions of the moments [5].

Let $h(Z)$ be a vector of measurable functions respect to Z, $\Theta = (\theta_1, \theta_2, \ldots, \theta_p)^T$ be a vector of parameters characterizing the distribution of random variable Z.

$$h(Z) = \begin{pmatrix} h_1(Z) \\ h_2(Z) \\ \vdots \\ h_p(Z) \end{pmatrix}$$

Then let $X = (x_1, x_2, \cdots, x_n)^T$ where $x_1, x_2, \cdots, x_n$ are independent realizations of Z. We define

$$G(X, \Theta) = \begin{pmatrix} G_1(X, \Theta) \\ G_2(X, \Theta) \\ \vdots \\ G_p(X, \Theta) \end{pmatrix} = \begin{pmatrix} \frac{1}{n}\sum_{i=1}^{n} h_1(x_i) - E(h_1(Z)) \\ \frac{1}{n}\sum_{i=1}^{n} h_2(x_i) - E(h_2(Z)) \\ \vdots \\ \frac{1}{n}\sum_{i=1}^{n} h_p(x_i) - E(h_p(Z)) \end{pmatrix},$$

Suppose

$$G(X,\Theta) = \begin{pmatrix} G_1(X,\Theta) \\ G_2(X,\Theta) \\ \vdots \\ G_p(X,\Theta) \end{pmatrix} = 0$$

Then, we have the system equation:

$$L(X,\Theta) = \bar{G}^T(X,\Theta)\bar{G}(X,\Theta).$$

Since it is exactly identified cases, we could solve for the Θ.

3.2 EXAMPLES OF THE METHOD OF MOMENTS

Example 3.1. The density function of normal distribution $f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp(-\frac{(x-\mu)^2}{2\sigma^2})$, where $x_1, x_2, \cdots, x_n$ is a simple random sample from $N(\mu, \sigma^2)$. Here $\Theta = (\mu, \sigma^2)$. According to the *Moments in normal distribution* in the last chapter, we could see $E(Z) = \mu$ and $E(Z^2) = \mu^2 + \sigma^2$. Equating these with the sample moments, we have

$$\hat{\mu} = E(Z) = \frac{1}{n}\sum_{i=1}^{n} x_i = \bar{x},$$

$$\hat{\mu}^2 + \hat{\sigma}^2 = E(Z^2) = \frac{1}{n}\sum_{i=1}^{n} x_i^2.$$

Arranging terms, we get for $\hat{\sigma}^2$,

$$\hat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^{n} x_i^2 - \hat{\mu}^2 = \frac{1}{n}\sum_{i=1}^{n} x_i^2 - \bar{x}^2.$$

Example 3.2. The t-distribution. Suppose Z is a random variable following a t-distribution with v degrees of freedom. The density is then

$$f(x; v) = \frac{\Gamma(\frac{v+1}{2})}{\sqrt{\pi v}\Gamma(\frac{v}{2})}[1 + (\frac{x}{v})^2]^{-\frac{1}{2}(v+1)}.$$

With a simple random sample $x_1, x_2, \cdots, x_n$ from the above density, the MM estimator is obtained as follows. Provided that $v > 2$ the expected value of a t-distributed random variable is zero ($E(Z) = 0$) and $E(Z^2) = \frac{v}{v-2}$.

Equate this again with the second sample moment and solve for v to obtain

$$\hat{v} = \frac{2\hat{\mu}}{\hat{\mu} - 1},$$

provided that $s^2 > 1$, where $s^2 = \frac{1}{n}\sum_{i=1}^{n} x_i^2$.

4 THE GENERALIZED METHOD OF MOMENTS (GMM) ESTIMATOR

4.1 MINIMUM DISTANCE ESTIMATION

The preceding chapter has considered exactly identified cases. In *Example 3.1.*, we used 2 moments to estimate 2 parameters and examined the normal distribution, a two-parameter family. But sometime we have more moments than what we need, it would seem counterproductive to simply discard the additional information.

Minimum distance estimation is a statistical parameter estimate technique that selects model parameters that minimize a goodness-of-fit statistic[5].

Minimum distance estimator(MDE) is defined as follows: Let $\bar{m}_{n,l}$ denote a sample statistic based on n observations such that

$$p\lim \bar{m}_{n,l} = g_l(\theta_o), l = 1, \ldots, k,$$

where θ_o is a vector of $p \leq k$ parameters to be estimated. Arrange these moments and functions in $k \times 1$ vectors $\bar{m}_n = (\bar{m}_{n1}, \bar{m}_{n2}, \cdots, \bar{m}_{nk})^T$ and $g(\theta_o) = (g_1(\theta_o), g_2(\theta_o), \cdots, g_k(\theta_o))^T$ and further assume that the statistics are jointly asymptotically normally distributed with Asymptotic.Variance$[\bar{m}_n] = \frac{1}{n}S$, where S is the covariance matrix. Define the criterion function

$$L(X, \Theta) = \bar{G}^T(X, \Theta) W \bar{G}(X, \Theta)$$

for a positive definite **weighting matrix**, where $|W| = 1$. The minimum distance estimator is the $\hat{\Theta}_{MDE}$ that minimizes L.

4.2 INTRODUCTION OF GMM

Let $h(Z) = (h_1(Z), h_2(Z), \cdots, h_k(Z))^T$ be a vector of measurable functions with respect to the random variable Z, $\Theta = (\theta_1, \theta_2, \ldots, \theta_p)^T$ be a vector of parameters that characterize the distribution of random variable X. Then let $X = (x_1, x_2, \cdots, x_m)^T$ where $x_1, x_2, \cdots, x_m$ are independent realizations of Z. We define

$$G(X,\Theta) = \begin{pmatrix} G_1(X,\Theta) \\ G_2(X,\Theta) \\ \vdots \\ G_k(X,\Theta) \end{pmatrix} = \begin{pmatrix} \frac{1}{n}\sum_{i=1}^{n} h_1(x_i) - E(h_1(Z)) \\ \frac{1}{n}\sum_{i=1}^{n} h_2(x_i) - E(h_2(Z)) \\ \vdots \\ \frac{1}{n}\sum_{i=1}^{n} h_k(x_i) - E(h_k(Z)) \end{pmatrix}.$$

Suppose

$$G(X,\Theta) = \begin{pmatrix} G_1(X,\Theta) \\ G_2(X,\Theta) \\ \vdots \\ G_k(X,\Theta) \end{pmatrix} = 0.$$

Where $k > p$, $G(X,\Theta) = 0$ indicates there are more equations than parameters. Most of the time, there is no solution for the system $G(X,\Theta) = 0$, so instead we minimize the loss function

$$L(X,\Theta) = G^T(X,\Theta) W G(X,\Theta),$$

where W is a $k \times k$ positive definite matrix and $|W| = 1$ [6].

Examples 4.1.(*Example 3.2.* continued): Suppose $v > 4$, then we can find

$$E(Z^4) = \frac{3v^2}{(v-2)(v-4)}.$$

Thus the moment conditions implied by the model are

$$\left[\begin{pmatrix} x^2 - \frac{v}{v-2} \\ x^4 - \frac{3v^2}{(v-2)(v-4)} \end{pmatrix}\right] = 0.$$

Generally we cannot find a single value for v that satisfies exactly the corresponding sample moment conditions

$$G(X,v) = \begin{pmatrix} \frac{1}{n}\sum_{i=1}^{n}(x_i^2) - \frac{v}{v-2} \\ \frac{1}{n}\sum_{i=1}^{n}(x_i^4) - \frac{3v^2}{(v-2)(v-4)} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

However, we can choose v so that both equations are satisfied as closely as possible. The closeness is measured in terms of (weighted) squared errors, which minimizes criterion function

$$L(X,v) = G^T(X,v)WG(X,v),$$

where $X = (x_1, x_2, \cdots, x_n)^T$, and W is a suitable 2×2 weighting matrix. If we choose $W = I$, the identity matrix, the loss function is the average of two individual loss functions. Improved results are, however, achieved if the "less noisy" moment conditions are weighted more than the "noisier" ones. Matrix W serves for this purpose. It turns out that an optimal weighting matrix is the inverse of $S = G(X,v)G^T(X,v)$, which will be proved later.

• **Two-step Iteration**

(1) Determine the moment condition of the statistical distribution and choose the moments we need. For example:
We could choose $E(x) = \mu, E(x^2) = \mu^2 + \sigma^2$ and $E(x^3) = \mu^3 + 3\mu\sigma^2$ these three moments to estimate the parameter μ and σ^2.

(2) Choose initial weight matrix $W_1 = I$, with initial start point Θ_0.

(3) Using the predetermined W_1, we use the Newton-Raphson method to find

$$arg_\theta \min G^T(X,\Theta)WG(X,\Theta),$$

In the following, we will investigate the derivatives of $L(X,\Theta)$. Let $G = G(X,\Theta)$, where $\Theta = (\theta_1, \theta_2, \ldots, \theta_p)^T$, then

$$\nabla L = \frac{\partial(G^TWG)}{\partial\Theta} = \begin{pmatrix} 2G^TW\frac{\partial G}{\theta_1} \\ 2G^TW\frac{\partial G}{\theta_2} \\ \vdots \\ 2G^TW\frac{\partial G}{\theta_p} \end{pmatrix}$$

$$\begin{aligned} H &= \frac{\partial^2(G^TWG)}{\partial\Theta\partial\Theta'} \\ &= \begin{pmatrix} \frac{\partial^2(G^TWG)}{\partial\theta_1\partial\theta_1} & \frac{\partial^2(G^TWG)}{\partial\theta_1\partial\theta_2} & \cdots & \frac{\partial^2(G^TWG)}{\partial\theta_1\partial\theta_p} \\ \vdots & \ddots & \cdots & \vdots \\ \vdots & \cdots & \ddots & \vdots \\ \frac{\partial^2(G^TWG)}{\partial\theta_p\partial\theta_1} & \frac{\partial^2(G^TWG)}{\partial\theta_p\partial\theta_2} & \cdots & \frac{\partial^2(G^TWG)}{\partial\theta_p\partial\theta_p} \end{pmatrix}_{p\times p} = (H_{ij}), \end{aligned}$$

where,

$$H_{ij} = \frac{\partial^2 (G^T W G)}{\partial \theta_i \partial \theta_j} = 2 \left[\left(\frac{\partial G}{\partial \theta_i} \right)^T W \frac{\partial G}{\partial \theta_j} + \left(\frac{\partial^2 G}{\partial \theta_i \partial \theta_j} \right)^T W G \right]$$

Then, by the equation of the Newton-Raphson method

$$\hat{\Theta_1} = \hat{\Theta_0} - \nabla L(\hat{\Theta_0}) H^{-1}(\hat{\Theta_0}).$$

(4) Based on $\hat{\Theta}_1$, use the minimum distance estimation and find the optional weight matrix

$$W_2 = S^{-1}(\hat{\Theta_1}),$$

where $S = G(X, \hat{\Theta}_1) G^T(X, \hat{\Theta}_1)$.

(5) Replace W_1 by W_2, and go back to Step (3).

- **Simulations**

Normal Distribution

We obtained the moment condition from normal distribution and used the **Two-step Iteration** to get the estimator of parameters (μ, σ^2).

$$\begin{aligned}
Moment_1 : E(Z) &= \mu, \\
Moment_2 : E(Z^2) &= \mu^2 + \sigma^2, \\
Moment_3 : E(Z^3) &= \mu^3 + 3\mu\sigma^2, \\
Moment_4 : E(Z^4) &= \mu^4 + 6\mu^2\sigma^2 + 3\sigma^4, \\
Moment_5 : E(Z^5) &= \mu^5 + 10\mu^3\sigma^2 + 15\sigma^4, \\
Moment_6 : E(Z^6) &= \mu^6 + 15\mu^4\sigma^2 + 45\mu^2\sigma^4 + 15\sigma^6.
\end{aligned}$$

Sample	Moments	IP	$\hat{\mu}$	Var($\hat{\mu}$)	$\hat{sigma^2}$	Var($\hat{sigma^2}$)
2000	1,2	(1.7,2.7)	2.001716	0.004245306	8.989496	0.08003274
1500	1,2	(1.7,2.7)	2.001965	0.006185911	8.988922	0.1055325
500	1,2	(1.7,2.7)	2.002166	0.01764613	8.997564	0.3439522
100	1,2	(1.7,2.7)	1.998816	0.08535948	8.935932	1.645354
50	1,2	(1.7,2.7)	1.994258	0.1815228	8.809013	3.117058

Table 1: Normal(2,9) with Moments(1,2)

Sample	Moments	IP	$\hat{\mu}$	Var($\hat{\mu}$)	$\hat{sigma^2}$	Var($\hat{sigma^2}$)
2000	1,2,3	(1.7,2.7)	2.001809	0.004222465	8.962274	0.08272404
1500	1,2,3	(1.7,2.7)	2.001936	0.006156444	8.944948	0.1888715
500	1,2,3	(1.7,2.7)	2.001643	0.01800421	8.885725	0.5106328
100	1,2,3	(1.7,2.7)	1.998167	0.09183895	8.449222	2.217536
50	1,2,3	(1.7,2.7)	1.98493	0.2189923	8.019732	4.189232

Table 2: Normal(2,9) with Moments(1,2,3)

Sample	Moments	IP	$\hat{\mu}$	Var($\hat{\mu}$)	$\hat{sigma^2}$	Var($\hat{sigma^2}$)
2000	1,2,4	(1.7,2.7)	1.998952	0.004240296	8.959797	0.08302504
1500	1,2,4	(1.7,2.7)	1.998145	0.006205644	8.948113	0.1094095
500	1,2,4	(1.7,2.7)	1.997512	0.0321644	8.797329	1.047829
100	1,2,4	(1.7,2.7)	1.905949	0.3616054	7.379862	7.8151
50	1,2,4	(1.7,2.7)	1.806648	0.633377	6.422587	10.19754

Table 3: Normal(2,9) with Moments(1,2,4)

Sample	Moments	IP	$\hat{\mu}$	Var($\hat{\mu}$)	$\hat{sigma^2}$	Var($\hat{sigma^2}$)
2000	1,3,4	(1.7,2.7)	2.004758	0.004244875	8.944718	0.1196607
1500	1,3,4	(1.7,2.7)	2.005773	0.006187196	8.933276	0.1633366
500	1,3,4	(1.7,2.7)	2.011936	0.01793469	8.840983	0.5015864
100	1,3,4	(1.7,2.7)	1.970027	0.1340835	7.84763	4.676793
50	1,3,4	(1.7,2.7)	1.903754	0.260443	6.884535	6.888938

Table 4: Normal(2,9) with Moments(1,3,4)

Sample	Moments	IP	$\hat{\mu}$	Var($\hat{\mu}$)	$\hat{sigma^2}$	Var($\hat{sigma^2}$)
2000	1,2,3,4	(1.7,2.7)	2.001922	0.004250559	8.9257	0.08511245
1500	1,2,3,4	(1.7,2.7)	2.001871	0.00621309	8.903975	0.1135357
500	1,2,3,4	(1.7,2.7)	2.001798	0.01816149	8.767234	0.3702928
100	1,2,3,4	(1.7,2.7)	1.955995	0.121808	7.550488	4.217835
50	1,2,3,4	(1.7,2.7)	1.926255	0.2579954	6.459146	6.376687

Table 5: Normal(2,9) with Moments(1,2,3,4)

Gamma Distribution

We obtained the moment condition from gamma distribution and used the **Two-step Iteration** to get the estimator of parameter (α, θ).

$$\begin{aligned} Moment_1 : E(Z^1) &= \alpha\theta \\ Moment_2 : E(Z^2) &= \theta^2\alpha(\alpha+1) \\ Moment_{-1} : E(Z^{-1}) &= \frac{1}{\theta(\alpha-1)} \end{aligned}$$

Sample	Moments	IP	$\hat{k}$	Var($\hat{k}$)	$\hat{\theta}$	Var($\hat{\theta}$)
6000	1,2	(2.6,0.12)	1.222952	2.205972	0.04091055	0.002447722
4000	1,2	(2.6,0.12)	1.392796	2.237164	0.04681453	0.002527929

Table 6: Gamma(3,0.1) with Moment(1,2)

Sample	Moments	IP	$\hat{k}$	Var($\hat{k}$)	$\hat{\theta}$	Var($\hat{\theta}$)
5000	-1,1,2	(2.6,0.12)	3.008671	0.004251128	0.09975014	5.44E-06
4000	-1,1,2	(2.6,0.12)	3.010722	0.005169905	0.09970456	6.69E-06
3000	-1,1,2	(2.6,0.12)	3.014043	0.006527589	0.09961974	8.52E-06
2500	-1,1,2	(2.6,0.12)	3.016897	0.007754502	0.09952032	1.02E-05

Table 7: Gamma(3,0.1) with Moment(-1,1,2)

If we check the **Table 1** through **Table 7**, we could find out that no matter which moments we pick, the larger sample will always produce the nicer parameter estimator which are closer to the true parameter value. For example, in Table 1, the

true value of the parameters are $(\mu_0, \sigma_0^2) = (2, 9)$, and the parameter estimators with sample size 2000 are $(\hat{\mu}, \hat{\sigma}^2) = (2.001809, 8.962274)$, and the parameter estimators with sample size 50 are $(\hat{\mu}, \hat{\sigma}^2) = (1.98493, 8.019732)$.

Based on the simulations, we could find out that in some cases, infrequent with large samples but not so infrequent with small samples, the estimates given by the generalized method of moments are far from the true parameter space; it does not make sense to rely on them then. In the coming section, we will study the large sample properties of generalized method of moments (GMM) estimators.

Comparison in different moments with same sample

Sample	Moments	IP	μ	Var(μ)	σ^2	Var(σ^2)
2000	1,2,3	(1.7,2.7)	2.001809	0.004222465	8.962274	0.08272404
2000	1,2,4	(1.7,2.7)	1.998952	0.004240296	8.959797	0.08302504
2000	1,3,4	(1.7,2.7)	2.004758	0.004244875	8.944718	0.1196607
2000	1,2,3,4	(1.7,2.7)	2.001922	0.004250559	8.9257	0.08511245

Table 8: Normal(2,9) with Sample 2000

Sample	Moments	IP	μ	Var(μ)	σ^2	Var(σ^2)
1500	1,2,3	(1.7,2.7)	2.001936	0.006156444	8.944948	0.1888715
1500	1,2,4	(1.7,2.7)	1.998145	0.006205644	8.948113	0.1094095
1500	1,3,4	(1.7,2.7)	2.005773	0.006187196	8.933276	0.1633366
1500	1,2,3,4	(1.7,2.7)	2.001871	0.00621309	8.903975	0.1135357

Table 9: Normal(2,9) with Sample 1500

Sample	Moments	IP	μ	Var(μ)	σ^2	Var(σ^2)
500	1,2,3	(1.7,2.7)	2.001643	0.01800421	8.885725	0.5106328
500	1,2,4	(1.7,2.7)	1.997512	0.0321644	8.797329	1.047829
500	1,3,4	(1.7,2.7)	2.011936	0.01793469	8.840983	0.5015864
500	1,2,3,4	(1.7,2.7)	2.001798	0.01816149	8.767234	0.3702928

Table 10: Normal(2,9) with Sample 500

Sample	Moments	IP	μ	Var(μ)	σ^2	Var(σ^2)
100	1,2,3	(1.7,2.7)	1.998167	0.09183895	8.449222	2.217536
100	1,2,4	(1.7,2.7)	1.905949	0.3616054	7.379862	7.8151
100	1,3,4	(1.7,2.7)	1.970027	0.1340835	7.84763	4.676793
100	1,2,3,4	(1.7,2.7)	1.955995	0.121808	7.550488	4.217835

Table 11: Normal(2,9) with Sample 100

If we look at the **Table 8** through **Table 11**, we could see that the different moments will provide us different parameter estimators. Based on the results of the simulation, we could find out that we probably will not get the best estimator even if we choose as many moments as we could find. For the comparison moment(1,2,4) with moment(1,2,3,4) in sample size 2000, although the latter model has one more moment that the former model, the parameter from moment(1,2,4) is closer to the true parameter.

• **Weight Matrix**

The matrix W tells how much weight to put on each moment condition, as well as the product of two conditions.

Different W gives different estimators. There are two methods to prove the $W = S^{-1}$, where $S = G(X, \hat{\Theta}_1)G^T(X, \hat{\Theta}_1)$, is the optimal weight matrix in GMM.

Before the proof, we made the following assumptions about the model and these empirical moments:

Assumption(1) Convergence of the Empirical Moments:

The data generating process is assumed to meet the conditions for large numbers to apply so that we may assume that the empirical moments converge in probability to their expectation:

$$\bar{G}(X, \Theta_o) = \frac{1}{n}\Sigma_{i=1}^{n} G(x_i, \Theta_o) \xrightarrow{p} 0.$$

Assumption(2) Identification:

If Θ_1 and Θ_2 are two different parameters vectors, then there exist data sets that

$$\bar{G}(X, \Theta_1) \neq \bar{G}(X, \Theta_2).$$

Assumption(3) Asymptotic distributions of empirical moments:

We assume that the empirical moments obey a central limit theorem. This assumes that the moments have a finite asymptotic covariance matrix $\frac{1}{n}\Phi$. So that

$$\sqrt{n}\bar{G}(X, \Theta_o) \xrightarrow{d} N[0, \Phi].$$

There are also some laws and theorems will be used in the following proof:

Similar matrix: If $B = P^{-1}AP$, B is similar as A, where P is nonsingular matrix.

Congruence matrix: If $P^TAP = B$, B is congruence with A.

Sylvester's law of inertia: Two congruent symmetric matrices with real entries have the same numbers of positive, negative, and zero eigenvalues. Any nonsingular matrix S of the same size transforms A into another symmetric matrix. B of order n defined by the rules:$S^TAS = B$ and B is said to be congruent to A.

Theorem 4.1. Similar matrices have the same eigenvalues.

Proof. $B = P^{-1}AP$

$$\begin{aligned} |B - \Lambda I| &= |P^{-1}AP - P^{-1}\Lambda IP| \\ &= |P^{-1}||A - \Lambda I||P| \\ &= |A - \Lambda I| \end{aligned}$$

So,$|B - \Lambda I| = |A - \Lambda I|$ So, A and B have the same eigenvalues.

Theorem 4.2. If A and B are two positive definite matrices and $S = AB$. All the eigenvalues of S are positive.

Proof. We could always find a square matrix C which makes $C^TAC = I$, so $C^TA = C^{-1}$.

We multiply BC on both sides and then could get $C^TABC = C^{-1}BC$.

Hence C^TABC is similar as B, based on the properties of similar matrix, which that all the eigenvalues of C^TABC are positive.

Since C^TABC is congruence with AB, where $AB = S$. Based on the *Sylvester's law of inertia:*, we could conclude that the AB and C^TABC have the same number of positive eigenvalues which all the eigenvalues are positive.

Theorem 4.3. Let $W = S^{-1}$, where $S = G(X, \hat{\Theta}_1)G^T(X, \hat{\Theta}_1)$,$W_n$ the optimal weight matrix in GMM. *Proof.* Looking for the optimal W_{opt} to minimize $G'WG$:

Without loss of generality, we assume $|W| = |S^{-1}| = |S|^{-1}$,and

$$G^T W^{\frac{1}{2}} W^{\frac{1}{2}} G = G^T (W^{\frac{1}{2}}) W^{\frac{1}{2}} G = Y^T Y,$$

where $Y = W^{\frac{1}{2}} G$.

$$E(G(W^{\frac{1}{2}})^T W^{\frac{1}{2}} G) = E(Y^T Y) = tr(E(Y^T Y)) = trE(Y^T Y).$$

Since $G \sim N[0, S]$ and $Y = W^{\frac{1}{2}} G$, $Y \sim N[0, W^{\frac{1}{2}} S W^{\frac{1}{2}}]$, and $E(Y^T Y) = W^{\frac{1}{2}} S W^{\frac{1}{2}}$.
Thus, $tr(E(Y^T Y)) = tr(W^{\frac{1}{2}} S W^{\frac{1}{2}}) = tr(SW)$.

For minimum distance estimation, we want the weight matrix to produce the smallest variance. Since $tr(E(Y^T Y)) = tr(SW)$, we should minimize $tr(SW)$.

$$tr(SW) = \sum_{i=1}^{k} \lambda_i,$$

where λs are the eigenvalues of SW.
Since $|W| = |S^{-1}|$, so $\det |SW| = \det |A| = \prod_{i=1}^{k} \lambda_i = 1$,
where all λ_i are positive.

Because $\prod_{i=1}^{k} \lambda_i = 1$ and all λ_i are positive, it is easy to see that when $\lambda_i = \lambda_j = 1$ that we could minimize $\sum_{i=1}^{k} \lambda_i$.

Since all the $\lambda_i = 1$, so $\Lambda = I$,
Therefore, $SW = A = Q \Lambda Q^{-1}$, where $Q = (q_1, q_2, \ldots, q_k)$ and q_i is eigenvector basis.
So, $SW = I$, and $W = S^{-1}$.

Theorem 4.4. Asymptotic Distribution of the GMM Estimator:
Under the preceding assumptions,

$$\hat{\Theta}_{GMM} \xrightarrow{p} \Theta_o,$$

$$\sqrt{n}(\hat{\Theta}_{GMM} - \Theta_o) \xrightarrow{d} N[0, V_{GMM}],$$

and V_{GMM} is defined as:

$$V_{GMM} = \frac{1}{n}[\Gamma^T W \Gamma]^{-1} = \frac{1}{n}[\Gamma^T S^{-1} \Gamma]^{-1},$$

where Γ is the matrix of derivatives with j^{th} row equal to

$$\Gamma^j = p \lim \frac{\partial \bar{G}_j(\Theta)}{\partial \Theta^T}.$$

Proof. Suppose the GMM estimator is obtained by minimizing the criterion function:

$$L(X, \Theta) = \bar{G}^T(X, \Theta) W \bar{G}(X, \Theta),$$

where W is the weighting matrix used. It is established that $L(\Theta)$ converges to a value $L_o(\Theta)$, and $L_n(\Theta_o)$ converges to 0.

For any finite n,we know that:

$$0 \leq L(\hat{\Theta}_{GMM}) \leq L(\Theta_o)$$

That is, in the finite sample, $\hat{\Theta}_{GMM}$ actually minimizes the function, so the sample value of the loss is not larger than $\hat{\Theta}_{GMM}$, including the true parameters.
Since $L(\Theta_o) \xrightarrow{p} 0$ and $L(\hat{\Theta}_{GMM}) \leq L(\Theta_o)$, we could obtain

$$L(\hat{\Theta}_{GMM}) \longrightarrow 0,$$

and

$$\bar{G}(X, \hat{\Theta}_{GMM}) \xrightarrow{p} \bar{G}(X, \Theta_o).$$

The first order condition for GMM estimator is:

$$\frac{\partial L(\hat{\Theta}_{GMM})}{\partial \hat{\Theta}_{GMM}} = 2\bar{g}^T(X, \hat{\Theta}_{GMM})W\bar{G}(X, \hat{\Theta}_{GMM}) = 0,$$

where $\bar{g}(\Theta) = \frac{\partial \bar{G}(\Theta)}{\partial \Theta}$.

We expand the moments in a linear Taylor Series around the true value Θ_o:

$$\bar{G}(X, \hat{\Theta}_{GMM}) = \bar{G}(X, \Theta_o) + \bar{g}(X, \bar{\Theta})(\hat{\Theta}_{GMM} - \Theta_o),$$

where $\bar{\Theta}$ is a point between $\hat{\Theta}_{GMM}$ and Θ_o.

Then we multiply $\bar{g}^T(X, \hat{\Theta}_{GMM})W$ on the both sides of the equation and get:

$$\bar{g}^T(X, \hat{\Theta}_{GMM})W\bar{G}(X, \hat{\Theta}_{GMM}) = \bar{g}^T(X, \hat{\Theta}_{GMM})W\bar{G}(X, \Theta_o) +$$

$$\bar{g}^T(X, \hat{\Theta}_{GMM})W\bar{g}(X, \bar{\Theta})(\hat{\Theta}_{GMM} - \Theta_o).$$

Since $\bar{g}^T(X, \hat{\Theta}_{GMM})W\bar{G}(X, \hat{\Theta}_{GMM}) = 0$, we could obtain

$$\bar{g}^T(X, \hat{\Theta}_{GMM})W\bar{G}(X, \Theta_o) + \bar{g}^T(X, \hat{\Theta}_{GMM})W\bar{g}(X, \bar{\Theta})(\hat{\Theta}_{GMM} - \Theta_o) = 0.$$

Solving this equation for the estimation error and multiply by $\sqrt{n}$. It produces

$$\sqrt{n}(\hat{\Theta}_{GMM} - \Theta_o) = -[\bar{g}^T(X, \hat{\Theta}_{GMM})W\bar{g}(X, \bar{\Theta})]^{-1}\bar{g}^T(X, \hat{\Theta}_{GMM})W\sqrt{n}\bar{G}(X, \Theta_o),$$

Where $(\bar{g}^T(X, \hat{\Theta}_{GMM})W\bar{g}(X, \bar{\Theta}))$ is a nonsingular matrix since it is a positive definite matrix. By the strict continuity assumption, it must also be the case that

$$\bar{g}(X, \bar{\Theta}) \xrightarrow{p} \bar{g}(X, \Theta_o),$$

$$\bar{g}(X, \hat{\Theta}_{GMM}) \xrightarrow{p} \bar{g}(X, \Theta_o).$$

We have also assumed that the weighting matrix W_n converges to a matrix of constants W. Collecting terms , we find that the limiting distribution of the vector on the left side must be the same as that on the right side.

Since $\sqrt{n}\bar{G}(X, \Theta_o) \longrightarrow N[0, S]$, we could have the result

$$V_{GMM} = \frac{1}{n}[\bar{g}^T(X, \Theta_o)W\bar{g}(X, \Theta_o)]^{-1}\bar{g}^T(X, \Theta_o)WSW\bar{g}(X, \Theta_o)[\bar{g}^T(X, \Theta_o)W\bar{g}(X, \Theta_o)]^{-1}.$$

Based on the result from **Theorem 4.3.**, let $W_{opt} = S^{-1}$, the expression collapses to

$$V_{GMM} = \frac{1}{n}[\bar{g}^T(X, \Theta_o)S^{-1}\bar{g}(X, \Theta_o)]^{-1} = \frac{1}{n}[\Gamma^T W\Gamma]^{-1} = \frac{1}{n}[\Gamma^T S^{-1}\Gamma]^{-1},$$

where $\Gamma^j = p\lim \frac{\partial \bar{G}_j(\Theta)}{\partial \Theta^T}$.

4.4 MOMENTS CONDITION IN GMM

Using many moment conditions can improve efficiency but makes the usual GMM inferences inaccurate. Two step GMM is biased. Generalized method of moments (GMM) provides an attractive estimation methodology that has been widely used in empirical research and is well suited to situations where economic information is given in terms of moment conditions. The approach has several well known advantages including an easily implemented asymptotic theory[8].

On the other hand, GMM asymptotic distributions depend on regularity conditions that are not always satisfied and which can affect finite sample performance adversely, under such weak moment conditions, GMM estimates are not consistent but converge weakly to a nondegenerate limit distribution.

Despite the impact of the weak moment condition and noisy moments[9], the newton-raphson method also will influence the result of the simulations. The following figures are the result from simulations with moments(1,2,3,4,5) and moments(1,2,3,4,5,6).

Moments:

$$\begin{aligned} E(x) &= \mu \\ E(x^2) &= \mu^2 + \sigma^2 \\ E(x^3) &= \mu^3 + 3\mu\sigma^2 \\ E(x^4) &= \mu^4 + 6\mu^2\sigma^2 + 3\sigma^4 \\ E(x^5) &= \mu^5 + 10\mu^3\sigma^2 + 15\sigma^4 \end{aligned}$$

Sample	Moments	IP	μ	Var(μ)	σ^2	Var(σ^2)
2000	1,2,3,4,5	(1.7,2.7)	1.638546	0.5438535	4.241407	15.02679
1000	1,2,3,4,5	(1.7,2.7)	1.572413	0.1729262	4.414884	14.31249
500	1,2,3,4,5	(1.7,2.7)	1.537928	0.2077437	4.353432	13.79348
50	1,2,3,4,5	(1.7,2.7)	1.438335	2.656404	3.146302	9.023368

Table 12: Normal(2,9) with Moments(1,2,3,4,5)

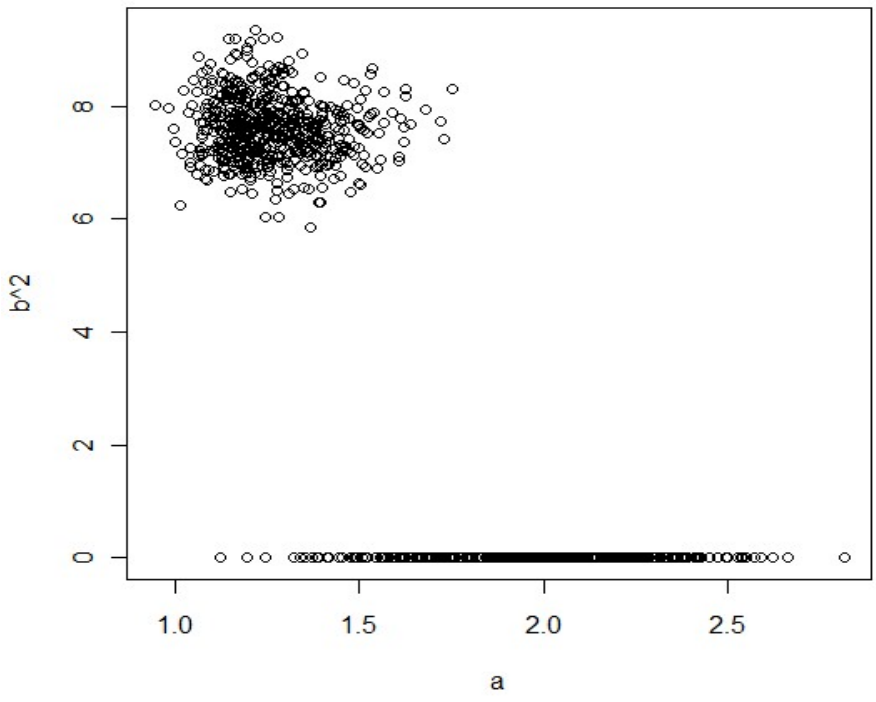

Figure 3: Normal Distribution 5 moments

Moments:

$$
\begin{aligned}
E(x) &= \mu \\
E(x^2) &= \mu^2 + \sigma^2 \\
E(x^3) &= \mu^3 + 3\mu\sigma^2 \\
E(x^4) &= \mu^4 + 6\mu^2\sigma^2 + 3\sigma^4 \\
E(x^5) &= \mu^5 + 10\mu^3\sigma^2 + 15\sigma^4 \\
E(x^6) &= \mu^6 + 15\mu^4\sigma^2 + 45\mu^2\sigma^4 + 15\sigma^6
\end{aligned}
$$

Sample	Moments	IP	μ	Var(μ)	σ^2	Var(σ^2)
2000	1,2,3,4,5,6	(1.7,2.7)	1.275399	0.2697881	5.648288	12.53805
1000	1,2,3,4,5,6	(1.7,2.7)	1.266091	0.3103147	4.991871	13.79390
500	1,2,3,4,5,6	(1.7,2.7)	1.216009	0.4325025	4.624603	14.07100
50	1,2,3,4,5,6	(1.7,2.7)	1.127662	3.236429	2.743732	8.069324

Table 13: Normal(2,9) with Moments(1,2,3,4,5,6)

Since the moment$_5$ ($E(x^5) = \mu^5 + 10\mu^3\sigma^2 + 15\sigma^4$) and moment$_6$ ($E(x^6) = \mu^6 + 15\mu^4\sigma^2 + 45\mu^2\sigma^4 + 15\sigma^6$) are very sensitive when we trying use Newton-Raphson method to find the solution. Almost all the solutions converge to the origin as the local minimum. We could see the estimates by the GMM are not necessarily sufficient statistics when we pick some the higher moments, sometimes fail to take into account all relevant information in the sample.

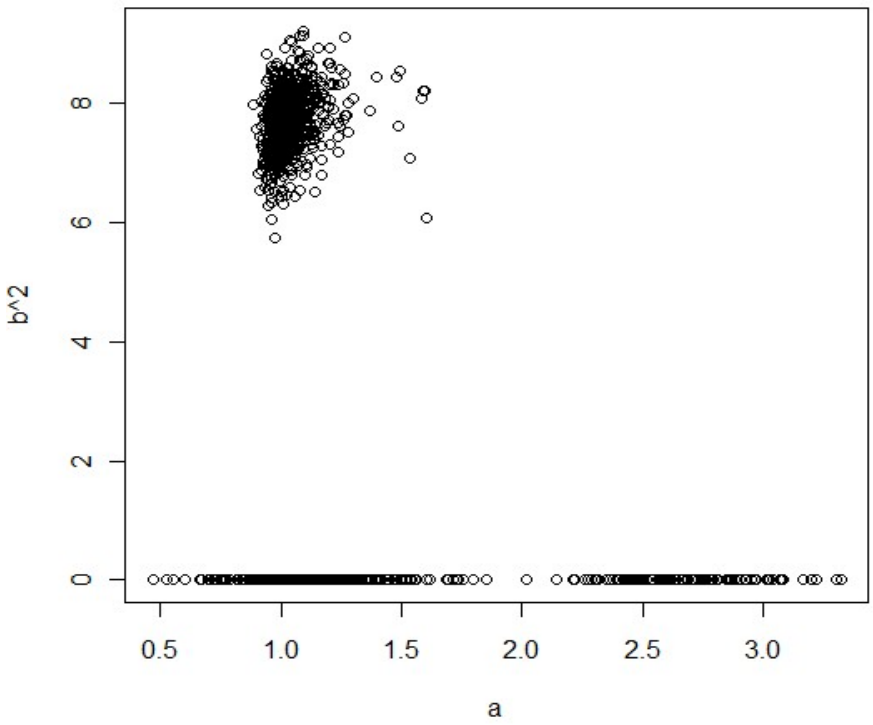

Figure 4: Normal Distribution 6 moments

5 SUMMARY AND CONCLUSION

The main purpose of the present study was to apply generalized method of moment parameter estimation (GMM) to the exponential distribution family. This was done by:

(1) Point out the difference between method of moment(MM) and the generalized method of moment (GMM);

(2) Show how to find the optimal weight matrix W and prove $W_{opt} = S^{-1}$;

(3) Show how the different moments effect the parameter estimation;

(4) Show how the Newton-Raphson method influences the process of two-step Iteration.

REFERENCES

[1] Paul G. Hoel, "Introduction to Mathematical Statistics",*John Wiley and Sons, INC.*

[2] Jan R. Magnus, "Matrix Differential Calculus with Applications in Statistics and Econometrics",*John Wiley and Sons, INC*, 1988

[3] E. Malinvaud, "Statistical Methods of Econometrics",*North-Holland Publishing Company*, 1980

[4] Peter Borwein, "Polynomials and polynomial inequalities",*New York : Springer Verlag*, 1995

[5] William H. Greene,"Econometric Analysis",*Prentice Hall.*

[6] Hansen, Lars Peter, 1982. "Large Sample Properties of Generalized Method of Moments Estimators",*Econometrica*, vol. 50(4), pages 1029-54, July.

[7] Alastair R. Hall, "Generalized Method of Moments (Advanced Texts in Econometrics)",*Oxford University Press.*

[8] Chirok Han, Peter C. B. Phillips,"GMM with Many Moment Conditions",*Econometrica*, 74(1) , 147-192, 2006.

[9] Whitney K. Newey, Frank Windmeijer,"GMM with Many Weak Moment Conditions",*Econometrica.*

[10] Darrell A. Turkington, "Matrix Calculus and Zero-One Matrices", *Cambridge University Press*

APPENDIX

A. R CODE FOR FOUR MOMENTS IN NORMAL DISTRIBUTION

```
number<-1000
a<-rep(0,number)
b<-rep(0,number)

   p<-1.7
   lamda<-2.7
   W<-diag(rep(1,4))
   max.count<-10^3;
   sample<-100;
   epsilon<-10^(-3)

for(k in 1:number)

{
   y<-rnorm(sample,2,3)
   y1<-mean(y)
   y2<-mean(y^2)
   y3<-mean(y^3)
   y4<-mean(y^4)
   set.seed(k)

   beta.step<-function(p,lamda,W)
   {
     flag<-1
     count<-0;

     while(flag==1)
     {
      f<-c(y1-p,y2-p^2-lamda^2,y3-p^3-3*p*lamda^2,
      y4-p^4-6*p^2*lamda^2-3*lamda^4)

      D.mat<-function(vec, var)
       {
        m<-length(var);
        n<-length(vec)
        d.vec<-NULL;

          for(i in 1:m)
```

```
     for(j in 1:n)
      {
       d.vec<-c(d.vec,D(vec[[j]],var[i]));
      }
     return(d.vec);
  }

Y1<-expression(y1-p)
Y2<-expression(y2-p^2-lamda^2)
Y3<-expression(y3-p^3-3*p*lamda^2)
Y4<-expression(y4-p^4-6*p^2*lamda^2-3*lamda^4)

vec<-c(Y1,Y2,Y3,Y4)
varp<-"p"
varlamda<-"lamda"

D1p.vec<-D.mat(vec,varp);

  fp1<-function(p,lamda){}
  body(fp1)<-D1p.vec[1]

  fp2<-function(p,lamda){}
  body(fp2)<-D1p.vec[2]

  fp3<-function(p,lamda){}
  body(fp3)<-D1p.vec[3]

  fp4<-function(p,lamda){}
  body(fp4)<-D1p.vec[4]

  fp<-c(fp1(p,lamda),fp2(p,lamda),\\
  fp3(p,lamda),fp4(p,lamda))

D1lamda.vec<-D.mat(vec,varlamda)

  flamda1<-function(p,lamda){}
  body(flamda1)<-D1lamda.vec[1]

  flamda2<-function(p,lamda){}
  body(flamda2)<-D1lamda.vec[2]

  flamda3<-function(p,lamda){}
  body(flamda3)<-D1lamda.vec[3]
```

```
  flamda4<-function(p,lamda){}
  body(flamda4)<-D1lamda.vec[4]

  flamda<-c(flamda1(p,lamda),flamda2(p,lamda),
  flamda3(p,lamda),flamda4(p,lamda))

D1pp.vec<-D.mat(D1p.vec,varp)

  fpp1<-function(p,lamda){}
  body(fpp1)<-D1pp.vec[1]

  fpp2<-function(p,lamda){}
  body(fpp2)<-D1pp.vec[2]

  fpp3<-function(p,lamda){}
  body(fpp3)<-D1pp.vec[3]

  fpp4<-function(p,lamda){}
  body(fpp4)<-D1pp.vec[4]

  fpp<-c(fpp1(p,lamda),fpp2(p,lamda),\\
  fpp3(p,lamda),fpp4(p,lamda))

D1plamda.vec<-D.mat(D1p.vec,varlamda)

  fplamda1<-function(p,lamda){}
  body(fplamda1)<-D1plamda.vec[1]

  fplamda2<-function(p,lamda){}
  body(fplamda2)<-D1plamda.vec[2]

  fplamda3<-function(p,lamda){}
  body(fplamda3)<-D1plamda.vec[3]

  fplamda4<-function(p,lamda){}
  body(fplamda4)<-D1plamda.vec[4]

  fplamda<-c(fplamda1(p,lamda),fplamda2(p,lamda),
  fplamda3(p,lamda),fplamda4(p,lamda))

D1lamdalamda.vec<-D.mat(D1lamda.vec,varlamda)

  flamdalamda1<-function(p,lamda){}
  body(flamdalamda1)<-D1lamdalamda.vec[1]
```

```
      flamdalamda2<-function(p,lamda){}
      body(flamdalamda2)<-D1lamdalamda.vec[2]

      flamdalamda3<-function(p,lamda){}
      body(flamdalamda3)<-D1lamdalamda.vec[3]

      flamdalamda4<-function(p,lamda){}
      body(flamdalamda4)<-D1lamdalamda.vec[4]

      flamdalamda<-c(flamdalamda1(p,lamda),
      flamdalamda2(p,lamda),\\
      flamdalamda3(p,lamda),flamdalamda4(p,lamda))

   F<-2*c(t(fp)%*%W%*%f,t(flamda)%*%W%*%f)

   H11<-t(f)%*%W%*%fpp+t(fp)%*%W%*%fp;
   H12<-t(f)%*%W%*%fplamda+t(fp)%*%W%*%flamda;
   H22<-t(f)%*%W%*%flamdalamda+t(flamda)%*%W%*%flamda;
   H1<-2*rbind(c(H11,H12),c(H12,H22));

   E<-solve(H1)%*%F
   p1<-p-E[1]
   lamda1<-lamda-E[2]

   if(((p-p1)^2+(lamda-lamda1)^2)<epsilon) {flag<-0; \\
   print("#####success #####");};

   p<-p1;
   lamda<-lamda1;
   count<-count+1;
   print(c(p,lamda));

if(count>max.count){ print("do not coverge"); break();};

  }
   return(list(p,lamda));
}


w.step<-function(pf,lamdaf)
{
 w<-matrix(0,nrow=4,ncol=4)
   for(i in 1:sample)
```

```
      {
       D<-c(y[i]-pf,(y[i])^2-pf^2-lamdaf^2,\\
       (y[i])^3-pf^3-3*pf*lamdaf^2,\\
       (y[i])^4-pf^4-6*pf^2*lamdaf^2-3*lamdaf^4)
       w<-w+D%*%t(D)
      }
    W1<-solve(w/sample)
   }

   flag.main<-1;
   result<-beta.step(p,lamda,W)
   p1<-result[[1]]
   lamda1<-result[[2]]

 while(flag.main==1)
  {
  W2<-w.step(p1,lamda1)
  result2<-beta.step(p1,lamda1,W2);
  p2<-result2[[1]]
  lamda2<-result2[[2]]

  if((p1-p2)^2+(lamda1-lamda2)^2<epsilon) \\
  { print("#####BINGO ####");flag.main=0};

  p1<-p2;
  lamda1<-lamda2;
  print(c(p1,lamda1))
  }

a[k]<-p1
b[k]<-lamda1

}

plot(a,b^2)
mean(a)
var(a)

mean(b^2)
var(b^2)
```

```
number<-1000
a<-rep(0,number)
b<-rep(0,number)

   p<-1.7
   lamda<-2.7
   W<-diag(rep(1,3))
   max.count<-10^3;
   sample<-100;
   epsilon<-10^(-3)

for(k in 1:number)

{
   y<-rgamma(sample,3,10)
   y1<-mean(y)
   y2<-mean(y^2)
   y3<-mean(1/y)
   set.seed(k)

   beta.step<-function(p,lamda,W)
   {
     flag<-1
     count<-0;

     while(flag==1)
     {
      f<-c(y1-p*lamda,y2-p*(p+1)*lamda^2,y3-1/(lamda*(p-1)))

      D.mat<-function(vec, var)
       {
        m<-length(var);
        n<-length(vec)
        d.vec<-NULL;

          for(i in 1:m)
           for(j in 1:n)
            {
             d.vec<-c(d.vec,D(vec[[j]],var[i]));
            }
           return(d.vec);
```

```
  }

Y1<-expression(y1-p*lamda)
Y2<-expression(y2-p*(p+1)*lamda^2)
Y3<-expression(y3-1/(lamda*(p-1)))

vec<-c(Y1,Y2,Y3)
varp<-"p"
varlamda<-"lamda"

D1p.vec<-D.mat(vec,varp);

  fp1<-function(p,lamda){}
  body(fp1)<-D1p.vec[1]

  fp2<-function(p,lamda){}
  body(fp2)<-D1p.vec[2]

  fp3<-function(p,lamda){}
  body(fp3)<-D1p.vec[3]

  fp<-c(fp1(p,lamda),fp2(p,lamda),fp3(p,lamda))

D1lamda.vec<-D.mat(vec,varlamda)

  flamda1<-function(p,lamda){}
  body(flamda1)<-D1lamda.vec[1]

  flamda2<-function(p,lamda){}
  body(flamda2)<-D1lamda.vec[2]

  flamda3<-function(p,lamda){}
  body(flamda3)<-D1lamda.vec[3]

  flamda<-c(flamda1(p,lamda),\\
  flamda2(p,lamda),flamda3(p,lamda))

D1pp.vec<-D.mat(D1p.vec,varp)

  fpp1<-function(p,lamda){}
  body(fpp1)<-D1pp.vec[1]

  fpp2<-function(p,lamda){}
  body(fpp2)<-D1pp.vec[2]
```

```
  fpp3<-function(p,lamda){}
  body(fpp3)<-D1pp.vec[3]

  fpp<-c(fpp1(p,lamda),fpp2(p,lamda),fpp3(p,lamda))

D1plamda.vec<-D.mat(D1p.vec,varlamda)

  fplamda1<-function(p,lamda){}
  body(fplamda1)<-D1plamda.vec[1]

  fplamda2<-function(p,lamda){}
  body(fplamda2)<-D1plamda.vec[2]

  fplamda3<-function(p,lamda){}
  body(fplamda3)<-D1plamda.vec[3]

  fplamda<-c(fplamda1(p,lamda),\\
  fplamda2(p,lamda),fplamda3(p,lamda))

D1lamdalamda.vec<-D.mat(D1lamda.vec,varlamda)

  flamdalamda1<-function(p,lamda){}
  body(flamdalamda1)<-D1lamdalamda.vec[1]

  flamdalamda2<-function(p,lamda){}
  body(flamdalamda2)<-D1lamdalamda.vec[2]

  flamdalamda3<-function(p,lamda){}
  body(flamdalamda3)<-D1lamdalamda.vec[3]

  flamdalamda<-c(flamdalamda1(p,lamda),
  flamdalamda2(p,lamda),flamdalamda3(p,lamda))

F<-2*c(t(fp)%*%W%*%f,t(flamda)%*%W%*%f)

H11<-t(f)%*%W%*%fpp+t(fp)%*%W%*%fp;
H12<-t(f)%*%W%*%fplamda+t(fp)%*%W%*%flamda;
H22<-t(f)%*%W%*%flamdalamda+t(flamda)%*%W%*%flamda;
H1<-2*rbind(c(H11,H12),c(H12,H22));

E<-solve(H1)%*%F
p1<-p-E[1]
lamda1<-lamda-E[2]
```

```
    if(((p-p1)^2+(lamda-lamda1)^2)<epsilon) {flag<-0; \\
    print("#### success ########");};

    p<-p1;
    lamda<-lamda1;
    count<-count+1;
    print(c(p,lamda));

  if(count>max.count){ print("do not coverge"); break();};

    }
    return(list(p,lamda));
  }


  w.step<-function(pf,lamdaf)
  {
   w<-matrix(0,nrow=3,ncol=3)
     for(i in 1:sample)
     {
      D<-c(y[i]-pf*lamdaf,\\
      (y[i])^2-pf*(pf+1)*lamdaf^2,1/y[i]-1/(lamdaf*(pf-1)))
      w<-w+D%*%t(D)
     }
   W1<-solve(w/sample)
  }

  flag.main<-1;
  result<-beta.step(p,lamda,W)
  p1<-result[[1]]
  lamda1<-result[[2]]

while(flag.main==1)
 {
 W2<-w.step(p1,lamda1)
 result2<-beta.step(p1,lamda1,W2);
 p2<-result2[[1]]
 lamda2<-result2[[2]]

 if((p1-p2)^2+(lamda1-lamda2)^2<epsilon)\\
 { print("###### BINGO #######")
 ;flag.main=0};
```

```
  p1<-p2;
  lamda1<-lamda2;
  print(c(p1,lamda1))
  }

a[k]<-p1
b[k]<-lamda1

}

plot(a,b)
mean(a)
var(a)

mean(b)
var(b)
```

```
number<-1000
a<-rep(0,number)
b<-rep(0,number)

   p<-1.7
   lamda<-2.7
   W<-diag(rep(1,3))
   max.count<-10^3;
   sample<-100;
   epsilon<-10^(-6)

for(k in 1:number)

{
   y<-rnorm(sample,2,3)
   y1<-mean(y)
   y2<-mean(y^2)
   y3<-mean(y^3)
   set.seed(k)

   beta.step<-function(p,lamda,W)
   {
     flag<-1
     count<-0;

     while(flag==1)
     {
      f<-c(y1-p,y2-p^2-lamda^2,y3-p^3-3*p*lamda^2)

      D.mat<-function(vec, var)
       {
        m<-length(var);
        n<-length(vec)
        d.vec<-NULL;

          for(i in 1:m)
           for(j in 1:n)
            {
             d.vec<-c(d.vec,D(vec[[j]],var[i]));
            }
           return(d.vec);
```

```
  }

Y1<-expression(y1-p)
Y2<-expression(y2-p^2-lamda^2)
Y3<-expression(y3-p^3-3*p*lamda^2)

vec<-c(Y1,Y2,Y3)
varp<-"p"
varlamda<-"lamda"

D1p.vec<-D.mat(vec,varp);

  fp1<-function(p,lamda){}
  body(fp1)<-D1p.vec[1]

  fp2<-function(p,lamda){}
  body(fp2)<-D1p.vec[2]

  fp3<-function(p,lamda){}
  body(fp3)<-D1p.vec[3]

  fp<-c(fp1(p,lamda),fp2(p,lamda),fp3(p,lamda))

D1lamda.vec<-D.mat(vec,varlamda)

  flamda1<-function(p,lamda){}
  body(flamda1)<-D1lamda.vec[1]

  flamda2<-function(p,lamda){}
  body(flamda2)<-D1lamda.vec[2]

  flamda3<-function(p,lamda){}
  body(flamda3)<-D1lamda.vec[3]

  flamda<-c(flamda1(p,lamda),flamda2(p,lamda),
  flamda3(p,lamda))

D1pp.vec<-D.mat(D1p.vec,varp)

  fpp1<-function(p,lamda){}
  body(fpp1)<-D1pp.vec[1]

  fpp2<-function(p,lamda){}
  body(fpp2)<-D1pp.vec[2]
```

```
  fpp3<-function(p,lamda){}
  body(fpp3)<-D1pp.vec[3]

  fpp<-c(fpp1(p,lamda),fpp2(p,lamda),fpp3(p,lamda))

D1plamda.vec<-D.mat(D1p.vec,varlamda)

  fplamda1<-function(p,lamda){}
  body(fplamda1)<-D1plamda.vec[1]

  fplamda2<-function(p,lamda){}
  body(fplamda2)<-D1plamda.vec[2]

  fplamda3<-function(p,lamda){}
  body(fplamda3)<-D1plamda.vec[3]

  fplamda<-c(fplamda1(p,lamda),fplamda2(p,lamda),
  fplamda3(p,lamda))

D1lamdalamda.vec<-D.mat(D1lamda.vec,varlamda)

  flamdalamda1<-function(p,lamda){}
  body(flamdalamda1)<-D1lamdalamda.vec[1]

  flamdalamda2<-function(p,lamda){}
  body(flamdalamda2)<-D1lamdalamda.vec[2]

  flamdalamda3<-function(p,lamda){}
  body(flamdalamda3)<-D1lamdalamda.vec[3]

  flamdalamda<-c(flamdalamda1(p,lamda),flamdalamda2(p,lamda),
  flamdalamda3(p,lamda))

F<-2*c(t(fp)%*%W%*%f,t(flamda)%*%W%*%f)

H11<-t(f)%*%W%*%fpp+t(fp)%*%W%*%fp;
H12<-t(f)%*%W%*%fplamda+t(fp)%*%W%*%flamda;
H22<-t(f)%*%W%*%flamdalamda+t(flamda)%*%W%*%flamda;
H1<-2*rbind(c(H11,H12),c(H12,H22));

E<-solve(H1)%*%F
p1<-p-E[1]
lamda1<-lamda-E[2]
```

```
    if(((p-p1)^2+(lamda-lamda1)^2)<epsilon)
    {flag<-0; print("#### success ####");};

    p<-p1;
    lamda<-lamda1;
    count<-count+1;
    print(c(p,lamda));

  if(count>max.count){ print("do not coverge"); break();};

    }
    return(list(p,lamda));
  }


  w.step<-function(pf,lamdaf)
  {
   w<-matrix(0,nrow=3,ncol=3)
    for(i in 1:sample)
    {
     D<-c(y[i]-pf,(y[i])^2-pf^2-lamdaf^2,
     (y[i])^3-pf^3-3*pf*lamdaf^2)
     w<-w+D%*%t(D)
    }
   W1<-solve(w/sample)
  }

  flag.main<-1;
  result<-beta.step(p,lamda,W)
  p1<-result[[1]]
  lamda1<-result[[2]]

while(flag.main==1)
 {
 W2<-w.step(p1,lamda1)
 result2<-beta.step(p1,lamda1,W2);
 p2<-result2[[1]]
 lamda2<-result2[[2]]

 if((p1-p2)^2+(lamda1-lamda2)^2<epsilon)
 { print("##### BINGO ######");flag.main=0};

 p1<-p2;
```

```
  lamda1<-lamda2;
  print(c(p1,lamda1))
  }

a[k]<-p1
b[k]<-lamda1

}

plot(a,b)
mean(a)
var(a)

mean(b)
var(b)
```

```
number<-1000
a<-rep(0,number)
b<-rep(0,number)

   p<-1.7
   lamda<-2.7
   W<-diag(rep(1,3))
   max.count<-10^3;
   sample<-100;
   epsilon<-10^(-3)

for(k in 1:number)

{
   y<-rnorm(sample,2,3)
   y1<-mean(y)
   y2<-mean(y^2)
   y3<-mean(y^4)
   set.seed(k)

   beta.step<-function(p,lamda,W)
   {
     flag<-1
     count<-0;

     while(flag==1)
     {
      f<-c(y1-p,y2-p^2-lamda^2,y3-p^4-6*p^2*lamda^2-3*lamda^4)

      D.mat<-function(vec, var)
       {
        m<-length(var);
        n<-length(vec)
        d.vec<-NULL;

          for(i in 1:m)
           for(j in 1:n)
            {
             d.vec<-c(d.vec,D(vec[[j]],var[i]));
            }
           return(d.vec);
```

```
  }

Y1<-expression(y1-p)
Y2<-expression(y2-p^2-lamda^2)
Y3<-expression(y3-p^4-6*p^2*lamda^2-3*lamda^4)

vec<-c(Y1,Y2,Y3)
varp<-"p"
varlamda<-"lamda"

D1p.vec<-D.mat(vec,varp);

  fp1<-function(p,lamda){}
  body(fp1)<-D1p.vec[1]

  fp2<-function(p,lamda){}
  body(fp2)<-D1p.vec[2]

  fp3<-function(p,lamda){}
  body(fp3)<-D1p.vec[3]

  fp<-c(fp1(p,lamda),fp2(p,lamda),fp3(p,lamda))

D1lamda.vec<-D.mat(vec,varlamda)

  flamda1<-function(p,lamda){}
  body(flamda1)<-D1lamda.vec[1]

  flamda2<-function(p,lamda){}
  body(flamda2)<-D1lamda.vec[2]

  flamda3<-function(p,lamda){}
  body(flamda3)<-D1lamda.vec[3]

  flamda<-c(flamda1(p,lamda),flamda2(p,lamda),
  flamda3(p,lamda))

D1pp.vec<-D.mat(D1p.vec,varp)

  fpp1<-function(p,lamda){}
  body(fpp1)<-D1pp.vec[1]

  fpp2<-function(p,lamda){}
  body(fpp2)<-D1pp.vec[2]
```

```
  fpp3<-function(p,lamda){}
  body(fpp3)<-D1pp.vec[3]

  fpp<-c(fpp1(p,lamda),fpp2(p,lamda),fpp3(p,lamda))

D1plamda.vec<-D.mat(D1p.vec,varlamda)

  fplamda1<-function(p,lamda){}
  body(fplamda1)<-D1plamda.vec[1]

  fplamda2<-function(p,lamda){}
  body(fplamda2)<-D1plamda.vec[2]

  fplamda3<-function(p,lamda){}
  body(fplamda3)<-D1plamda.vec[3]

  fplamda<-c(fplamda1(p,lamda),fplamda2(p,lamda),
  fplamda3(p,lamda))

D1lamdalamda.vec<-D.mat(D1lamda.vec,varlamda)

  flamdalamda1<-function(p,lamda){}
  body(flamdalamda1)<-D1lamdalamda.vec[1]

  flamdalamda2<-function(p,lamda){}
  body(flamdalamda2)<-D1lamdalamda.vec[2]

  flamdalamda3<-function(p,lamda){}
  body(flamdalamda3)<-D1lamdalamda.vec[3]

  flamdalamda<-c(flamdalamda1(p,lamda),
  flamdalamda2(p,lamda),flamdalamda3(p,lamda))

F<-2*c(t(fp)%*%W%*%f,t(flamda)%*%W%*%f)

H11<-t(f)%*%W%*%fpp+t(fp)%*%W%*%fp;
H12<-t(f)%*%W%*%fplamda+t(fp)%*%W%*%flamda;
H22<-t(f)%*%W%*%flamdalamda+t(flamda)%*%W%*%flamda;
H1<-2*rbind(c(H11,H12),c(H12,H22));

E<-solve(H1)%*%F
p1<-p-E[1]
lamda1<-lamda-E[2]
```

```
    if(((p-p1)^2+(lamda-lamda1)^2)<epsilon)
    {flag<-0; print("#### success ####");};

    p<-p1;
    lamda<-lamda1;
    count<-count+1;
    print(c(p,lamda));

  if(count>max.count){ print("do not coverge"); break();};

    }
    return(list(p,lamda));
  }


  w.step<-function(pf,lamdaf)
  {
   w<-matrix(0,nrow=3,ncol=3)
     for(i in 1:sample)
     {
      D<-c(y[i]-pf,(y[i])^2-pf^2-lamdaf^2,
      (y[i])^4-pf^4-6*pf^2*lamdaf^2-3*lamdaf^4)
      w<-w+D%*%t(D)
     }
   W1<-solve(w/sample)
  }

  flag.main<-1;
  result<-beta.step(p,lamda,W)
  p1<-result[[1]]
  lamda1<-result[[2]]

while(flag.main==1)
 {
 W2<-w.step(p1,lamda1)
 result2<-beta.step(p1,lamda1,W2);
 p2<-result2[[1]]
 lamda2<-result2[[2]]

 if((p1-p2)^2+(lamda1-lamda2)^2<epsilon)
  { print("##### BINGO ####");flag.main=0};

 p1<-p2;
```

```
  lamda1<-lamda2;
  print(c(p1,lamda1))
  }

a[k]<-p1
b[k]<-lamda1

}

plot(a,b^2)
mean(a)
var(a)

mean(b^2)
var(b^2)
```

```
number<-1000
a<-rep(0,number)
b<-rep(0,number)
number<-1000
a<-rep(0,number)
b<-rep(0,number)

   p<-1.7
   lamda<-2.7
   W<-diag(rep(1,3))
   max.count<-10^3;
   sample<-100;
   epsilon<-10^(-3)

for(k in 1:number)

{
   y<-rnorm(sample,2,3)
   y1<-mean(y)
   y2<-mean(y^3)
   y3<-mean(y^4)
   set.seed(k)

   beta.step<-function(p,lamda,W)
   {
     flag<-1
     count<-0;

     while(flag==1)
     {
      f<-c(y1-p,y2-p^3-3*p*lamda^2,y3-p^4-6*p^2*lamda^2-3*lamda^4)

      D.mat<-function(vec, var)
       {
        m<-length(var);
        n<-length(vec)
        d.vec<-NULL;

          for(i in 1:m)
           for(j in 1:n)
            {
```

```
       d.vec<-c(d.vec,D(vec[[j]],var[i]));
      }
     return(d.vec);
  }

Y1<-expression(y1-p)
Y2<-expression(y2-p^3-3*p*lamda^2)
Y3<-expression(y3-p^4-6*p^2*lamda^2-3*lamda^4)

vec<-c(Y1,Y2,Y3)
varp<-"p"
varlamda<-"lamda"

D1p.vec<-D.mat(vec,varp);

  fp1<-function(p,lamda){}
  body(fp1)<-D1p.vec[1]

  fp2<-function(p,lamda){}
  body(fp2)<-D1p.vec[2]

  fp3<-function(p,lamda){}
  body(fp3)<-D1p.vec[3]

  fp<-c(fp1(p,lamda),fp2(p,lamda),fp3(p,lamda))

D1lamda.vec<-D.mat(vec,varlamda)

  flamda1<-function(p,lamda){}
  body(flamda1)<-D1lamda.vec[1]

  flamda2<-function(p,lamda){}
  body(flamda2)<-D1lamda.vec[2]

  flamda3<-function(p,lamda){}
  body(flamda3)<-D1lamda.vec[3]

  flamda<-c(flamda1(p,lamda),flamda2(p,lamda),
  flamda3(p,lamda))

D1pp.vec<-D.mat(D1p.vec,varp)

  fpp1<-function(p,lamda){}
  body(fpp1)<-D1pp.vec[1]
```

```
  fpp2<-function(p,lamda){}
  body(fpp2)<-D1pp.vec[2]

  fpp3<-function(p,lamda){}
  body(fpp3)<-D1pp.vec[3]

  fpp<-c(fpp1(p,lamda),fpp2(p,lamda),fpp3(p,lamda))

D1plamda.vec<-D.mat(D1p.vec,varlamda)

  fplamda1<-function(p,lamda){}
  body(fplamda1)<-D1plamda.vec[1]

  fplamda2<-function(p,lamda){}
  body(fplamda2)<-D1plamda.vec[2]

  fplamda3<-function(p,lamda){}
  body(fplamda3)<-D1plamda.vec[3]

  fplamda<-c(fplamda1(p,lamda),fplamda2(p,lamda),
  fplamda3(p,lamda))

D1lamdalamda.vec<-D.mat(D1lamda.vec,varlamda)

  flamdalamda1<-function(p,lamda){}
  body(flamdalamda1)<-D1lamdalamda.vec[1]

  flamdalamda2<-function(p,lamda){}
  body(flamdalamda2)<-D1lamdalamda.vec[2]

  flamdalamda3<-function(p,lamda){}
  body(flamdalamda3)<-D1lamdalamda.vec[3]

  flamdalamda<-c(flamdalamda1(p,lamda),
  flamdalamda2(p,lamda),flamdalamda3(p,lamda))

F<-2*c(t(fp)%*%W%*%f,t(flamda)%*%W%*%f)

H11<-t(f)%*%W%*%fpp+t(fp)%*%W%*%fp;
H12<-t(f)%*%W%*%fplamda+t(fp)%*%W%*%flamda;
H22<-t(f)%*%W%*%flamdalamda+t(flamda)%*%W%*%flamda;
H1<-2*rbind(c(H11,H12),c(H12,H22));
```

```
    E<-solve(H1)%*%F
    p1<-p-E[1]
    lamda1<-lamda-E[2]

    if(((p-p1)^2+(lamda-lamda1)^2)<epsilon)
    {flag<-0; print("### success ###");};

    p<-p1;
    lamda<-lamda1;
    count<-count+1;
    print(c(p,lamda));

  if(count>max.count){ print("do not coverge"); break();};

    }
    return(list(p,lamda));
  }


  w.step<-function(pf,lamdaf)
  {
   w<-matrix(0,nrow=3,ncol=3)
    for(i in 1:sample)
    {
     D<-c(y[i]-pf,(y[i])^3-pf^3-3*pf*lamdaf^2,
     (y[i])^4-pf^4-6*pf^2*lamdaf^2-3*lamdaf^4)
     w<-w+D%*%t(D)
    }
   W1<-solve(w/sample)
  }

  flag.main<-1;
  result<-beta.step(p,lamda,W)
  p1<-result[[1]]
  lamda1<-result[[2]]

while(flag.main==1)
 {
 W2<-w.step(p1,lamda1)
 result2<-beta.step(p1,lamda1,W2);
 p2<-result2[[1]]
 lamda2<-result2[[2]]

 if((p1-p2)^2+(lamda1-lamda2)^2<epsilon)
```

```
   { print("##### BINGO ###");flag.main=0};

  p1<-p2;
  lamda1<-lamda2;
  print(c(p1,lamda1))
  }

a[k]<-p1
b[k]<-lamda1

}

plot(a,b^2)
mean(a)
var(a)

mean(b^2)
var(b^2)
```