THE UNIVERSITY OF QUEENSLAND
AUSTRALIA

Faculty of Business, Economics and Law
School of Economics

Estimating scale efficiency using a parametric stochastic production frontier model

By
Zi Yin
Bachelor of Economics

Supervisors: Dr Antonio Peyrache and Professor Christopher O'Donnell

An Honours Thesis submitted to the School of Economics, The University of Queensland, in partial fulfilment of the requirements for the degree of BEcon(Honours)

November 2017

Approximate Word Count: 20,000

Declaration Statement

I declare that the work presented in this Honours thesis is, to the best of my knowledge and belief, original and my own work, except as acknowledged in the text, and that material has not been submitted, either in whole or in part, for a degree at this or any other university.

.....

Zi Yin, 1 November 2017

Acknowledgements

Firstly, I would like to thank my supervisors, Dr Antonio Peyrache and Professor Christopher O'Donnell, for their guidance during my honours year. Secondly, I wish to thank my family and friends for supporting me in my academic endeavours. Finally, I would like to thank everyone who volunteered their time to proofread this thesis.

Abstract

In parametric productivity and efficiency analysis (whether in deterministic frontier or stochastic frontier analysis), it is common to assume a Cobb-Douglas production function[1]. The problem in this normal setting is that the level of scale efficiency is not a well-defined measure. This thesis proposes a new affine transformed model by adding a location parameter μ so that the frontier does not have to cross the origin. This not only generalises the normal model but also makes it possible to obtain a more meaningful measure of scale efficiency. Estimation methods for this new model are introduced and analytical solutions for productivity and various efficiency measures are deduced afterwards. Simulation methods are used to assess the properties of the affine transformed model. Finally, an example is given by applying the new model to panel data collected from Philippines rice farmers over the eight years from 1990 to 1997 and precise estimates of the drivers of productivity change are derived accordingly.

[1]Strictly speaking, in stochastic frontier analysis, it is common to assume the production function can be *approximated* by a Cobb-Douglas function.

Contents

List of Tables

List of Figures

List of Abbreviations

CRS	Constant Returns to Scale
CD	Cobb-Douglass
DFA	Deterministic Frontier Analysis
DGP	Data Generating Process
DRS	Decreasing Returns to Scale
IDF	Input Distance Function
IRS	Increasing Returns to Scale
ISE	Input-oriented Scale Efficiency
ITE	Input-oriented Technical Efficiency
MIOS	Mix-Invariant Optimal Scale
ML	Maximum Likelihood
MSE	Mean Squared Error
MVOS	Mix-Variant Optimal Scale
ODF	Output Distance Function
OLS	Ordinary Least Squares
OSE	Output-oriented Scale Efficiency
OTE	Output-oriented Technical Efficiency
SFA	Stochastic Frontier Analysis
TFP	Total Factor Productivity
TSME	Technical Scale Mix Efficiency

Chapter 1

Introduction

1.1 Background

Productivity and efficiency analysis normally involves the measurement of absolute and relative levels of economic performance. Methods of performance measurement can be applied to any production unit (e.g., an individual, a network, an industry, or an economy, which in this thesis is denoted as a decision-making unit (DMU)) that can transform inputs into outputs. One of the common measures for overall economic performance is total factor productivity (TFP) and it can be defined by the ratio of a weighted average of output quantities and a weighted average of input quantities. The calculation is straightforward once the aggregated inputs and outputs are obtained and it is often so simple that a DMU can make TFP comparisons without knowing anything about technologies, markets or firm behaviour. However, if measures of TFP are to be used to inform economic policy and firm strategy, then decision-makers may also want to know the maximum TFP possible in a certain industry, and whether shortfalls in TFP are due to factors that are within or outside the control of firms. Estimating the maximum TFP that is possible involves estimating production frontiers. Explaining shortfalls in TFP requires a good knowledge of markets and firm behaviour. Since scale efficiency is one of the main drivers of TFP (e.g., a firm could be too small in its size, increasing its scale can improve its productivity), constructing a meaningful measure of scale efficiency can be greatly helpful to explain TFP. This paper explains how to frame the analysis of scale efficiency and how affine transformed models can be applied to estimate production frontiers using parametric methods.

The thesis makes contributions to current literature in two main areas. First, it proves the existence of the optimum scale point and provides the analytical solution for this point under the frameworks of the affine transformed model, which will eventually lead to a meaningful measure of the level of scale efficiency. This is important in practice because by knowing the location of the optimum scale point and the "score" of a firm's current economies of scale, the firm manager could determine whether to increase, decrease or keep its current production scale. Nevertheless, when the conventional Cobb-Douglas

(CD) model is used to construct the production frontier, the level of scale efficiency of a specific firm (the ratio between TFP at the firm's technically efficient point and TFP at the optimal scale point) is not well defined. This is because the standard CD production function always emanates from the origin (i.e., it exhibits neither a y-axis nor an x-axis intercept). To be more specific, if the industry is experiencing decreasing returns to scale (DRS), when a standard CD function is used to estimate the frontier, the optimal scale point will always be equal to the origin and if the industry is experiencing increasing returns to scale (IRS) the optimal scale point will always be equal to positive infinity. In either way, the location of the optimal scale point could not provide much helpful information to the firm manager since it is unprofitable to produce nothing within DRS and infeasible to produce infinite products within IRS in the real world. Even if this optimum scale point is accepted, the TFP at the optimal scale point will always be infinity, which will lead the scale efficiency measure of any firm to become zero, which does not make any real sense. Balk (2001), instead of computing the level of scale efficiency, tried to solve the problem by constructing the change of the scale efficiency using the ratio of two scale efficiency measures, where the common denominators (TFP at the optimal scale point) are canceled out. This measure of change can help in the situation where the decision makers want to know how the scale efficiency measures of firms change overtime in relative terms. However, if the purpose is to justify economic policies and identify the scale efficiency gap between a specific firm and the optimum firm possible in the industry, the change in scale efficiency is not a plausible measure any more. An absolute level measurement is required in this case. This creates the initial motivation for working with an affine transformed model, which extends the standard CD model to a more general one by adding a location parameter μ, hence allowing the x-axis intercept to move[1].

Furthermore, this affine transformed model is argued to be more plausible in most real-world situations because it is unlikely that a positive level of output is obtained with only an infinitesimal amount of input(s). Indeed, it is hard to think of any contemporary construction (buildings, roads, and bridges) process where minimum or threshold levels of the required inputs (labor, materials, energy, and machine hours) are not the norm rather than the exception. After adjusting the standard model, a meaningful optimum scale point (therefore a plausible measure of scale efficiency) can be easily constructed by using the parameters of the affine transformed model. For the estimation of the frontier, the thesis proposes a new method called corrected non-linear least squares that combines the ideas of corrected ordinary least squares (Afriat, 1972) and non-linear least squares methods. Furthermore, this thesis has shown that when there are measurement errors in output, a well-defined measure of scale efficiency can still be constructed by using stochastic models (estimated by maximum likelihood methods) and making certain assumptions.

The second aim of the thesis is to expound how different econometric estimations (e.g., cor-

[1]In utility literature, this generalisation of the Cobb-Douglas function is also known as the Stone-Geary function (Geary, 1950; Stone, 1954), which can be interpreted as that people need a minimal level of consumption to be alive. Beattie and Aradhyula (2015) explores the implications of the Stone-Geary utility function as a production function model.

rected nonlinear least squares and maximum likelihood estimation) could be implemented to decompose total factor productivity efficiency into measures of technical efficiency, scale efficiency, residual-mix efficiency and in some cases, statistical noise. This decomposition into meaningful components is of importance because firms can have a better understanding of the magnitude of different factors that drive the economic performance and hence implement different strategies to fit.

A Monte Carlo simulation is designed to test the statistical performance of the estimation methods of the affine transformed model. R, a statistical software package, is used to generate different samples of observations for the estimation of affine transformed frontiers. In both DFA and SFA contexts, even in a small sample with 50 observations, the estimates of the affine transformed model are very close to the parameters of the true model. The results were consolidated into tables to allow comparisons between the data generating process (DGP) parameters and DFA/SFA estimates. Different measures of efficiency calculated by using these estimates are also put into tables to show the validity of the TFP decomposition.

Finally, an empirical illustration of the affine transformed method to a firm-level panel dataset assembled by the International Rice Research Institute (IRRI) is used to explain the drivers of economic performance. The main features of the dataset are described in Pandey et al. (1999). The panel data contains observations on the quantities and prices of four inputs and one output for forty-three Philippines rice farmers over the eight years from 1990 to 1997.

1.2 Outline

The outline of this thesis is as follows. Chapter 2 discusses the relevant productivity and efficiency measurement concepts. Chapter 3 and Chapter 4 explain how to obtain the analytical solution for the different measures of efficiency and how to implement the affine transformed method in DFA and SFA frameworks using corrected non linear least squares and maximum likelihood estimations respectively. Chapter 5 applies the theory developed in Chapter 3 and 4 in a Monte Carlo simulation. The results obtained from the simulation are also discussed. Chapter 6 describes an empirical application of the affine transformed model using an agriculture dataset. Chapter 7 concludes with a summary of key results and a discussion regarding to the limitations of the thesis as well as the directions for future research.

Chapter 2

Productivity and Efficiency Measures

2.1 Introduction

This chapter provides a detailed explanation of the concepts and methodologies required for understanding different measures of efficiency as well as the affine transformed frontier estimation in Chapter 3 and 4. Section 2.2 details common assumptions about technologies that are sufficient for distance functions to exist. Section 2.3 explains how managers behave under different restrictions and the importance of identifying different optimisation problems faced by them. Section 2.4 defines several measures of efficiency associated with the actions of a firm. Section 2.5 concludes.

2.2 Technologies

In general, any production process can be represented as a process that transforming inputs(x) into outputs(q). Where both x and q are column vectors to account for a multi-dimension problem. Knowledge of the technology is vital for measuring efficiency since the technology will determine the level and shape of the frontier.

In this thesis, a technology is defined as a technique, method or system for transforming inputs into outputs (O'Donnell, 2016). In general, it is helpful to think of a technology as a recipe, or a book of instructions. If a technology is regarded as a book of instructions, a technology set can be naturally regarded as a library that includes all kinds of "technology". A common way of describing the "technology" of a typical firm in economics is to define a production possibilities set $\Psi(t)$ that consists of all possible feasible combinations of inputs x and outputs q in time period t

$$\Psi(t) \equiv \{(x,q) \in \Re_+^N \times \Re_+^M : \text{q is producible from x in period t}\}$$

Without any constraints the above production possibilities set is too general to be practically useful. Hence the following properties, often called the assumptions of production theory that reflect people's common sense or common belief on what a typical production possibilities set should be, will be imposed (Balk, 2013).

Assumption 1: Inactivity is possible (i.e. it is possible to use nonnegative inputs to produce zero outputs);

Assumption 2: Strong disposability of inputs and outputs (i.e. for outputs disposability: if x is capable of producing q^*, it can also produce any other q that is not larger than q^*);

Assumption 3: Boundedness (i.e. only a limited quantity of outputs is producible using a given input vector); and

Assumption 4: Weak essentiality (i.e. no free lunch).

Färe and Primont (1995) explained that if a production possibilities set satisfies assumptions 2 and 3, then distance functions are equivalent representations of the production possibilities set $\Psi(t)$[1]. Distance functions can be used to describe the production possibilities set in a way that allows productivity and efficiency to be measured. The concept of distance functions were introduced by Shepherd (1970) and is heavily used in productivity theory and contemporary efficiency and productivity analysis. Distance functions are used to measure the distance between an input-output combination and the frontier. The values of these distance functions are also independent of any units of measurement. An output distance function (ODF) is represented in terms of proportional changes in outputs and is defined as:

$$D_O^t(x,q) = \inf\left\{\theta > 0 : \left(x, \frac{q}{\theta}\right) \in \Psi(t)\right\}$$

Several simple properties of the ODF follow from the axioms specified on the production possibilities set (See Färe and Primont (1995) for a more detailed explanation):

1. $D_O^t(x, \mathbf{0}_M) = 0$ for all $x \in \mathbb{R}_+^N$;
2. $D_O^t(x, \mu q) = \mu D_O^t(x,q)$ for all (x,q) in $\mathbb{R}_+^{N+M}$ and for all $\mu > 0$ (linearly homogeneous);
3. $D_O^t(x,q)$ is lower semi-bounded on $\mathbb{R}_+^M$: $\left\{q : D_O^t(x,q) \leqq \alpha\right\}$ is bounded for all $\alpha > 0$ and $x \in \mathbb{R}_+^N$;

[1]Strictly speaking, a production possibilities set can be represented by distance functions as long as two assumptions are satisfied: weak disposability of inputs and outputs and boundedness assumption.

4. $D_O^t(x,q)$ is lower semi-continuous $\mathbb{R}_+^M$: $\left\{q : D_O^t(x,q) \leqq \alpha\right\}$ is closed for all $\alpha > 0$ and $x \in \mathbb{R}_+^N$;
5. $D_O^t(\mathbf{0}_N, q) = +\infty$ for all $q \geq \mathbf{0}_M$
6. $D_O^t(x,q)$ is non-decreasing in q and non-increasing in x;
7. If $(x,q) \in \Psi(t)$, then $D_O^t(x,q) \leq 1$; and
8. $D_O^t(x,q) = 1$ if (x,q) belongs to the frontier of $\Psi(t)$.

An input distance function (IDF) is represented in terms of proportional changes in inputs and is defined as:

$$D_I^t(x,q) = \sup\left\{\lambda > 0 : \left(\frac{x}{\lambda}, q\right) \in \Psi(t)\right\}$$

Several simple properties of the IDF follow from the axioms specified on the production possibilities set (See Färe and Primont (1995) for a more detailed explanation):

1. $D_I^t(x, \mathbf{0}_M) = +\infty$ for all $x \in \mathbb{R}_+^N$;
2. $D_I^t(\lambda x, q) = \mu D_I^t(x,q)$for all (x,q) in $\mathbb{R}_+^{N+M}$ and for all $\lambda > 0$ (linearly homogeneous);
3. $D_I^t(x,q)$ is upper semi-bounded on $\mathbb{R}_+^M$;
4. $D_I^t(x,q)$ is upper semi-continuous $\mathbb{R}_+^M$;
5. $D_I^t(x,q)$ is is non-decreasing in $\boldsymbol{x}$ and non-increasing in q;
6. If $(x,q) \in \Psi(t)$, then $D_I^t(x,q) \geq 1$; and
7. $D_I^t(x,q) = 1$ if (x,q) belongs to the frontier of $\Psi(t)$.

2.3 Managerial Behaviour

One of the difficult problems that continuously bother the board of directors nowadays is to evaluate how well a manager manages a firm. A measure that is commonly used to quantify a manager's performance is total factor productivity (TFP). TFP is defined as a measure of total output divided by a measure of total input (O'Donnell, 2012). However, this measure of absolute economic performance may not work if managers are facing some constraints or events that are beyond their control. For example, suppose there are two firms A and B in a certain industry with the structure of increasing returns to scale and both input and output mixes are predetermined. It is unfair to compare the TFPs of two managers A and B when A can freely change the scale of output and inputs while B cannot. As manager A faces less restrictions than B, the more sensible measure in

this case could be a component of TFP, namely technical efficiency (TE), which measures how well the manager makes use of existing techniques. TE is more suitable in this case for evaluating managers A and B because both of them can choose the technique, or the "book of instructions" freely while they face different constraints with regard to the scale of inputs and outputs.

Besides standing on the perspective of an appraiser, it is also true for managers themselves. Different managers will generally use different strategies depending on what they value and what kind of constraints they are facing. For example, if managers can choose technique and scale of outputs and inputs freely but are not able to choose mix of outputs and inputs at will or simply they are in the situation of single input and single output, they will tend to maximise TFP by maximising both technical efficiency and scale efficiency (TSE).

To be more specific, firm managers may face the following different optimisation problems in practice. For clarity, start from this section, firm and time subscripts will be introduces into the notation. Thus, for example, $q_{it} = (q_{1it}, ..., q_{Nit})'$ and $x_{it} = (x_{1it}, ..., x_{Mit})'$ now denote the output and input vectors of firm i in period t.

When inputs are predetermined and the manager of firm i can only choose output vectors that are scalar multiples of q_{it}, then the period-t output-maximisation problem can be written as

$$\max_{q \geq 0} \{Q(q) : q \propto q_{it}, D_O^t(x_{it}, q) \leq 1\} \tag{2.1}$$

where $Q(.)$ is any nonnegative, nondecreasing, linearly-homogeneous, scalar-valued aggregator function satisfying $Q(q_{it}) > 0$ and $D_O^t(x_{it}, q)$ is the output distance function at time period t. Notice that as inputs are predetermined in this case, maximising output is the same to maximise TFP.

When the manager of firm i can only use a scalar multiple of x_{it} to produce a scalar multiples of q_{it}, then the period-t TFP-maximisation problem can be written as

$$\max_{x \geq 0, q \geq 0} \{Q(q)/X(x) : x \propto x_{it}, q \propto q_{it}, D_o^t(x, q) \leq 1\} \tag{2.2}$$

where $Q(.)$ and $X(.)$ are nonnegative, nondecreasing, linearly-homogeneous, scalar-valued aggregator functions that satisfy $TFP(x_{it}, q_{it}) = Q(q_{it})/X(x_{it}) > 0$ and $D_O^t(x, q)$ is the output distance function at time period t.

Finally, when the manager of firm i can choose outputs and inputs freely, then the period-t TFP-maximisation problem can be written as

$$\max_{x \geq 0, q \geq 0} \{Q(q)/X(x) : D_O^t(x, q) \leq 1\} \tag{2.3}$$

where $Q(.)$ and $X(.)$ are nonnegative, nondecreasing, linearly-homogeneous, scalar-valued aggregator functions that satisfy $TFP(x_{it}, q_{it}) = Q(q_{it})/X(x_{it}) > 0$ and $D_O^t(x, q)$ is the output distance function at time period t. Notice here we can change the output inequality $D_O^t(x, q) \leq 1$ to the input inequality $D_I^t(x, q) \geq 1$, which will yield the same results.

To sum up, the complexities of the situations and the different optimisation problem faced by the managers in reality lead to the initial motivation to construct different efficiency measures introduced in what follows.

2.4 Measures of Productivity and Efficiency

This section describes the various ways and techniques to measure productivity and efficiency.

2.4.1 Total Factor Productivity (TFP)

In this thesis, to avoid ambiguity we define the general productivity as total factor productivity (TFP) which is a measure of productivity involving all factors of production and all outputs in a multiple output setting. O'Donnell (2012) defines the TFP of firm i in period t as:

$$TFP(x_{it}, q_{it}) = \frac{Q(q_{it})}{X(x_{it})} \tag{2.4}$$

The measures of total (or aggregate) output and input are $Q(q_{it})$ and $X(x_{it})$ where $Q(.)$ and $X(.)$ are non-decreasing, non-negative and homogeneous of degree one functions that transform vectors (q_{it}, x_{it}) to scalars $(Q(q_{it}), X(x_{it}))$. As in this thesis only single output case will be discussed, it is sufficient to write $Q(q_{it}) = q_{it}$ where q_{it} is already a scalar. TFP is useful when we want to compare the level of performance among different firms. However, it is not a sufficient measurement if we want to know how wide the gap is between the performance of a specific firm and the performance of the most productive firm possible in the industry. Therefore the technical scale mixed efficiency (TSME) of firm i in period t is introduced and defined as follows:

$$TSME^t(x_{it}, q_{it}) = \frac{TFP(x_{it}, q_{it})}{TFP(x_{it}^*, q_{it}^*)} \tag{2.5}$$

where $TFP(x_{it}^*, q_{it}^*)$ represents the maximum TFP that is possible in period t, and the associated input-output combination is (x_{it}^*, q_{it}^*). The reason that the ratio between the individual TFP and the maximum potential TFP is named as TSME will be explained after different measures of efficiency are introduced. The main motivation for decomposing this

overall performance measure to other efficiency components, as we talked about in Section 2.3, is due to the various constrained optimisation problems faced by managers. For the most general case, when a firm's manager faces an unconstrained optimisation problem[2], (s)he can choose inputs freely and aims to maximise TFP by solving problem (2.3). TSME can be viewed as a measure of how well (s)he solves this optimisation problem. There may be several pairs of output and input that solve this problem. Let (x_{it}^*, q_{it}^*) denote one such pair. Then the associated maximum TFP will be $Q(q_{it}^*)/X(x_{it}^*)$, which is exactly the denominator of 2.5. One of the main targets of this thesis is to identify the combination of inputs and outputs that give rise to $TFP(x_{it}^*, q_{it}^*)$.

2.4.2 Technical Efficiency (TE)

A firm could be assessed as overall unproductive due to technical inefficiency. Technical efficiency is a measure of how close a firm is operating to the production frontier in a given time period, which was first introduced by Farrell (1957), who incorporated the work of Debreu (1951) and Koopmans et al. (1951) to define a measure of efficiency that could allow for a multiple inputs scenario. Farrell (1957) claims that technical efficiency reflects a firm's ability to produce maximal output using a given set of inputs. Technical efficiency is a good method to evaluate a manger's performance when the manager could neither change the scale of inputs nor the mixes of inputs and outputs, instead the only thing that is within his/her control is the choice of techniques[3]. For example, two bread makers A and B, are both using 1 cup of water and 2 cups white bread flour and 2 tablespoons white sugar to produce a loaf of white bread. However, due to the deficient technology A chose, half of the bread (s)he made has to be discarded, in the mean time all of the bread made by B can be used. Hence bread maker B is said to be technically more efficient than A. Output-oriented technical efficiency (OTE) of firm i in period t is defined as:

$$OTE^t(x_{it}, q_{it}) = D_O^t\left(x_{it}, q_{it}\right) \tag{2.6}$$

Input-oriented technical efficiency (ITE) of firm i in period t is defined as:

$$ITE^t(x_{it}, q_{it}) = \frac{1}{D_I^t\left(x_{it}, q_{it}\right)}$$

By analogy with TSME in equation (2.5), technical efficiency can also be defined in terms of TFP. For example

[2]Strictly speaking, there is still a hidden constraint that (x_{it}, q_{it}) should belong to the production possibilities set.

[3]O'Donnell (2016) argues that there are several reasons for a firm to be technically inefficient. 1) a firm may not be able to control its inputs and the revenue-maximising point lies within the frontier. 2) a firm may not be able to can make full use of the existing techniques (i.e., books of instructions). 3) a firm is able to choose the best possible technology but is simply making mistakes.

$$OTE^t(x_{it}, q_{it}) = \frac{TFP(x_{it}, q_{it})}{TFP(x_{it}, \overline{q}_{it})} \tag{2.7}$$

where $TFP(x_{it}, \overline{q}_{it}) = Q(\bar{q}_{it})/X(x_{it})$ and $Q(\bar{q}_{it}) = Q(q_{it})/D_O^t(x_{it}, q_{it})$ is the maximum aggregate output possible holding x_{it} and the output mix fixed. In a single output case, we could simply write $OTE^t(x_{it}, q_{it}) = q_{it}/\overline{q}_{it}$ where $\bar{q}_{it} = q_{it}/D_O^t(x_{it}, q_{it})$ is the maximum output possible holding x_{it} fixed. As a result, the constrained optimisation problem faced by a firm's manager is simplified to maximise TFP holding x_{it} and the output mix fixed, which is formally defined in problem (2.1). Then output-oriented technical efficiency can be viewed as a measure of how well (s)he solves this problem. Notice that the solution to this optimisation problem is $\overline{q}_{it}$, together with the input x_{it}, the associated TFP will be $TFP(x_{it}, \overline{q}_{it})$, which leads to the denominator of equation (2.6).

In the same way ITE can be defined as

$$ITE^t(x_{it}, q_{it}) = \frac{TFP(x_{it}, q_{it})}{TFP(\bar{x}_{it}, q_{it})}$$

where $TFP(\bar{x}_{it}, q_{it}) = Q(q_{it})/X(\bar{x}_{it})$ and $X(\bar{x}_{it}) = X(x_{it})/D_I^t(x_{it}, q_{it})$ is the minimum aggregate input possible holding q_{it} fixed and the input mix fixed.

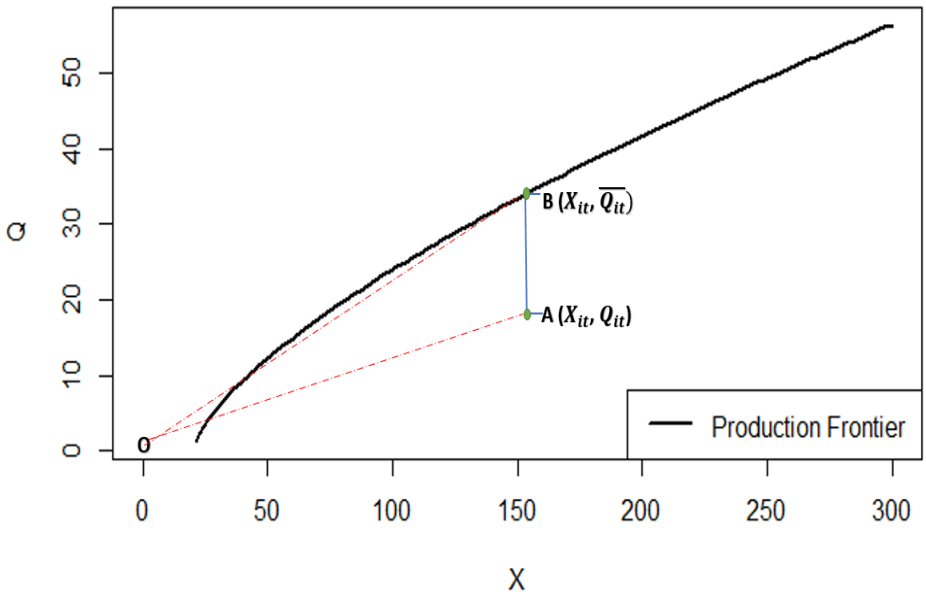
An affine transformed frontier with mix restriction, named mix restricted affine transformed frontier will be used as a graphical example (Figure 2.1) to demonstrate the intuition of OTE. Noticing that TE can be identified in both standard and affine transformed frontiers, however, the affine transformed frontier will show its superior in the next subsection where the goal is to identify a meaningful measure of scale efficiency. For illustration purposes, it is assumed that the managers are subject to the following restrictions.

Mix restriction: Managers could only choose the predetermined mixes of inputs and outputs

Input scale restriction: Managers could only choose the predetermined scale of aggregated input.

Under the above two restrictions, the only way to enhance a firm's TFP is to improve TE, or be more precise, the OTE. Since managers basically could not change any level of inputs, they have to come up with a more effective book of instruction and follow the correct instructions in order to take full advantage of the fixed inputs. Suppose a firm's inputs and outputs are (x_{it}, q_{it}), therefore the aggregate input and output are $(X(x_{it}), Q(q_{it}))$. We denote the aggregate output-input combination $(X(x_{it}), Q(q_{it}))$ as point A, which is below the production frontier. The reason of this deficiency could be that the firm's manager A applies out-of-date technologies to make production. Assume the board of this firm hires another manager B, who applies the newest technology to manufacture a higher level of aggregate output $Q(\bar{q}_{it})$ while using the same input x_{it},

Figure 2.1: *OTE demonstration*



which is denoted as point B $(X(x_{it}), Q(\bar{q}_{it}))$. The TFP at point A is $Q(q_{it})/X(x_{it}) =$ slope OA. The TFP at point B is $Q(\bar{q}_{it})/X(x_{it}) =$ slope OB. From Figure 2.1, it is clear to see the slope of OA is lower than the slope of OB, hence we could conclude that under the scale and mix constraints, manager B performs better than manager A due to the higher OTE manager B creates. The OTE of this firm at manager A's control is $TFP(x_{it}, q_{it})/TFP(x_{it}, \overline{q}_{it}) = Q(q_{it})/Q(\overline{q}_{it})$, which is obviously smaller than 1, and the OTE at manager B's control is $TFP(x_{it}, \overline{q}_{it})/TFP(x_{it}, \overline{q}_{it}) = Q(\overline{q}_{it})/Q(\overline{q}_{it}) = 1$. Noting that although any point between A and B has a higher TFP than A, we could only use B's TFP as the denominator of OTE because point B is on the frontier where its aggregate output $Q(\overline{q}_{it})$ is indeed the maximum aggregate output possible at inputs x_{it} under the stated two restrictions. From Table 2.1 we can clearly see that under the mix and input scale restriction, $TFP(x_{it}, q_{it}) = TFP(x_{it}, \overline{q}_{it})OTE^t(x_{it}, q_{it})$, the reason why B reaches the maximum TFP is that (s)he managed to hit the maximum OTE, which is equal to 1.

Table 2.1: *TFP decomposition under mix and input scale restriction*

Observation	TFP	$TFP(x_{it}, \overline{q}_{it})$	OTE
A	$\frac{Q(q_{it})}{X(x_{it})}$	$\frac{Q(\bar{q}_{it})}{X(x_{it})}$	$\frac{Q(q_{it})}{Q(\bar{q}_{it})}$
B	$\frac{Q(\bar{q}_{it})}{X(x_{it})}$	$\frac{Q(\bar{q}_{it})}{X(x_{it})}$	1

However, the board may no longer be satisfied with the full technical efficiency if now the manager B is able to freely choose input and output vectors that are scalar multiples of x_{it} and q_{it}. After the scale constraint is released, this firm could have further improved its overall production performance by adjusting its production scale. Nevertheless, because the manager B has never learned micro-economics basic concepts like decreasing returns to scale or increasing returns to scale, (s)he believes the current scale of inputs and outputs is efficient enough and does not want to change it at all even though (s)he is able to.

2.4.3 Scale Efficiency (SE)

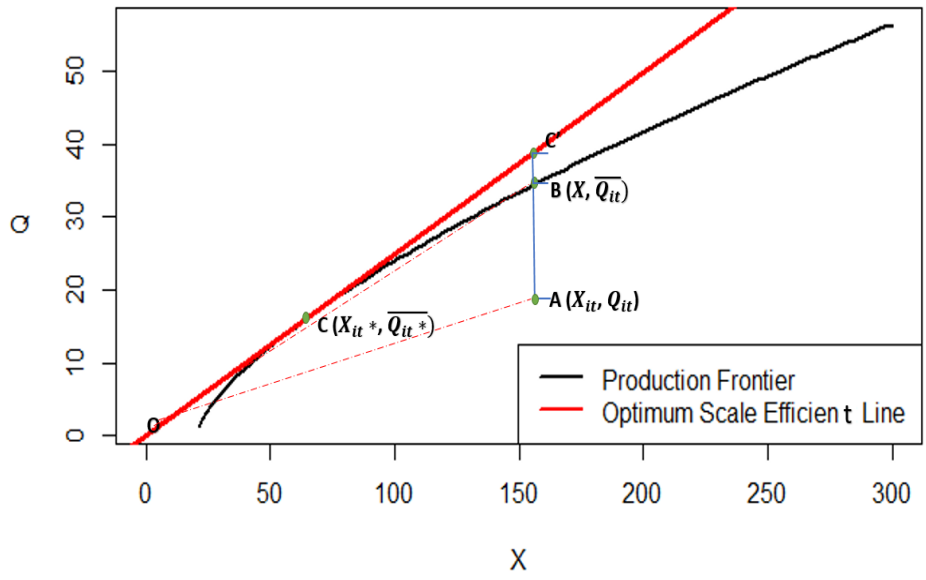
A firm could be assessed as overall unproductive even though it is fully technical efficient. In O'Donnell (2008), scale efficiency is defined as a measurement of economies of scale and implies whether the scale of production of an individual firm is optimum. For example, a firm could be too small in its scale, hence by increasing its size the firm could increase its productivity, and vice versa. A firm could be technically efficient but scale inefficient. O'Donnell (2012) explains that if both input and output mixes are held fixed, a firm will maximise TFP by moving to a point of mix-invariant optimal scale (MIOS), also known as the optimum scale point, where a ray from the origin is tangent to a mix-restricted frontier. One of the characteristics of the tradition frontier model is the frontier curve is settled to cross the origin, as a result it cannot identify an informative MIOS, which will be explained in detail in Section 3.3. The main contribution of this thesis is to provide a new model, namely the affine transformed model to identify a meaningful MIOS point. After obtaining the MIOS point, scale efficiency is constructed as the ratio between TFP at a technically efficient input-output combination and TFP at MIOS point. Output-oriented scale efficiency (OSE) and input-oriented scale efficiency (ISE) of firm i in period t can be defined as

$$OSE^t(x_{it}, q_{it}) = \frac{TFP(x_{it}, \overline{q}_{it})}{TFP(\breve{x}_{it}, \breve{q}_{it})} \tag{2.8}$$

$$ISE^t(x_{it}, q_{it}) = \frac{TFP(\overline{x}_{it}, q_{it})}{TFP(\breve{x}_{it}, \breve{q}_{it})} \tag{2.9}$$

where $\breve{x}_{it}$ and $\breve{q}$ denote the input and output vectors at the point of MIOS; $\overline{x}_{it}$ and $\overline{q}_{it}$ are as defined in Section 2.4.2. As a result, the constrained optimisation problem under

Figure 2.2: *OTE and OSE demonstration*



the mix restriction can be simplified to problem (2.2) where the restriction of fixed mix of inputs and outputs is defined by $x \propto x_{it}, q \propto q_{it}$. Then output-oriented technical and scale efficiency can be viewed as a measure of how well (s)he solves this problem. Notice that the solution to this optimisation problem is $(\breve{x}_{it}, \breve{q}_{it})$ and the associated TFP will just be the denominator of equation (2.8).

Again, a graphical example (Figure 2.2) is given by using the mix restricted affine transformed frontier in order to demonstrate the intuition of OSE. Here it is assumed the managers are only subject to the mix restriction.

Mix restriction: Managers could only choose the predetermined mixes of inputs and outputs

Now it is possible to improve a firm's TFP in two ways: increasing TE and increasing SE. Suppose the previous firm's combination of aggregate input and output under manager B's control are fully technical efficient, denoted as point B $(X(x_{it}), Q(\bar{q}_{it}))$. Assume the board of this firm is not content with B's performance any longer as the board finds out that B is not operating the firm at the point of MIOS, which leads the board to hire another manager C, who has a better understanding of the idea of economies of scale than manager B. After re-scaling the inputs and outputs, manager C succeeds to improve TFP from point B to point C. Noting that point C is the tangent point pinpointed by the optimum scale efficient line (red line) and the production frontier (black curve). The TFP at point B is $Q(\bar{q}_{it})/X(x_{it}) =$ slope OB. The TFP at point C, known as the MIOS point, is $Q(\breve{q}_{it})/X(\breve{x}_{it}) = slope\, OC = slope\, OC'$. C' is the point on the optimum scale efficient line which has the same aggregate input as A, it is being pointed out because it is easier to

see the gap between B and C', which results in the scale inefficiency. From Figure 2.2, it is clear to see that the slope of OB is lower than the slope of OC, hence we could conclude that under the mix constraint alone, manager C performs better than manager B due to the higher SE manager C creates. As both point B and point C are on the production frontier, they have the same OTE, which are maximised to 1. However, the OSE at manager B's control is $(Q(\bar{q}_{it})/X(x_{it})) \times (X(\breve{x}_{it})/Q(\breve{q}_{it})) < 1$ while the OSE at manager C's control is $(Q(\breve{q}_{it})/X(\breve{x}_{it})) \times (X(\breve{x}_{it})/Q(\breve{q}_{it})) = 1$. Noticing that the crown of the maximum TFP possible under the mix restriction is awarded to C now. We could use C's TFP as the denominator of OSE as C is exactly the point of MIOS. From Table 2.2 it is clear to see that under the mix restriction, $TFP(x_{it}, q_{it}) = TFP(\breve{x}_{it}, \breve{q}_{it})OTE^t(x_{it}, q_{it})OSE^t(x_{it}, q_{it})$. The implication of this equation is essential to understand TFP, for example, if we look at the most inefficient manager A, instead of vaguely saying (s)he is a bad manager, we can actually diagnose the reason of the bad performance by looking at A's OTE and OSE score. For the same reasoning, the most efficient manager C is awarded the maximum TFP because (s)he managed to hit the maximum OTE and maximum OSE at the same time which are both equal to 1.

Is this the end of the story? Yes, but only if it is in the single input and single output case or the manager is under the mix constraint, where we can conclude that manager C has taken the full advantage of the available resources and has reached the maximum possible TFP under certain constraint. Nevertheless, it is not true in the most general case. Suppose after a while, the board realises that it is possible to free the mix constraint which will lead to a even higher TFP. As a consequence, the board is no longer satisfied with the full technical efficiency and the full scale efficiency. Although manager C has a title of economics Ph.D, (s)he does not have an engineer background and could not figure out the technical details of inputs and outputs mix. Now the firm need a versatile CEO who can not only handle economy of scale problem but also know how to find the optimum mix among numerous combinations of inputs and outputs in order to ultimately exploit the available resource and maximise TFP .

Table 2.2: *TFP decomposition under mix restriction*

Observation	TFP	$TFP(\breve{x}_{it}, \breve{q}_{it})$	OTE	OSE
A	$\frac{Q(q_{it})}{X(x_{it})}$	$\frac{Q(\breve{q}_{it})}{X(\breve{x}_{it})}$	$\frac{Q(q_{it})}{Q(\bar{q}_{it})}$	$\frac{Q(\bar{q}_{it})}{X(x_{it})} \frac{X(\breve{x}_{it})}{Q(\breve{q}_{it})}$
B	$\frac{Q(\bar{q}_{it})}{X(x_{it})}$	$\frac{Q(\breve{q}_{it})}{X(\breve{x}_{it})}$	1	$\frac{Q(\bar{q}_{it})}{X(x_{it})} \frac{X(\breve{x}_{it})}{Q(\breve{q}_{it})}$
C	$\frac{Q(\breve{q}_{it})}{X(\breve{x}_{it})}$	$\frac{Q(\breve{q}_{it})}{X(\breve{x}_{it})}$	1	1

2.4.4 Residual Mix Efficiency (RME)

A firm could be judged overall unproductive even though it is both fully technically efficient and scale efficient due to residual mix inefficiency. O'Donnell (2008) views residual mix efficiency as a component of TSME that remains after accounting for pure technical and pure scale efficiency effects, which shows potential changes in productivity when input and

output mix are free to vary. Relaxing the restrictions on the input and output mix can lead to an expansion of the set of input-output combinations available to a firm. O'Donnell (2012) defines the boundary of this expanded set the unrestricted frontier, as opposed to the mix restricted frontier mentioned in Section 2.4.2 and 2.4.3. It is clear to see that the unrestricted frontier is exactly the boundary of the production possibilities set $\Psi(t)$ defined in Section 2.2, which by definition must envelop all feasible mix restricted frontiers. A firm could be inefficient due to residual mix inefficiency. Suppose we are in the case of multiple outputs and multiple inputs and at the point of MIOS, where both TE and SE are equal to 1, if now we free the mix restriction, a firm could further maximise TFP by moving to a point we call mix-variant optimal scale (MVOS), where a ray from the origin is tangent to the unrestricted frontier. Notice that MVOS point is also known as the maximum TFP point in many common productivity and efficiency textbooks. Residual mix efficiency is the ratio between TFP at a point of MIOS on the mix restricted frontier and TFP at a point of MVOS on the unrestricted frontier. The use of the term "mix" is straightforward—the movement from MIOS to MVOS is a movement from an optimal point on a mix restricted frontier to an optimal point on an unrestricted frontier, so the changes in TFP is essentially a mix effect. However, the term residual is added because this movement may also involve a change in scale.

Residual mix efficiency is defined in O'Donnell (2008) as:

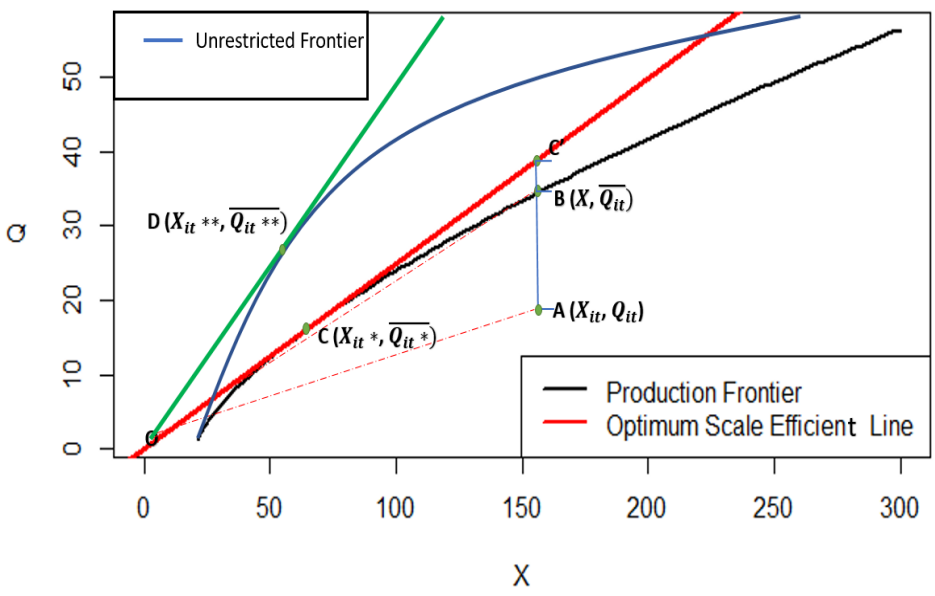
$$RME^t(x_{it}, q_{it}) = \frac{TFP(\breve{x}_{it}, \breve{q}_{it})}{TFP(x^*_{it}, q^*_{it})} \tag{2.10}$$

where x^*_{it} and q^*_{it} denote the input and output vectors at MVOS point and $\breve{x}_{it}$, and $\breve{q}$ are as defined in Section 2.4.3. After a long way solving different kinds of maximisation problem of TFP under certain constraints we can finally come back to the most general case discussed in Section 2.4.1 - to find the maximum TFP that is possible in period t where inputs, outputs and technology are all free to vary. The optimisation problem has already been defined in equation (2.3). The solution to this optimisation problem is (x^*_{it}, q^*_{it}) and the associated TFP will just be the denominator of equation (2.10).

For a graphical example (Figure 2.3), here we add a new unrestricted affine transformed frontier (blue curve) to previous Figure 2.2 in order to demonstrate the meaning of RME in a more intuitive way. It is also assumed that managers can choose input and output freely to maximise TFP.

Under this most general case, it is possible to enhance a firm's TFP through TE, SE and RME. Knowing this condition, the board of previous firm will be again unhappy with C's performance as C could further improve TFP by moving from MIOS point to MVOS point, which leads the board to hire another manager D, who is familiar with the manipulation of inputs and outputs mixes. After re-mixing the inputs and outputs, manager D succeeds to improve TFP from point C to point D. Noting that point D is the tangent point pinpointed by the unrestricted optimum technical scale

Figure 2.3: *OTE, OSE and RME demonstration*



mix efficient line (green line) and the unrestricted production frontier (blue curve). The TFP at point C is $Q(\breve{q}_{it})/X(\breve{x}_{it})$ = slope OC. The TFP at point D (MVOS point), is $Q(q^*_{it})/X(x^*_{it})$ = slope OD. From Figure 2.3 it is clear to see that the slope of OC is lower than the slope of OD, hence we could conclude that when no restriction is imposed, manager D performs better than manager C due to the higher RME manager C creates. As both point C and point D are on its respective production frontier, they have the same $OTE = 1$. In addition, as they are both at the tangent point of the ray crossing the origin and its respective frontier, they have the same $OSE = 1$ for their respective frontiers. However, the crown of maximum TFP is given to D now as the RME at manager C's control is $(Q(\breve{q}_{it})/X(\breve{x}_{it})) \times (X(x^*_{it})/Q(q^*_{it})) < 1$ while the RME at manager D's control is $(Q(q^*_{it})/X(x^*_{it})) \times (X(x^*_{it})/Q(q^*_{it})) = 1$. We could use D's TFP as the denominator of RME because D is exactly the point of MVOS and it reaches the maximum TFP possible in the most general case (no constraint). From Table 2.3 we can clearly see that when no restriction is imposed, $TFP(x_{it}, q_{it}) = TFP(x^*_{it}, q^*_{it})OTE^t(x_{it}, q_{it})OSE^t(x_{it}, q_{it})RME^t(x_{it}, q_{it})$. In the end, the board of the firm could have a rest and keep the most efficient manager D to perform at time period t since D achieves the maximum TFP possible in the most general case by hitting $OTE = OSE = RME = 1$ at the same time.

Table 2.3: *TFP decomposition under no restriction*

Observation	TFP	$TFP(x^*_{it}, q^*_{it})$	OTE	OSE	RME
A	$\frac{Q(q_{it})}{X(x_{it})}$	$\frac{Q(q^*_{it})}{X(x^*_{it})}$	$\frac{Q(q_{it})}{Q(\bar{q}_{it})}$	$\frac{Q(\bar{q}_{it})}{X(x_{it})}\frac{X(\breve{x}_{it})}{Q(\breve{q}_{it})}$	$\frac{Q(\breve{q}_{it})}{X(\breve{x}_{it})}\frac{X(x^*_{it})}{Q(q^*_{it})}$
B	$\frac{Q(\bar{q}_{it})}{X(x_{it})}$	$\frac{Q(q^*_{it})}{X(x^*_{it})}$	1	$\frac{Q(\bar{q}_{it})}{X(x_{it})}\frac{X(\breve{x}_{it})}{Q(\breve{q}_{it})}$	$\frac{Q(\breve{q}_{it})}{X(\breve{x}_{it})}\frac{X(x^*_{it})}{Q(q^*_{it})}$
C	$\frac{Q(\breve{q})}{X(\breve{x}_{it})}$	$\frac{Q(q^*_{it})}{X(x^*_{it})}$	1	1	$\frac{Q(\breve{q}_{it})}{X(\breve{x}_{it})}\frac{X(x^*_{it})}{Q(q^*_{it})}$
D	$\frac{Q(q^*_{it})}{X(x^*_{it})}$	$\frac{Q(q^*_{it})}{X(x^*_{it})}$	1	1	1

2.5 Conclusion

In this chapter, necessary concepts (e.g., technologies and distance functions) for understanding measures of productivity and efficiency have been introduced. As different managers will generally behave differently depending on their values and restrictions, managerial behaviours have also been discussed and a number of optimisation problems faced by managers have been formalised. Motivated by measuring how well managers could solve these optimisation problems, a number of efficiency measures (e.g., technical efficiency, scale efficiency, residual mix efficiency and technical scale mixed efficiency) have been constructed. The next chapter will use regression analysis to estimate production frontiers and discuss both the limitations of the standard model and the merits of the affine transformed model under the deterministic frontier framework.

Chapter 3

Deterministic Frontier Analysis

3.1 Introduction

Productivity and efficiency analysis involves estimating production frontiers. One way to estimate the frontier is using regression models in which the explanatory variables are deterministic (i.e., not random). The associated frontiers are known as deterministic frontiers. From now on, it is always assumed single output q_{it} and multiple inputs x_{it}, where x_{it} is a vector with M elements in it.

As ODF is homogeneous of degree one in output (property 2 in Section 2.2), it implies $D_O^t(x_{it}, q_{it}) = q_{it} D_O^t(x_{it}, 1)$, therefore

$$\begin{aligned} q_{it} &= \frac{D_O^t(x_{it}, q_{it})}{D_O^t(x_{it}, 1)} \\ &= F^t(x_{it}) D_O^t(x_{it}, q_{it}) \end{aligned} \tag{3.1}$$

where $F^t(x_{it}) = \frac{1}{D_O^t(x_{it},1)}$ is a production frontier (eg., linear, Cobb-Douglas, Translog). After some simple algebra, this function can be rewritten as

$$y_{it} = G^t(x_{it}) - u_{it} \tag{3.2}$$

where $y_{it} \equiv \ln q_{it}$ is the logarithm of the output, $G^t(x_{it}) \equiv \ln F^t(x_{it})$ is the logarithm of the frontier function, and $u_{it} \equiv -\ln D_O^t(x_{it}, q_{it})$. From equation (2.6), it is easy to obtain $u_{it} \equiv -\ln OTE^t(x_{it}, q_{it})$, which represents the nonnegative output-oriented technical inefficiency effect.

The parameters of deterministic frontier models (DFMs) can be estimated using growth accounting (GA), least squares (LS) and maximum likelihood (ML) methods. Due to the limitations of space, only LS methods will be discussed here.

The most common DFMs are underpinned by the following assumptions:

DFA1: all inputs and outputs variables are observed and measured without error; and

DFA2: the functional form of the ODF is known.

If these assumptions are true, then the output distance function can be rewritten as a regression model with error terms representing technical inefficiency and the slope estimators obtained by LS will be unbiased and consistent (if the inefficiency term is statistical independent from the vector of inputs).

3.2 Standard Model (COLS)

When $F^t(x_{it})$ is a standard Cobb-Douglas function, $F^t(x_{it}) = A\Pi_m(x_{mit})^{b_m}$. Equation (3.2) can be written as

$$y_{it} = a + \Sigma_{m=1}^{M} b_m \ln(x_{mit}) - u_{it} \tag{3.3}$$

where $y_{it} \equiv \ln q_{it}$, $a = \ln A$ and $u_{it} \equiv -\ln OTE^t(x_{it}, q_{it})$. Notice that the above equation shows a linear relationship between y_{it} and $\ln x_{mit}$, which makes it possible to use ordinary least squares to conduct estimations.

Least squares (LS) estimation of DFMs involves choosing the unknown parameters to minimise the sum of squared inefficiency effects. In the efficiency literature, it is common to assume that u_{it} is a random variable with the following properties:

LS1: $E(u_{it}) = \mu \geq 0$ for all i and t.

LS2: $Cov(u_{it}, \ln x_{it}) = 0$ for all i and t.

LS3: $Var(u_{it}) \propto \sigma_u^2$ for all i and t.

LS4: $Cov(u_{it}, u_{ks}) = 0$ if $i \neq k$ or $t \neq s$.

LS1 says the inefficiency effects have the same mean. LS2 says u_{it} is uncorrelated with the explanatory variables. LS3 says they have the same variance (i.e., they are homoskedastic). LS4 says they are serially and spatially uncorrelated. It is possible to test LS2–LS4 using standard tests available in introductory econometrics textbooks. If the DFM is linear in the unknown parameters and assumptions LS1–LS4 are true, then ordinary least squares (OLS) estimators for the slope parameters in the model are unbiased and consistent. However, the OLS estimator for the intercept is biased and inconsistent. There are several estimation strategies available to estimate the intercept consistently like linear (quadratic) programming (Aigner and Chu, 1968) and modified ordinary least squares

(the terminology was suggested by Lovell (1993)). For simplicity, we will only talk about the corrected ordinary least squares (COLS) estimation method here. In his discussion of Farrel's original paper, Winsten (1957) suggested that the deterministic production frontier model could be estimated by COLS in two steps. In the first step ordinary least squares (OLS) is used to minimise the residual sum of squares and obtain consistent and unbiased estimates of the slope parameters and a biased estimate of the intercept parameter. In the second step the biased OLS intercept is shifted up ("corrected") to ensure that the estimated frontier bounds the data from above. The COLS intercept is estimated consistently by

$$\hat{a} = \hat{a}^* + \max\{\hat{u}_{it}\}$$

where the $\hat{u}_{it}$ are the OLS residuals and $\hat{a}^*$ is the first step OLS intercept estimator.

The COLS technique is easy to implement, and it generates an estimated production frontier that lies on or above the data.

3.3 Affine Transformed Model (NLS)

In the standard Cobb-Douglas model (equation 3.3), if there is only one input and we are willing to impose the decreasing returns to scale assumption to the function by constraining $0 < b_1 < 1$, the graph of the production frontier will look like Figure 3.1, where y-axis (Q) is the output, x-axis (X) is the input, and the optimum scale point will be equal to $(0, 0)$. As a result, the perfect competition market structure with infinite tiny firms with infinitesimal input is desirable. While if we assume the market exhibits increasing returns to scale by changing the constraint to $b_1 > 1$, the production frontier will look like Figure 3.2, where the optimum scale point will be equal to $(+\infty, +\infty)$. In this case the monopoly market structure is most sensible. Finally if we assume the constant returns to scale where $b_1 = 1$ all the time, the optimum scale point could be any point on the production frontier.

Figure 3.1: *Production Frontier with DRS*

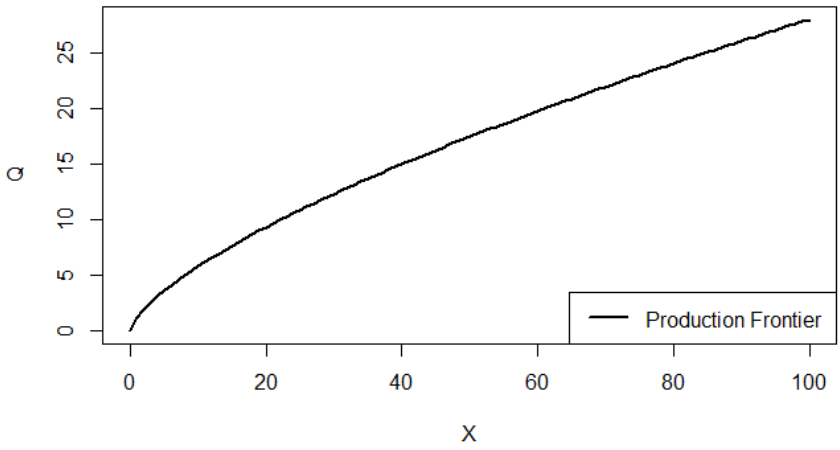
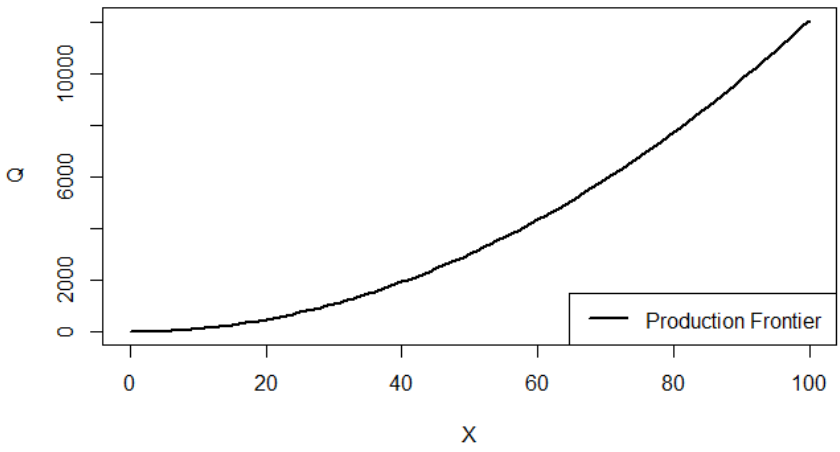


Figure 3.2: *Production Frontier with IRS*



In standard microeconomics textbooks, a firm's minimum efficient scale (MES) is defined as the lowest scale necessary for a firm to achieve the economies of scale required to operate efficiently and competitively in its industry. The average cost as a function of output normally has a "U" shape and the MES will just be the minimum of this "U" curve. However, in our case when $0 < b_1 < 1$, the average cost will keep increasing as output increases, so the MES will be zero; in case two when $b_1 > 1$, the average cost will keep decreasing as output increases, so the MES will be positive infinity; in case three when $b_1 = 1$, the average cost curve will just be a constant, so the MES can be any point. The reason for this inconsistency is due to the fact that in standard microeconomics textbooks the production curve normally has a "S" shape, which lead to a "U" shaped average cost curve. Obviously, the standard Cobb-Douglas function fails to demonstrate this characteristics.

In conclusion, the standard Cobb-Douglas model is not appropriate for the choice of scale efficiency because it is not profitable for firms to produce nothing and practically infeasible for firms to produce infinite products. Furthermore, as the TFP at optimum scale point will always be equal to infinity when $b_1 \neq 1$, the scale efficiency measure for any other firm will be equal to zero, which does not provide any helpful information at all.

In order to identify a feasible optimum scale point and therefore obtain the level of OSE, an affine transformation model is proposed. By adding M location parameters $\mu_1, \mu_2, ..., \mu_M$ into the standard Cobb-Douglas function, the new model becomes $F^t(x_{it}) = A\Pi_{m=1}^{M}(x_{mit} - \mu_m)^{b_m}$. Notice that when all of the location parameters are zero, the affine transformed model collapses to the standard model, which implies the standard model is a special case of the affine transformed model. Hence equation (3.2) can be written as

$$y_{it} = a + \Sigma_{m=1}^{M} b_m \ln(x_{mit} - \mu_m) - u_{it} \tag{3.4}$$

where $y_{it} \equiv \ln q_{it}$, $a = \ln A$ and $u_{it} \equiv -\ln OTE^t(x_{it}, q_{it})$.

Notice that here COLS estimation is no longer feasible because output does not have a linear relationship with inputs in the affine transformed models. To solve this problem, corrected non-linear least squares (CNLS) estimation is proposed.

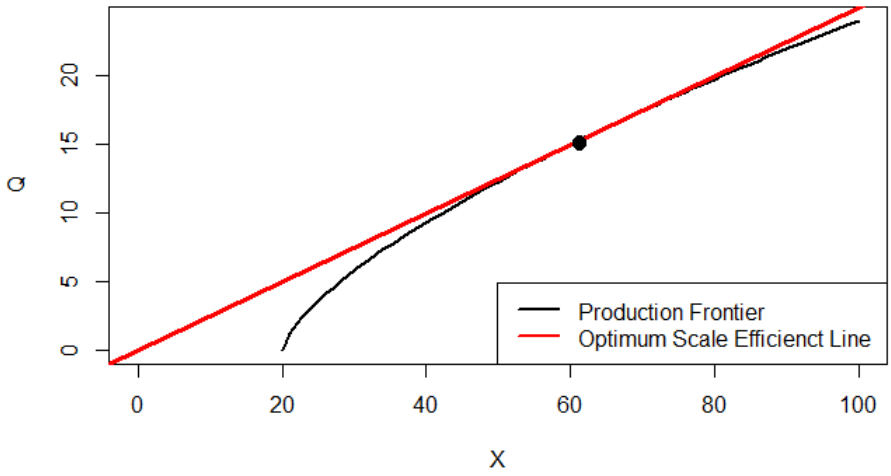
Similar to the standard model estimated by COLS before, the affine transformed deterministic production frontier model could be estimated in two steps using CNLS. In the first step non-linear least squares (NLS) is used to minimise the sum squared error and obtain consistent and unbiased estimates of the slope parameters and a biased estimate of the intercept parameter. Notice that here because there is no analytical solution for this optimum problem, numerical methods are used to solve it (See Goldfeld et al. (1972) for a more detailed explanation). In the second step the biased NLS intercept is shifted up ("corrected") to ensure that the estimated frontier bounds the data from above. The CNLS intercept is estimated consistently by

$$\hat{a} = \hat{a}^* + \max\{\hat{u}_{it}\}$$

where the $\hat{u}_i$ are the NLS residuals and $\hat{a}^*$ is the first step NLS intercept estimator.

The main merit of this affine transformation is that in this setting a meaningful optimum scale point can be easily identified. For a graphical example (Figure 3.3), in a single input production case, when $0 < b_1 < 1$, the optimum scale point, denoted as a black point, can be easily pinpointed by the red tangent line across the origin and the production frontier. Notice here the industry has increasing returns to scale up to the black point, after the black point it has decreasing returns to scale, which in turns justifies why the black point is called the optimum scale point.

Figure 3.3: *Transformed Production Frontier with DRS*



3.4 Measures of Efficiency

After obtaining the estimates of the unknown parameters in deterministic affine transformed frontiers, in this section we will conduct some post-estimation analyses which involves computing the different measures of efficiency, in particular the level of scale efficiency. We will start with the simplest case where there is only one input, then extend to two inputs and finally reach towards multiple inputs. Notice that the analytical solutions for OSE and RME are provided when the number of inputs is smaller or equal to two. When the number of inputs is greater than two, numerical methods can be used. For other measure of efficiency (OTE and TSME), analytical solutions are available in all cases.

3.4.1 Single Input

In this Section we will show how to obtain the analytical solution for OTE, OSE and TSE in a setting of single input using the affine transformed Cobb-Douglas production function form $F^t(x_{it})$ where

$$F^t(x_{it}) = A(x_{it} - \mu)^b$$

Now we want to maximise TFP by choosing the optimum input under certain restrictions for the parameters. The first constraint is $\mu < \min(x_{it})$ because we want all the production possibility points to be under the production frontier, we also want μ to be non-negative in order to satisfy the no free lunch assumption, so together we have $\min(x_{it}) > \mu \geq 0$. The second constraint is $1 > b > 0$. Since the strong disposability assumption ensures $b > 0$ (e.g., if $b < 0$, it implies when input increases, output decreases, which violates the strong disposability assumption) and a well-defined optimum scale point need $b < 1$ (e.g., if $b \geq 1$, the optimum scale point will be $(+\infty, +\infty)$, which is not well defined). The last constraint is $A > 0$, which is also due to the strong disposability of inputs and outputs assumption. Together we have the following optimisation problem:

$$\begin{aligned} &\max\ f(x_{it}) = \frac{A(x_{it} - \mu)^b}{x_{it}} \\ &\text{subject to } min(x_{it}) > \mu \geq 0, 1 > b > 0, A > 0 \end{aligned}$$

The global maximum is reached when input $x^*_{it} = \mu(1-b)^{-1}$. Notice here the maximum TFP point is a function of parameter of the production frontier only. The detailed derivations can be found in Appendix A.1.1.

Now we could construct the productivity measure and efficiency measure as a function of the parameters.

For each observation i in period t with input and output (x_{it}, q_{it}) the TFP defined in equation (2.4) will simply be:

$$TFP(x_{it}, q_{it}) = \frac{q_{it}}{x_{it}}$$

OTE defined in equation (2.7) of a specific firm i in period t will be:

$$\begin{aligned} OTE^t(x_{it}, q_{it}) =& \frac{TFP(x_{it}, q_{it})}{TFP(x_{it}, \overline{q}_{it})} \\ =& \frac{q_{it}}{\overline{q}_{it}} \\ =& \exp(-u_{it}) \end{aligned}$$

where $-u_{it} = y_{it} - a - b\ln(x_{it} - \mu)$, $a = \ln A$ and $y_{it} \equiv \ln q_{it}$. Notice that $OTE^t(x_{it}, q_{it}) = \exp(-u_{it})$ can also be obtained by simply using the definition of $u_{it} \equiv -\ln D_O^t(x_{it}, q_{it})$ and equation (2.6).

OSE defined in equation (2.8) of a specific firm i in period t will be:

$$\begin{aligned} OSE^t(x_{it}, q_{it}) &= \frac{TFP(x_{it}, \overline{q}_{it})}{TFP(\breve{x}_{it}, \breve{q}_{it})} \\ &= \frac{TFP(x_{it}, \overline{q}_{it})}{TFP(x^*_{it}, q^*_{it})} \\ &= \frac{\overline{q}_{it}/x_{it}}{q^*_{it}/x^*_{it}} \\ &= \frac{q_{it}/(\exp(-u_{it}) * x_{it})}{\exp(a + bln(\frac{\mu}{1-b} - \mu))/\mu(1-b)^{-1}} \\ &= \mu(1-b)^{-1}\exp(bln\frac{(x_{it} - \mu)(1-b)}{b\mu})\frac{1}{x_{it}} \end{aligned}$$

Noticing that since we have only one input, there is no residual mix efficiency left or $RME = 1$ all the time. As the maximum potential frontier we can obtain is the mix-restricted frontier, the possible maximum TFP point (x^*_{it}, q^*_{it}) will be equal to the point of MIOS $(\breve{x}_{it}, \breve{q}_{it})$ which explains why $TFP(\breve{x}_{it}, \breve{q}_{it}) = TFP(x^*_{it}, q^*_{it})$. Notice that only when $1 > b > 0$ and $\mu \neq 0$, we could obtain a meaningful measure of OSE. When $\mu = 0$, as in standard CD functions, $OSE^t(x_{it}, q_{it})$ is not well defined.

As we define TSME as the ratio of $TFP(x_{it}, q_{it})$ and $TFP(x^*_{it}, q^*_{it})$ in equation (2.5), in a single input and single output case, because residual mix efficiency does not play any role we could write $TSME$ as technical scale efficiency (TSE).

$$\begin{aligned} TSE^t(x_{it}, q_{it}) &= \frac{TFP(x_{it}, q_{it})}{TFP(x^*_{it}, q^*_{it})} \\ &= \frac{q_{it}}{x_{it}} / \frac{q^*_{it}}{x^*_{it}} \\ &= \mu(1-b)^{-1}\frac{\exp(y_{it} - a + bln\frac{(1-b)}{b\mu})}{x_{it}} \end{aligned}$$

After multiplying and dividing the right hand side with $\exp(-u_{it})$, rearranging the equation we can obtain

$$\begin{aligned} TSE^t(x_{it}, q_{it}) &= \exp(-u_{it}) \frac{q_{it}/\exp(-u_{it}) * x_{it}}{q^*_{it}/x^*_{it}} \\ &= \exp(-u_{it}) \frac{\overline{q}_{it}/x_{it}}{q^*_{it}/x^*_{it}} \\ &= OTE^t(x_{it}, q_{it}) OSE^t(x_{it}, q_{it}) \end{aligned}$$

The above equation justifies why we name the ratio of $TFP(x_{it}, q_{it})$ and $TFP(x^*_{it}, q^*_{it})$ as TSE in a single input case: this ratio can be exhaustively decomposed to the product of OTE and OSE.

In the same fashion, for input-oriented cases, it is easy to show that

$$TSE^t(x_{it}, q_{it}) = ITE^t(x_{it}, q_{it}) ISE^t(x_{it}, q_{it})$$

In conclusion, in the single input affine transformed case, we could easily obtain OTE, OSE and TSE as functions of parameters and observed input-output combination from the above formulas, where the estimates of the parameters could be obtained from the CNLS estimation.

3.4.2 Two Inputs

In this Section we will show how to obtain the analytical solution for OTE, OSE, RME and TSME in a setting of two inputs and one output using the affine transformed Cobb-Douglas production function form $F^t(x_{it})$ where

$$F^t(x_{it}) = A(x_{1it} - \mu_1)^{b_1} (x_{2it} - \mu_2)^{b_2}$$

In order to solve for TSME, we have to first identify the maximum TFP point. Therefore we want to maximise TFP under some necessary restrictions for the parameters. The first constraint is $\mu_1 < \min(x_{1it})$ and $\mu_2 < \min(x_{2it})$, where we want all the production possibility points below the new production frontier, we also want μ to be non-negative to satisfy the no free lunch assumption, so together we have $\min(x_{mit}) > \mu_m \geq 0$ where m can be 1 or 2. The second constraint is $b_1 > 0$, $b_2 > 0$ and $b_1 + b_2 < 1$, by which we could certainly find a meaningful TFP maximisation point. Noticing that if $b_i < 0$, that means when one of the inputs increases, output decreases holding other input fixed, which violates the strong disposability assumption. On the other hand, if $b_1 + b_2 \geq 1$, the TFP

maximisation point will be again, not well defined (the detail explanations can be found in Appendix A.1.2). The last constraint is $A > 0$, which is also due to the strong disposability of inputs and outputs assumption. Together the optimisation problem becomes

$$\begin{aligned} &\max\ f(\mathbf{x_{it}}) = \frac{A(x_{1it} - \mu_1)^{b_1}(x_{2it} - \mu_2)^{b_2}}{X(x_{1it}, x_{2it})} \\ &\text{subject to } \min(x_{mit}) > \mu_m \geq 0, 1 > b_1 + b_2 > 0, b_m > 0, A > 0 \end{aligned} \tag{3.5}$$

where m can be 1 or 2 and $X(x_{1it}, x_{2it}) = a_1 x_{1it} + a_2 x_{2it}$[1] and a_1, a_2 are weights for input one and two. Normally the weights are proxied by the average price of respective inputs.

The global maximum is reached when

$$\begin{aligned} x^*_{1it} &= \frac{b_1}{a_1}\frac{a_1\mu_1 + a_2\mu_2}{1 - b_1 - b_2} + \mu_1 \\ x^*_{2it} &= \frac{b_2}{a_2}\frac{a_1\mu_1 + a_2\mu_2}{1 - b_1 - b_2} + \mu_2 \\ X(x^*_{1it}, x^*_{2it}) &= \frac{a_1\mu_1 + a_2\mu_2}{1 - b_1 - b_2} \end{aligned}$$

The detailed derivations can be found in Appendix A.1.2.

In conclusion, we obtain the TFP maximisation point (x^*_{it},q^*_{it}) where

$$\begin{aligned} x^*_{it} &= (\frac{b_1}{a_1} * \frac{a_1\mu_1 + a_2\mu_2}{1 - b_1 - b_2} + \mu_1, \frac{b_2}{a_2} * \frac{a_1\mu_1 + a_2\mu_2}{1 - b_1 - b_2} + \mu_2) \\ q^*_{it} &= A(\frac{b_1}{a_1})^{b_1}(\frac{b_2}{a_2})^{b_2}(\frac{a_1\mu_1 + a_2\mu_2}{1 - b_1 - b_2})^{b_1+b_2} \end{aligned}$$

Next we will show how to obtain the analytical solution for the mix-invariant optimal scale (MIOS) point ($\breve{x}_{it}, \breve{q}_{it}$), so that OSE can be constructed later. Since in this problem, manager could not choose the mix of inputs freely, we can set a mix parameter $k_{it} = x_{2it}/x_{1it}$ where k_{it} is a non-negative constant as inputs are always non-negative. Notice that as the mix of inputs is fixed, when the first input is chosen, the second input will be automatically set to $x_{2it} = k_{it}x_{1it}$. Under some similar constraints as in the single input case, the optimisation problem becomes maximising TFP by choosing the first input.

[1] In general, $X(.)$ can be any nonnegative, nondecreasing, linearly-homogeneous, scalar-valued aggregator functions that satisfy $X(x_{it}) > 0$. For the sake of simplicity, it is assumed $X(.)$ to be a linear function of inputs.

$$\max f(x_{1it}) = \frac{A(x_{1it} - \mu_1)^{b_1}(k_{it}x_{1it} - \mu_2)^{b_2}}{a_1 x_{1it} + a_2 k_{it} x_{1it}} \tag{3.6}$$
$$\text{subject to } \min(x_{mit}) > \mu_m \geq 0, b_1 + b_2 > 0, b_m > 0, A > 0, k_{it} > 0$$

where m can be 1 or 2. Observing that the solution to this maximisation problem is independent of the weights of inputs and parameter A because we could factor out the positive constant $A/(a_1 + a_2 k_{it})$. After using the quadratic formula it is easy to obtain

$$x_{1it}^{-} = \frac{-\bar{b} - \sqrt{\bar{b}^2 - 4\bar{a}\bar{c}}}{2\bar{a}}$$
$$x_{1it}^{+} = \frac{-\bar{b} + \sqrt{\bar{b}^2 - 4\bar{a}\bar{c}}}{2\bar{a}}$$

where $k_{it} = x_{2it}/x_{1it}$, $\bar{a} = k_{it}(b_1 + b_2 - 1)$, $\bar{b} = \mu_1 k_{it}(1 - b_2) + \mu_2(1 - b_1)$ and $\bar{c} = -\mu_1\mu_2$. The details of this derivations can be found in Appendix A.2.

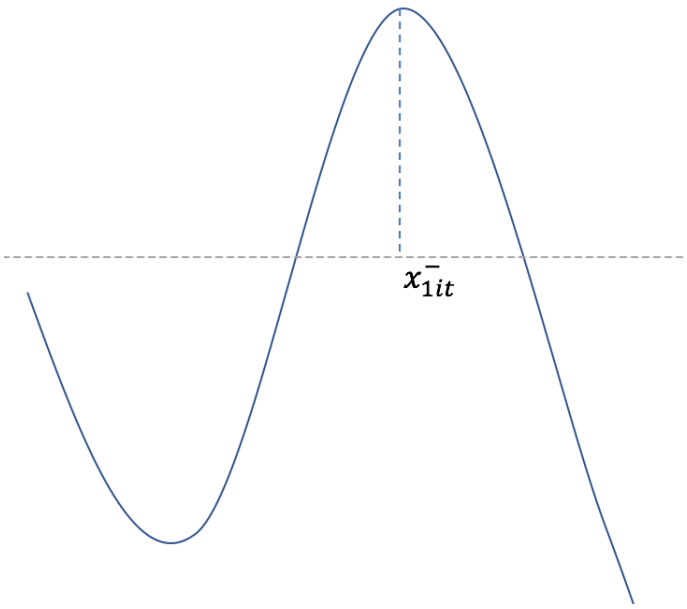
It is clear to see from the derivation below that $\bar{b}^2 - 4\bar{a}\bar{c}$ will always be non-negative under $k_{it} > 0$, $\mu_m \geq 0$ and $b_m > 0$ restrictions, so the roots of the quadratic equation always exist.

$$\begin{aligned}
\bar{b}^2 - 4\bar{a}\bar{c} &= [\mu_1 k_{it}(1 - b_2) + \mu_2(1 - b_1)]^2 + 4\mu_1\mu_2 k_{it}(b_1 + b_2 - 1) \\
&= [\mu_1 k_{it}(1 - b_2) - \mu_2(1 - b_1)]^2 + 4\mu_1\mu_2 k_{it}[(1 - b_1)(1 - b_2) + (b_1 + b_2 - 1)] \\
&= [\mu_1 k_{it}(1 - b_2) - \mu_2(1 - b_1)]^2 + 4\mu_1\mu_2 k_{it}[1 - b_1 - b_2 + b_1 b_2 + b_1 + b_2 - 1] \\
&= [\mu_1 k_{it}(1 - b_2) - \mu_2(1 - b_1)]^2 + 4\mu_1\mu_2 k_{it} b_1 b_2
\end{aligned}$$

Now let us look at the two situations when $b_1 + b_2 < 1$ and $b_1 + b_2 \geq 1$.

When $b_1 + b_2 < 1$ we can draw the graph of function $f(x)$

Figure 3.4: *Graph of* $f(x)$ *when* $b_1 + b_2 < 1$



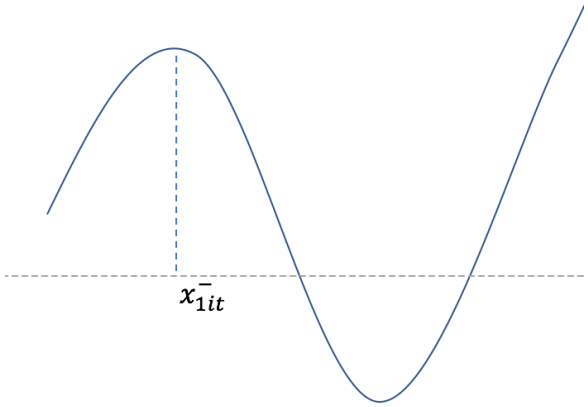
From Figure 3.4 we immediately know that $x_{1it}^{-} = (-\bar{b} - \sqrt{\bar{b}^2 - 4\bar{a}\bar{c}})/2\bar{a}$ root will give us the local maximum (as $\bar{a} < 0$) and the other root (x_{1it}^{+}) will give us the local minimum. Therefore we narrow down the candidates for the global maximum to the boundary point where x_{1it} equals to μ_1 and the local maximum at x_{1it}^{-}. As $f(\mu_1) = 0$, we have $f(x_{1it}^{-}) > 0 = f(\mu_1)$, we can safely conclude that the local maximum at x_{1it}^{-} is indeed the global maximum.

As a special case when $\mu_1 = \mu_2 = 0$, we go back to the normal Cobb-Douglas model where we know the optimum scale point is setting all inputs equal zeros. From the above formula we have $\bar{b}^2 - 4\bar{a}\bar{c} = 0$ so there is only one root and $x_{1it}^{-} = x_{1it}^{+} = 0$ and the optimum scale point becomes $(x_{1it}^{-}, x_{2it}^{-}) = (0, 0)$, which is exactly what we expect in a decreasing returns to scale case with a standard Cobb-Douglas function. The limit of $f(x_{1it})$ when x_{1it} converges to zero is positive infinity as shown below.

$$\begin{aligned}
\lim_{x_{1it}\to 0} f(x_{1it}) &= \lim_{x_{1it}\to 0} \frac{a}{a_1 + a_2 k_{it}} \frac{(x_{1it} - \mu_1)^{b_1} (k_{it} x_{1it} - \mu_2)^{b_2}}{x_{1it}} \\
&= \frac{a}{a_1 + a_2 k_{it}} \lim_{x_{1it}\to 0} k_{it}^{b_2} x_{1it}^{b_1 + b_2 - 1} \\
&= \infty
\end{aligned}$$

When $b_1 + b_2 > 1$ the graph of function $f(x)$ is in Figure 3.5.

Figure 3.5: *Graph of* $f(x)$ *when* $b_1 + b_2 > 1$



From Figure 3.5 we know the $x^{-}_{1it} = (-\bar{b} - \sqrt{\bar{b}^2 - 4\bar{a}\bar{c}})/2\bar{a}$ root will still give us the local maximum (as $\bar{a} > 0$) and the other root (x^{+}_{1it}) will give us the local minimum. However, here x^{-}_{1it} now is a negative number, which is not a legitimate value for inputs, furthermore the local maximum is not the global maximum as we can push inputs to positive infinity to obtain more scale efficiency. Hence the optimum scale point should be $(x_{1it}, x_{2it}) = (\infty, \infty)$. As a special case when $\mu_1 = \mu_2 = 0$, $f(x_{1it})$ reaches the global maximum when inputs go to infinity, which is exactly what we expect in increasing returns to scale case with a standard Cobb-Douglas function.

Finally, when $b_1 + b_2 = 1$, $f(x_{1it})$ reaches the global maximum when inputs are equal to positive infinity.

In conclusion, only when $1 > b_1 + b_2 > 0$ and at least one of the location parameters is not equal to zero, could we obtain a meaningful optimum scale point $(\breve{x}_{it}, \breve{q}_{it})$ where

$$\begin{aligned}
\breve{x}_{1it} &= \frac{-\bar{b} - \sqrt{\bar{b}^2 - 4\bar{a}\bar{c}}}{2\bar{a}} \\
\breve{x}_{2it} &= k_{it}\breve{x}_{1it} \\
\breve{q}_{it} &= A(\breve{x}_{1it} - \mu_1)^{b_1}(k_{it}\breve{x}_{1it} - \mu_2)^{b_2}
\end{aligned}$$

where $\bar{a} = k_{it}(b_1 + b_2 - 1)$, $\bar{b} = \mu_1 k_{it}(1 - b_2) + \mu_2(1 - b_1)$, $\bar{c} = -\mu_1\mu_2$ and $k_{it} = x_{2it}/x_{1it}$.

Now we can construct the productivity measure and efficiency measure as a function of the parameters.

For each observation i in period t with input and output (x_{it}, q_{it}) the TFP defined in equation (2.4) will be:

$$TFP(x_{it}, q_{it}) = \frac{q_{it}}{X(x_{1it}, x_{2it})}$$

where $X(x_{1it}, x_{2it}) = a_1 x_{1it} + a_2 x_{2it}$, and a_1, a_2 are the weights for input one and two.

OTE defined in equation (2.6) of a specific firm i in period t will be:

$$OTE^t(x_{it}, q_{it}) = \exp(-u_{it})$$

where $-u_{it} = y_{it} - a - b_1 \ln(x_{1it} - \mu_1) - b_2 \ln(x_{2it} - \mu_2))$, $a = \ln A$ and $y_{it} \equiv \ln q_{it}$.

OSE defined in equation (2.8) of a specific firm i in period t will be:

$$\begin{aligned} OSE^t(x_{it}, q_{it}) &= \frac{TFP(x_{it}, \overline{q}_{it})}{TFP(\breve{x}_{it}, \breve{q}_{it})} \\ &= \frac{\overline{q}_{it}/X(x_{it})}{\breve{q}_{it}/X(\breve{x}_{it})} \end{aligned}$$

where $X(x_{it}) = a_1 x_{1it} + a_2 x_{2it}$, $X(\breve{x}_{it}) = a_1 \breve{x}_{1it} + a_2 k_{it} \breve{x}_{1it}$, and a_1, a_2 are the weights for input one and two; $\overline{q}_{it} = A(x_{1it} - \mu_1)^{b_1}(x_{2it} - \mu_2)^{b_2}$ and $\breve{q}_{it} = A(\breve{x}_{1it} - \mu_1)^{b_1}(k_{it} * \breve{x}_{1it} - \mu_2)^{b_2}$; $\breve{x}_{1it} = (-\bar{b} - \sqrt{\bar{b}^2 - 4\bar{a}\bar{c}})/2\bar{a}$, $\bar{a} = k_{it}(b_1 + b_2 - 1)$, $\bar{b} = \mu_1 k_{it}(1 - b_2) + \mu_2(1 - b_1)$, $\bar{c} = -\mu_1 \mu_2$ and $k_{it} = x_{2it}/x_{1it}$. Notice that only when $1 > b_1 + b_2 > 0$ and at least one of the location parameters is not equal to zero, we could obtain a meaningful measure of OSE. When $\mu_1 = \mu_2 = 0$, as in standard CD functions, $OSE^t(x_{it}, q_{it})$ is not well defined.

RME defined in equation (2.10) for a specific firm i in period t is

$$\begin{aligned} RME^t(x_{it}, q_{it}) &= \frac{TFP(\breve{x}_{it}, \breve{q}_{it})}{TFP(x^*_{it}, q^*_{it})} \\ &= \frac{\breve{q}_{it}/X(\breve{x}_{it})}{q^*_{it}/X(x^*_{it})} \end{aligned}$$

where $X(x^*_{it}) = \frac{a_2\mu_2 + a_1\mu_1}{1 - b_1 - b_2}$, $q^*_{it} = A(\frac{b_1}{a_1})^{b_1}(\frac{b_2}{a_2})^{b_2}(\frac{a_1\mu_1 + a_2\mu_2}{1 - b_1 - b_2})^{b_1 + b_2}$ and $\breve{q}_{it}, X(\breve{x}_{it})$ are defined the same way in $OSE^t(x_{it}, q_{it})$.

$TSME$ defined in equation (2.5) for a specific firm i in period $t$is

$$\begin{aligned} TSME(x_{it}, q_{it}) &= \frac{TFP(x_{it}, q_{it})}{TFP(x^*_{it}, q^*_{it})} \\ &= \frac{q_{it}}{X(x_{it})} \Big/ \frac{q^*_{it}}{X(x^*_{it})} \end{aligned}$$

where $X(x_{it})$, $X(x^*_{it})$, q^*_{it}, q_{it} are defined the same way in $OSE^t(x_{it}, q_{it})$ and $RME^t(x_{it}, q_{it})$.

After multiplying and dividing the right hand side with $TFP(x_{it}, \overline{q}_{it})$ and $TFP(\breve{x}_{it}, \breve{q}_{it})$, rearranging the equation we could obtain the following result.

$$\begin{aligned} TSME(x_{it}, q_{it}) &= \frac{TFP(x_{it}, q_{it})}{TFP(x_{it}, \overline{q}_{it})} \frac{TFP(x_{it}, \overline{q}_{it})}{TFP(\breve{x}_{it}, \breve{q}_{it})} \frac{TFP(\breve{x}_{it}, \breve{q}_{it})}{TFP(x^*_{it}, q^*_{it})} \\ &= OTE^t(x_{it}, q_{it}) OSE^t(x_{it}, q_{it}) RME^t(x_{it}, q_{it}) \end{aligned}$$

The above equation justifies why we name the ratio of $TFP(x_{it}, q_{it})$ and $TFP(x^*_{it}, q^*_{it})$ as $TSME(x_{it}, q_{it})$ in multiple input case: this ratio can be exhaustively decomposed to the product of OTE , OSE and RME.

In the same fashion, for the input-oriented cases, it is easily to show that

$$TSME(x_{it}, q_{it}) = ITE^t(x_{it}, q_{it}) ISE^t(x_{it}, q_{it}) RME^t(x_{it}, q_{it})$$

To conclude, we can use the parameters of the frontier to obtain analytical solutions for the maximum TFP point (x^*_{it}, q^*_{it}) and the MIOS point $(\breve{x}_{it}, \breve{q}_{it})$, therefore to obtain the measures of OTE, OSE, RME and TSME in two inputs cases.

3.4.3 Multiple Inputs

In this Section we will show how to obtain the analytical solutions for OTE and TSME and the numerical solutions for OSE and RME in a setting of M inputs and one output using the affine transformed Cobb-Douglas production function form $F^t(x_{it})$ where

$$F^t(x_{it}) = A\Pi^M_{m=1}(x_{mit} - \mu_m)^{b_m}$$

First of all, we want to identify the possible maximum TFP point (x^*_{it},q^*_{it}) by solving the following constrained optimisation problem.

$$\begin{aligned} &\max\ f(x_{it}) = \frac{A\Pi^M_{m=1}(x_{mit} - \mu_m)^{b_m}}{X(x_{1it}, x_{2it}, ..., x_{Mit})} \\ &\text{subject to } min(x_{mit}) > \mu_m \geq 0, 1 > \Sigma^M_{m=1} b_m > 0, b_m > 0, A > 0 \end{aligned}$$

where $X(x_{1it}, x_{2it}, ..., x_{mit}) = \Sigma_{m=1}^{M} a_m x_{mit}$ and a_m is the weights for input m. The global maximum is reached when

$$x^*_{mit} = \frac{b_m}{a_m} \frac{\Sigma_{m=1}^{M}(a_m \mu_m)}{1 - (\Sigma_{m=1}^{M} b_m)} + \mu_m$$
$$X(x^*_{1it}, x^*_{2it}, ..., x^*_{Mit}) = \frac{\Sigma_{m=1}^{M}(a_m \mu_m)}{1 - (\Sigma_{m=1}^{M} b_m)}$$

See Appendix A1.1.3 for details.

Therefore we obtain the maximum TFP point (x^*_{it},q^*_{it}) where

$$x^*_{it} = (x^*_{1it}, x^*_{2it}, ..., x^*_{mit})$$
$$q^*_{it} = \Pi_{m=1}^{M} a(x^*_{mit} - \mu_m)^{b_m}$$

Notice that the two inputs maximisation problem is just a special case of this more general optimisation problem when $M = 2$.

Next we want to identify the MIOS point ($\breve{x}_{it}, \breve{q}_{it}$) so that OSE can be computed in multiple inputs cases. As the mix of inputs is fixed, there is a fixed relationship between $x_{1it}, ..., x_{mit}, ..., x_{Mit}$. As a result, we can set $k_{mit} = x_{mit}/x_{1it}$, where m can vary from 1 to M and k_{mit} is a nonnegative real number. Under some similar constraints as in the two inputs case, the optimisation problem becomes maximising TFP by choosing the first input.

$$\max f(x_{1it}) = \frac{A\Pi_{m=1}^{M}(x_{1it}k_{mit} - \mu_m)^{b_m}}{\Sigma_{m=1}^{M} a_m x_{1it} k_{mit}} \tag{3.7}$$
$$\text{subject to } \min(x_{mit}) > \mu_m \geq 0, 1 > \Sigma_{m=1}^{M} b_m > 0, b_m > 0, A > 0$$

Observing that the maximisation problem is independent with the weights of inputs and A as we could factor out $A/(\Sigma_{m=1}^{M} a_m k_{mit})$, which is always positive.

When $m = 2$, we can take the derivative of x_{1it}, and it becomes a root finding problem for a quadratic function, we can use the quadratic formula to solve this optimisation problem as we showed in Section 3.4.2. However, when m increases, it becomes a root finding problem for a m^{th} degree polynomial equation. For polynomials up to degree 4, there are explicit solution formulas similar to that for the quadratic equation (the Cardano

formulas for third-degree equations, and the Ferrari formula for degree 4). Unfortunately, for higher degrees, no general formula exists (or more precisely, no formula in terms of addition, subtraction, multiplication, division, arbitrary constants and n-th roots). This result is proved in Galois theory and is known as the Abel-Ruffini theorem (Dummit and Foote, 2004). Nevertheless, finding solutions to polynomial formulas is quite easy by using numerical methods (e.g., Newton's method). These methods are independent of the degree of the polynomial.

After solving this maximisation problem by using numerical methods, we can obtain the MIOS point $(\breve{x}_{it}, \breve{q}_{it})$. Together with the analytical solution for the maximum TFP point (x^*_{it}, q^*_{it}) we can easily construct the OTE, OSE, RME and TSME in a multiple inputs scenario.

For each observation i in period t with input and output (x_{it}, q_{it}) the TFP defined in equation (2.4) will be:

$$TFP(x_{it}, q_{it}) = \frac{q_{it}}{X(x_{it})}$$

where $X(x_{it}) = \Sigma_{m=1}^{M} a_m x_{mit}$ and a_m is the weighting for input m.

OTE defined in equation (2.6) of a specific firm i in period $t$will be:

$$OTE^t(x_{it}, q_{it}) = exp(-u_{it})$$

where $-u_{it} = y_{it} - a - \Sigma_{m=1}^{M} b_m \ln(x_{mit} - \mu_m)$, $a = \ln A$ and $y_{it} \equiv \ln q_{it}$.

OSE defined in equation (2.8) of a specific firm i in period t will be:

$$\begin{aligned} OSE^t(x_{it}, q_{it}) &= \frac{TFP(x_{it}, \overline{q}_{it})}{TFP(\breve{x}_{it}, \breve{q}_{it})} \\ &= \frac{\overline{q}_{it}/X(x_{it})}{\breve{q}_{it}/X(\breve{x}_{it})} \end{aligned}$$

where $X(x_{it}) = \Sigma_{m=1}^{M} a_m x_{mit}$, $X(\breve{x}_{it}) = \Sigma_{m=1}^{M} a_m k_{mit} \breve{x}_{1it}$, $k_{mit} = x_{mit}/x_{1it}$ and a_m is the weight for input m; $\overline{q}_{it} = A\Pi_{m=1}^{M}(x_{mit} - \mu_m)^{b_m}$ and $\breve{q}_{it} = A\Pi_{m=1}^{M}(k_{mit}\breve{x}_{1it} - \mu_m)^{b_m}$; $\breve{x}_{1it}$ is the numerical solution for problem (3.7). Notice that only when $1 > \Sigma_{m=1}^{M} b_m > 0$ and at least one of the location parameters is not equal to zero, we could obtain a meaningful measure of OSE. When $\mu_m = 0$, as in standard CD functions, $OSE^t(x_{it}, q_{it})$ is not well defined.

RME defined in equation (2.10) for a specific firm i in period $t$is

$$\begin{aligned} RME^t(x_{it}, q_{it}) &= \frac{TFP(\breve{x}_{it}, \breve{q}_{it})}{TFP(x^*_{it}, q^*_{it})} \\ &= \frac{\breve{q}_{it}/X(\breve{x}_{it})}{q^*_{it}/X(x^*_{it})} \end{aligned}$$

where $X(x^*_{it}) = \frac{\Sigma^M_{m=1}(a_m\mu_m)}{1-(\Sigma^M_{m=1}b_m)}$, $q^*_{it} = \Pi^M_{m=1}a(\frac{b_m}{a_m}\frac{\Sigma^M_{m=1}(a_m\mu_m)}{1-(\Sigma^M_{m=1}b_m)})^{b_m}$; $\breve{q}_{it}$ and $X(\breve{x}_{it})$ are defined the same way in $OSE^t(x_{it}, q_{it})$.

TSME defined in equation (2.5) for a specific firm i in period t is

$$\begin{aligned} TSME(x_{it}, q_{it}) &= \frac{TFP(x_{it}, q_{it})}{TFP(x^*_{it}, q^*_{it})} \\ &= \frac{q_{it}}{X(x_{it})} / \frac{q^*_{it}}{X(x^*_{it})} \end{aligned}$$

where $X(x_{it})$, $X(x^*_{it})$, q^*_{it}, q_{it} are defined the same way in $OSE^t(x_{it}, q_{it})$ and $RME^t(x_{it}, q_{it})$.

Another statistics we may be interested in is the elasticity of scale. In the standard model (equation 3.3), the elasticity of scale will be a constant equals to $\Sigma^M_{m=1}b_m$. However, there are slight changes in the affine transformed model as the elasticity of scale is not only a function of parameters, but also inputs.

When there are m inputs and the functional form is an affine transformed CD function (equation 3.4), the elasticity of scale η will be

$$\begin{aligned} \eta_{it} &= \Sigma^M_{m=1}\frac{\Delta\ln(q_{it})}{\Delta\ln(x_{mit})} \\ &= \Sigma^M_{m=1}\frac{\Delta\ln(q_{it})}{\Delta\ln(x_{mit}-\mu_m)}\frac{\Delta\ln(x_{mit}-\mu_m)}{x_{mit}-\mu_m}\frac{x_{mit}-\mu_m}{\Delta\ln(x_{mit})} \\ &= \Sigma^M_{m=1}b_m\frac{x_{mit}}{x_{mit}-\mu_m} \end{aligned}$$

For a single input case, the elasticity of scale will simply be $\eta_{it} = bx_{it}/(x_{it}-\mu)$.

To conclude, we can use the parameters of the frontier to obtain the analytical solutions for the maximum TFP point (x^*_{it}, q^*_{it}) and numerical solutions for the MIOS point $(\breve{x}_{it}, \breve{q}_{it})$, therefore to obtain the measures of OTE, OSE, RME and TSME in multiple inputs cases.

3.5 Conclusion

This chapter has explained how to use the deterministic frontier model, where it is assumed there is no statistical noise, to conduct the productivity and efficiency analysis. Corrected ordinary least squares methods have been first introduced to estimate the standard model (Cobb-Douglas function). Since the measure of scale efficiency is not well defined in the standard model, there is motivation for the use of a more general model (affine transformed model). An original estimation method (CNLS), which combines the ideas of both COLS and NLS methods, has been proposed to estimate this new model. Most importantly, the analytical (in some cases numerical) solutions for the measure of OTE, OSE, RME, TSME and the elasticity of scale have been derived. The next chapter will introduce the stochastic frontier model, where statistical noise is allowed.

Chapter 4

Stochastic Frontier Analysis

4.1 Introduction

Deterministic frontier estimation will yield unbiased and consistent estimators only if DFA1 and DFA2 assumptions are satisfied. However, in practice, it is very likely that these assumptions will not hold. The solution to this problem is using a new frontier estimation, namely stochastic frontier model, which exploits the fact that any unknown functional relationship can be approximated by an arbitrary function chosen by the researcher. Aigner et al. (1977) and Meeusen and van den Broeck (1977) simultaneously introduced the stochastic production frontier model. This model not only allows for technical inefficiency, but also statistical noise. Generally speaking, there are four sources of statistical noise: measurement errors in one or more variables, omission of relevant variables, inclusion of irrelevant variables and functional form errors. Depending on the sources of noise, it may not be possible to interpret the coefficients in the usual way (e.g., if noise represents functional form errors, then the coefficients in a CD function can no longer be interpreted as elasticities) and the usual normality assumption may not be tenable (e.g., if noise represents an omitted binary variable, then noise is a Bernoulli random variable). In order to introduce a normally distributed noise component without complicating the interpretation of the coefficients (and functions of the coefficients, such as the elasticity of scale), from now on, the noise is always assumed to be caused by measurement errors in the dependent variable.

Let y_{it} be the true value of logarithm of the output, when there is measurement error in the dependent variable, the observable logarithm of output is $\breve{y}_{it} = y_{it} + v_{it}$ where v_{it} represents the stochastic noise (measurement error). In this case, equation (3.2) becomes

$$\breve{y}_{it} = G^t(x_{it}) - u_{it} + v_{it} \tag{4.1}$$

where $\breve{y}_{it}$ is the observed logarithm of output, $G^t(\mathbf{x_{it}}) \equiv \ln F^t(\mathbf{x_{it}})$ is the logarithm of the frontier function, $u_{it} \equiv -\ln OTE^t(\mathbf{x_{it}}, q_{it})$ is the nonnegative output-oriented technical

inefficiency effect and $v_{it} = \breve{y}_{it} + u_{it} - G^t(x_{it})$ represents the measurement error of the dependent variable. Since now the error term has two components, the stochastic production frontier model is often referred to as a "composed error" model. The noise component v_{it}, is assumed to be independent and identically distributed ($i.i.d.$), has a symmetric shape and independent of u_{it}. Thus the overall error term $\varepsilon_{it} = v_{it} - u_{it}$ is asymmetric as $u_{it} \geq 0$.

Suppose the statistical noise v_{it} is known, it is then possible to recover the the true value y_{it} by constructing

$$y_{it} = \breve{y}_{it} - v_{it}$$

As there is no statistical noise in the above model, we could simply apply the results computed from DFA Chapter into this noiseless model to obtain measures of productivity and efficiency.

In general, the advantage of stochastic production frontier models is that it allows for all sources of statistical noise. The parameters of stochastic frontier models (SFMs) can be estimated using least squares (LS), maximum likelihood (ML) and Bayesian methods. Due to limitations of space only the ML method will be discussed here.

4.2 Standard Model (MLE)

When $F^t(x_{it})$ is a standard Cobb-Douglas function where $F^t(x_{it}) = A\Pi_m(x_{mit})^{b_m}$, equation (4.1) can be written as

$$\breve{y}_{it} = a + \Sigma_{m=1}^{M} b_m \ln(x_{mit}) - u_{it} + v_{it} \tag{4.2}$$

where $\breve{y}_{it}$ is the observed logarithm of output, $a = \ln A$, $u_{it} \equiv -\ln OTE^t(\mathbf{x_{it}}, q_{it})$ and $v_{it} = y_{it} - u_{it} - (a + \Sigma_{m=1}^{M} b_m \ln(x_{mit}))$.

Maximum likelihood (ML) estimation is used here to obtain unbiased and consistent estimates of the production technology parameters. ML estimation of the stochastic frontier models (SFM) involves choosing the unknown parameters to maximise the joint probability (i.e., likelihood) of observing the data. In order to do so we have to first differentiate the effects of the statistical noise and technical inefficiency, and this requires distributional assumptions on the two error components. To form the likelihood function, it is common to assume that

ML1 v_{it} is an independent $N(0, \sigma_v^2)$ random variable, and

ML2 u_{it} is an independent $N^+(0, \sigma_u^2)$ random variable.

ML3 $\text{Cov}(v_{it} - u_{it}, \ln x_{mit}) = 0$ for all m, i and t.

If these assumptions are true, ML estimation can be conducted by maximising the log-likelihood function, which is a function that expresses the joint probability of observing the sample data as a function of the unknown parameters.

The log-likelihood function is

$$lnL(y \mid x, \theta) = N\ln(\frac{2}{\sqrt{2\pi\sigma^2}} - \frac{1}{2\sigma^2}\Sigma_{t=1}^{T}\Sigma_{i=1}^{I}(y_{it} - G^t(\mathbf{x_{it}}))^2 + \Sigma_{t=1}^{T}\Sigma_{i=1}^{I}\ln\Phi(\frac{-\lambda(y_{it} - G^t(\mathbf{x_{it}}))}{\sigma}) \tag{4.3}$$

where $G^t(\mathbf{x_{it}}) = a + \Sigma_{m=1}^{M} b_m \ln(x_{mit})$, I_t is the number of firms in the sample in period t, $N = \Sigma_{i=1}^{I} I_t$ is the total number of observations in the sample, $\Phi(.)$ is the standard normal cumulative distributive function (CDF), $\sigma^2 = \sigma_u^2 + \sigma_v^2$, $\lambda = \frac{\sigma_u}{\sigma_v}$, and θ is a vector containing σ^2, λ and all the unknown parameters in $G^t(\mathbf{x_{it}})$.

After maximising the log-likelihood function above by using numerical methods we can obtain the estimators of the unknown parameters easily.

The last question is how good are those estimators. It turns out that if ML1-3 assumptions are true, the ML estimators for the unknown parameters are consistent and asymptotically efficient.

4.3 Affine Transformed Model (MLE)

The reason we want to use the affine transformed model in stochastic frontier estimation is exactly the same reasoning in DFA as 1) the transformed model is a more general model compared to the standard model; 2) the transformed model can identify a meaningful MIOS point thus the measure of scale efficiency.

When $F^t(x_{it})$ is the affine transformed Cobb-Douglas function where $F^t(x_{it}) = A\Pi_m(x_{mit} - \mu_m)^{b_m}$, equation (4.1) can be written as

$$y_{it} = a + \Sigma_{m=1}^{M} b_m \ln(x_{mit} - \mu_m) - u_{it} + v_{it} \tag{4.4}$$

where $y_{it} \equiv \ln q_{it}$, $a = \ln A$, $u_{it} \equiv -\ln OTE^t(x_{it}, q_{it})$ and $v_{it} = y_{it} - u_{it} - (a + \Sigma_{m=1}^{M} b_m \ln(x_{mit} - \mu_m))$.

Compared with the standard Cobb-Douglas function (equation 4.2), the new SFA introduces M location parameters $\mu_1, \mu_2, ..., \mu_M$ into the production frontier function. It is

clear to see that when all of the location parameters are zeros, the affine transformed model collapses to the standard model, which implies the standard model is a special case of the affine transformed model.

ML estimation can be implemented by maximising the log-likelihood function (4.3) exactly the same way as in Section 4.2, the only difference is that $G^t(x_{it})$ here is $G^t(x_{it}) = a + \Sigma_{m=1}^{M} b_m \ln(x_{mit} - \mu_m)$. To distinguish the MLE implemented in a standard model (MLES), we name the MLE implemented in an affine transformed model the transformed maximum likelihood estimation (MLET). Again, if ML1-3 assumptions are true, the MLET estimators for the unknown parameters are consistent and asymptotically efficient.

4.4 Measures of Efficiency

After obtaining the estimates of unknown parameters in stochastic affine transformed frontiers, we will conduct some post-estimation analyses in this section which will involve some calculations of the measure of productivity and different measures of efficiency. Assuming there are M inputs and single output and the production function is affine transformed Cobb-Douglas form $F^t(x_{it})$ where

$$F^t(x_{it}) = A\Pi_{m=1}^{M}(x_{mit} - \mu_m)^{b_m}$$

For each observation i in period t with input and output (x_{it}, q_{it}) the TFP defined in equation (2.4) will be:

$$TFP(x_{it}, q_{it}) = \frac{q_{it}}{X(x_{it})}$$

where $X(x_{it}) = \Sigma_{m=1}^{M} a_m x_{mit}$ and a_m is the weights for input m.

Next we will show how to obtain estimates of the technical efficiency for each producer.

Unlike DFA, where it is simple to obtain $OTE^t(x_{it}, q_{it}) = exp(-u_{it}) = exp(y_{it} - G^t(x_{it}))$, in SFA we have to find a way to distinguish between the technical inefficiency and statistical noise. Here for simplicity we will only introduce Jondrow et al. (1982)'s method to obtain an estimate of $OTE^t(x_{it}, q_{it})$ for firm i in time period t. The associated OTE predictor is called Jondrow, Lovell, Materov, and Schmidt (1982) (hereafter JLMS) predictor. The interested readers may also read Battese and Coelli (1988)'s method.

Jondrow et al. (1982)'s method says when ML1-2 assumptions are true, $E(u_{it} \mid e_{it})$ can be used as a predictor of u_{it} and the associated JLMS predictor for technical efficiency is consistent.

OTE defined in equation (2.6) of a specific firm i in period t will be:

$$OTE^t(x_{it}, q_{it}) = exp(-u_{it})$$

where u_{it} can be predicted by $E(u_{it} \mid e_{it})$ and

$$E(u_{it} \mid e_{it}) = \mu^*_{it} + \sigma^*(\frac{\phi(\mu^*_{it}/\sigma^*)}{\Phi(\mu^*_{it}/\sigma^*)})$$

where $\mu^*_{it} = -e_{it}\sigma_u^2/(\sigma_u^2 + \sigma_v^2)$, $e_{it} = y_{it} - G^t(x_{it})$ and $\sigma^{*2} = \sigma_v^2\sigma_u^2/(\sigma_u^2 + \sigma_v^2)$

and σ_v^2, σ_u^2 are calculated by using

$$\begin{aligned}\sigma_v^2 &= \frac{\sigma^2}{\lambda^2 + 1} \\ \sigma_u^2 &= \sigma^2 - \sigma_v^2 \\ &= \frac{\sigma^2\lambda^2}{\lambda^2 + 1}\end{aligned}$$

where σ^2, λ are defined in the log-likelihood function (4.3).

In order to construct the measures of OSE, RME and TSME, we have to first obtain an estimate of noiseless output y_{it}, which is

$$\hat{y}_{it} = \breve{y}_{it} - \hat{v}_{it}$$

where $\hat{v}_{it} = \hat{\varepsilon}_{it} + \hat{u}_{it}$, $\hat{u}_{it}$ is the standard JLMS predictor for technical efficiency and $\hat{\varepsilon}_{it}$ is the ML residual. This estimated noiseless output, together with the ML parameter estimates, can be used to construct the formulas for OSE, RME and TSME, where the procedure are exactly the same as in Chapter 3.4.

Moreover, as the stochastic model involves measurement errors in the dependent variable, here we are also interested in the stochastic noise (SN) for a specific firm i in period t, which can be defined as

$$SN(x_{it}, q_{it}) = \exp(v_{it})$$

where again $\hat{v}_{it} = \hat{\varepsilon}_{it} + \hat{u}_{it}$, $\hat{u}_{it}$ is the standard JLMS predictor for technical efficiency and $\hat{\varepsilon}_{it}$ is the ML residual.

To conclude, even when there are measurement errors in the dependent variable, by using the ML method and making certain assumptions, we can still obtain the estimators of the parameters of the likelihood function and then construct the measures of OTE, OSE, RME, SN and TSME in multiple inputs cases.

4.5 Conclusion

This chapter explains how to use the stochastic frontier model, where measurement errors are involved in the dependent variable, to conduct the productivity and efficiency analysis. ML method is used to estimate both the standard and affine transformed models. The analytical (in some cases numerical) solutions for the measures of OTE, OSE, RME, SN, and TSME have been derived. In the next chapter, a number of Monte Carlo simulations will be performed in order to compare the performance of affine transformed models and standard models.

Chapter 5

Monte Carlo Simulation and Results

5.1 Introduction

A number of Monte Carlo simulations are designed in R to first illustrate a numerical and graphical example of the affine transformed model and then to make statistical inferences of the new method before it is applied to an empirical dataset. Given that the affine transformed frontier is a new model, there is uncertainty about the performance of the CNLS estimation in the context of DFA and ML estimation in the context of SFA especially in small sample scenarios. Therefore, standard practice is followed and the new method is tested using simulated data. The simulation also facilitates comparison between a standard model estimation and an affine transformed model estimation, allowing for identification of a meaningful measure of scale efficiency. In addition, statistical efficiency improvements made by the new model have been demonstrated.

In Section 5.2, for the purpose of illustration, we will start with a single input case since it is easy to demonstrate the input-output sets in a two dimensions picture. After gaining a better understanding of the measure of technical efficiency and scale efficiency in a simple scenario, in Section 5.3, we will move to multiple inputs cases, where measures of statistical efficiency including bias and mean squared error between the standard model and the affine transformed model are compared. Section 5.4 concludes.

5.2 Illustration of Single Input Affine Transformed Model

The data generating process (DGP) in Section 5.1 involves no stochastic noise and the true functional form is an affine transformed Cobb-Douglas function. The R code for each

model is presented in Appendix B.1. A random number generation command was used to provide the same numerical results for consistency and reproducibility.

The number of samples were specified as $M = 50$ for small sample cases and it is assumed that there is no technical change. The true model is defined using the definition in equation (3.1) and $D_O^t(x_{it}, q_{it}) = \exp(-u_{it})$, together we have

$$q_{it} = F^t(x_{it})\exp(-u_{it})$$

The inefficiency was generated using a half-normal distribution:

$$u \sim \mathbb{N}^+\left(0, \sigma_u^2\right)$$
$$\text{where } \sigma_u = 0.2$$

The single input were generated using a uniform distribution:

$$x \sim \text{Uniform}\,(\mu, 10)$$

The production frontier was constructed by using the affine transformed Cobb-Douglas production function.

For single input single output case we have:

$$F^t(x_{it}) = a(x_{it} - \mu)^b$$

where

$$a = 1$$
$$b = 0.7$$
$$\mu = 2$$

Table 5.1 makes a comparison between the true model and the affine transformed model estimated by CNLS. It reports the parameters (estimators); the minimum, mean and maximum of OTE and OSE; the elasticity of scale for the first observation in the simulated data. The table shows that even in a small sample situation, the affine transformed model performs well in the sense that all of the estimators are very close to their true values.

Table 5.1: *Comparison between the true model and the affine transformed model estimated by CNLS*

	True	Transformed
a	1.000	1.000
b	0.700	0.704
μ	2.000	1.994
$OTE(min)$	0.847	0.845
$OTE(mean)$	0.858	0.855
$OTE(max)$	0.865	0.860
$OSE(min)$	0.883	0.893
$OSE(mean)$	0.903	0.913
$OSE(max)$	0.916	0.926
η_1	1.925	1.937

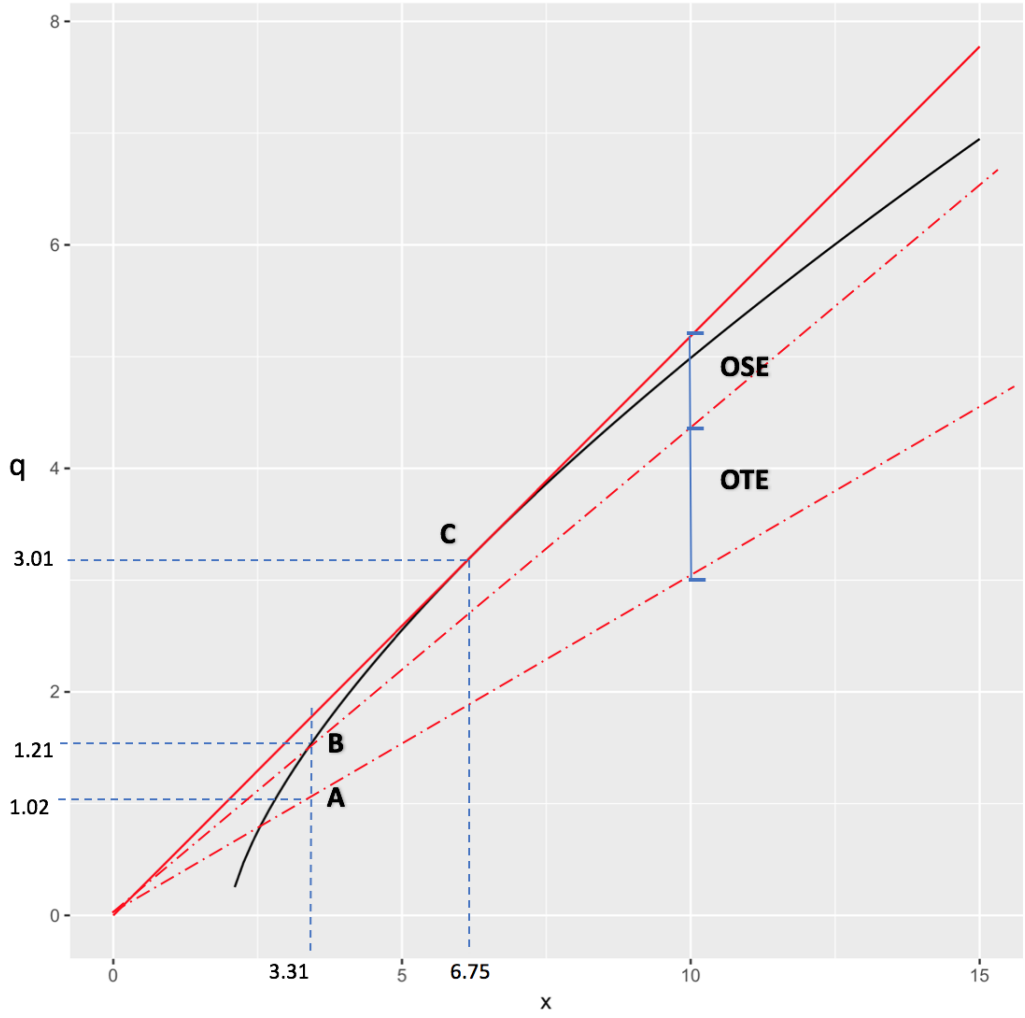
For a numerical and graphical example, consider a certain firm A which is not in the simulated data and we want to predict the measure of OTE and OSE of that firm. In Figure 5.1, Firm A is denoted by point A with $(x_A, q_A) = (3.31, 1.02)$. The estimated deterministic frontier is $\widehat{f(x_{it})} = \hat{a}(x_{it} - \hat{\mu})^{\hat{b}}$ where $\hat{a} = 1, \hat{b} = 0.704$ and $\hat{\mu} = 1.994$, which is denoted as the black solid curve in the figure. After obtaining the estimates of the parameters, it is easy to calculate the full technical efficient point B by using the frontier function and A's input and yields $(x_B, q_B) = (3.31, 1.21)$ where $q_B = \bar{q}_A = (3.31 - 1.994)^{0.704} = 1.21$. From point A to B, productivity increases because technical efficiency increases. The MIOS point is calculated by using the analytical solution discussed in Section 3.4.1 and denoted by C $(x_C, q_C) = (6.75, 3.01)$ where $x_C = \breve{x} = x^* = 1.994/(1 - 0.704) = 6.75$ and $q_C = (6.75 - 1.994)^{0.704} = 3.01$. From point B to C, productivity increases because scale efficiency increases. In Figure 5.1, the gap between points A and B is due to technical inefficiency and the gap between points B and C is due to scale inefficiency. Measures of OTE and OSE of firm A are computed by using

$$OTE(x_A, q_A) = \frac{TFP(x_A, q_A)}{TFP(x_A, \overline{q}_A)} = \frac{q_A}{\overline{q}_A} = \frac{1.02}{1.21} = 0.84$$

and

$$OSE(x_A, q_A) = \frac{TFP(x_A, \overline{q}_A)}{TFP(\breve{x}, \breve{q})} = \frac{\overline{q}_{it}/x_{it}}{q^*_{it}/x^*_{it}} = \frac{1.21/3.31}{3.01/6.75} = 0.82$$

Figure 5.1: *Demonstration of single input TFP decomposition in the transformed model*



5.3 Comparison Between Standard Model and Affine Transformed Model

The main aim in this section is to compare the performance of the affine transformed model and the standard model in both DFM and SFM frameworks. The number of samples is specified as $N = 50$ for small sample cases and $N = 500$ for sufficient sample cases, and it is assumed that there is no technical change.

The true model for DFM is

$$q_{it} = F^t(x_{it})\exp(-u_{it})$$

And the true model for SFM is

$$q_{it} = F^t(x_{it})\exp(-u_{it})\exp(v_{it})$$

The inefficiency was generated using a half-normal distribution:

$$u \sim \mathbb{N}^{+}\left(0, \sigma_u^2\right)$$
$$\text{where } \sigma_u = 0.2$$

The stochastic noise was generated using a normal distribution:

$$v \sim \mathbb{N}\left(0, \sigma_v^2\right)$$
$$\text{where } \sigma_v = 0.1$$

The inputs were generated using a uniform distribution:

$$x_1 \sim \text{Uniform}\left(\mu_1, 10\right)$$
$$x_2 \sim \text{Uniform}\left(\mu_2, 10\right)$$

where the lower bounds μ_1 and μ_2 are the location parameters.

The production frontier was constructed by using the affine transformed Cobb-Douglas production function where in the two inputs case $F^t(x_{it}) = a(x_{1it} - \mu_1)^{b_1}(x_{2it} - \mu_2)^{b_2}$. The detail settings of the parameter values will be specified in the following sections.

Finally, bias and mean squared error (MSE) were calculated for each execution of the simulation as measures of statistical efficiency:

$$\text{Bias} = \mathbb{E}\left[\hat{\theta} - \theta\right] = \mathbb{E}\left[\hat{\theta}\right] - \theta$$
$$\text{MSE} = \mathbb{E}\left[\left(\hat{\theta} - \theta\right)^2\right] = Bias^2 + Var\left(\theta\right)$$

In Section 5.3.1, the COLS and CNLS methods were compared in the framework of DFA where it is assumed there is no stochastic noise. In Section 5.3.2, the standard maximum likelihood estimation (MLES) and transformed maximum likelihood estimation (MLET) methods were compared when there are measurement errors in output.

5.3.1 Two Inputs Without Error

The data generating process (DGP) in this section involves no stochastic noise and the true function form is affine transformed Cobb-Douglas. Hence the true model is

$$q_{it} = a(x_{1it} - \mu_1)^{b_1}(x_{2it} - \mu_2)^{b_2}\exp(-u_{it})$$

where inputs and inefficiency term are generated the same way in Section 5.2 and parameters $a, b_1, b_2, \mu_1, \mu_2$ are specified in the tables below.

The main aim here is to compare the performance of affine transformed models and standard models when the true model is deterministic. Bias and MSE are used and reported as statistical efficiency measures in both small sample and large sample to compare the performance of COLS and CNLS. In addition, we want to test whether the estimators of CNLS are robust when the location parameters $\mu_1 = \mu_2 = 0$ so the affine transformed model collapses into the standard model. We also would like to know the results of the estimation when the scale parameters $b_1 + b_2 > 1$ and we expect OSE to be not well-defined in this case. Besides the parameters specified in the model, we are also interested in how precise OTE, OSE and elasticity of scale are measured. As OTE, OSE and elasticity of scale will change when observations vary in affine transformed models, without loss of generality, the minimum, mean and max of OTE and OSE and the elasticity of scale of the first observation in the simulated dataset are reported.

First of all, let us examine what happens within each model when the number of observation changes. Table 5.2 and Table 5.4 compare the performance of the estimators of the standard and affine transformed models when $\mu_1 = \mu_2 = 0$. When sample size increases from 50 to 500, both of the standard model and the affine transformed model perform better as both bias and MSE become closer to zero. It is as expected because as sample size goes up, estimators will become more precise due to more information explored. However, for the tables where location parameters are non-zero (Table 5.3 and Table 5.5), when sample size increases, even though the bias and MSE for the affine transformed model decrease, both of the two measures increase for the standard model. This implies the estimator for the affine transformed model is unbiased and consistent and the estimator for the standard model is biased and inconsistent. Furthermore, when we increase the sample size to 100,000, both of the bias and MSE of all estimates in the affine transformed model converge to zero. The bad performance of the standard model is because that in the standard setting, the location parameters are always restricted to zero, which is a false restriction when the true DGP sets $\mu_m \neq 0$. Moreover, from the tables it is easy to see that for the standard model estimation, the location estimators always have a downward bias. This is due to the fact the true location parameter $\mu_m > 0$. In addition, when $\hat{\mu}$ is downward biased, $\hat{b}$ will always be upward biased.

Table 5.2: *Comparison between standard and affine transformed models without measurement error when* $b_1 = 0.3, b_2 = 0.4, \mu_1 = \mu_2 = 0$

Estimators						
$a = 1, b_1 = 0.3, b_2 = 0.4, \mu_1 = \mu_2 = 0, \sigma_u^2 = 0.04$						
	True		Standard		Affine Transform	
	N=50	N=500	N=50	N=500	N=50	N=500
a	1.000	1.000	1.009	1.009	1.052	1.017
b_1	0.300	0.300	0.300	0.300	0.289	0.299
b_2	0.400	0.400	0.400	0.400	0.390	0.398
μ_1	0.000	0.000	0.000	0.000	0.045	0.002
μ_2	0.000	0.000	0.000	0.000	0.030	0.002
$OTE(min)$	0.852	0.848	0.846	0.840	0.845	0.838
$OTE(mean)$	0.858	0.859	0.852	0.851	0.851	0.849
$OTE(max)$	0.866	0.866	0.859	0.859	0.860	0.856
$OSE(min)$	NA	NA	NA	NA	0.397	0.196
$OSE(mean)$	NA	NA	NA	NA	0.408	0.205
$OSE(max)$	NA	NA	NA	NA	0.422	0.216
η_1	0.700	0.700	0.700	0.700	0.697	0.698

Bias					MSE				
$a = 1, b_1 = 0.3, b_2 = 0.4, \mu_1 = \mu_2 = 0, \sigma_u^2 = 0.04$					$a = 1, b_1 = 0.3, b_2 = 0.4, \mu_1 = \mu_2 = 0, \sigma_u^2 = 0.04$				
	Standard		Affine Transform			Standard		Affine Transform	
	N=50	N=500	N=50	N=500		N=50	N=500	N=50	N=500
a	0.009	0.009	0.052	0.017	a	0.000	0.000	0.001	0.000
b_1	0.000	0.000	-0.011	-0.001	b_1	0.000	0.000	0.000	0.000
b_2	0.000	0.000	-0.010	-0.002	b_2	0.000	0.000	0.000	0.000
μ_1	0.000	0.000	0.045	0.002	μ_1	0.000	0.000	0.002	0.000
μ_2	0.000	0.000	0.030	0.002	μ_2	0.000	0.000	0.001	0.000

Table 5.3: *Comparison between standard and affine transformed models without measurement error when* $b_1 = 0.3, b_2 = 0.4, \mu_1 = 1.5, \mu_2 = 3.5$

Estimators						
$a = 1, b_1 = 0.3, b_2 = 0.4, \mu_1 = 1.5, \mu_2 = 3.5$						
$\sigma_u^2 = 0.04$						
	True		Standard		Affine Transform	
	N=50	N=500	N=50	N=500	N=50	N=500
a	1.000	1.000	0.099	0.097	0.998	1.012
b_1	0.300	0.300	0.555	0.562	0.303	0.300
b_2	0.400	0.400	1.228	1.235	0.405	0.400
μ_1	1.500	1.500	0.000	0.000	1.473	1.499
μ_2	3.500	3.500	0.000	0.000	3.482	3.500
$OTE(min)$	0.852	0.848	0.732	0.697	0.843	0.837
$OTE(mean)$	0.858	0.859	0.743	0.716	0.849	0.848
$OTE(max)$	0.865	0.866	0.751	0.729	0.859	0.856
$OSE(min)$	0.852	0.853	NA	NA	0.846	0.853
$OSE(mean)$	0.868	0.869	NA	NA	0.862	0.869
$OSE(max)$	0.881	0.886	NA	NA	0.876	0.885
η_1	2.817	3.353	1.796	1.796	2.865	3.304

Bias					MSE				
$a = 1, b_1 = 0.3, b_2 = 0.4, \mu_1 = 1.5, \mu_2 = 3.5$ $\sigma_u^2 = 0.04$					$a = 1, b_1 = 0.3, b_2 = 0.4, \mu_1 = 1.5, \mu_2 = 3.5$ $\sigma_u^2 = 0.04$				
	Standard		Affine Transform			Standard		Affine Transform	
	N=50	N=500	N=50	N=500		N=50	N=500	N=50	N=500
a	-0.901	-0.903	-0.002	0.012	a	0.116	0.117	0.001	0.000
b_1	0.255	0.262	0.003	0.000	b_1	0.010	0.010	0.000	0.000
b_2	0.828	0.835	0.005	0.000	b_2	0.102	0.100	0.000	0.000
μ_1	-1.500	-1.500	-0.027	-0.001	μ_1	0.321	0.321	0.005	0.000
μ_2	-3.500	-3.500	-0.018	0.000	μ_2	1.750	1.750	0.002	0.000

Second, if we look at Table 5.2, when the true values for μ_1 and μ_2 are equal to zeros, fix the sample size to 500 and compare the bias and MSE between standard and affine transform models, we may conclude that the affine transform model performs worse than the standard model. The reason this happens is that here the standard model can be regarded as a special case of the affine transformed model where we have imposed the location restrictions into it. In this specific case the true DGP is just $\mu_1 = \mu_2 = 0$, so basically we are imposing a "true" restriction into the unrestricted model (affine transformed model) and obtain a restricted model (standard model), hence it is plausible to yield more precise and efficient estimates. However, if the restriction is not true, the standard model estimation will be biased. For example, in Table 5.3, when the true values the location parameters are $\mu_1 = 1.5, \mu_2 = 3.5$, still fix the sample size to 50 and compare the bias and MSE between standard and affine transform models, this time the affine transform model performs much better than the standard model. This is because now we are imposing a wrong restriction $\mu_1 = \mu_2 = 0$ hence the standard model estimators are all biased as we

discussed in the last paragraph. Furthermore, increasing sample size will not help in this case.

Third, we want to know whether the affine transformed estimation is still robust when the DGP sets $b_1 + b_2 > 1$. From Table 5.4 and Table 5.5, it is easy to see the estimators of the affine transformed model are always consistent and unbiased even in small sample scenarios. It is also important to notice that in Table 5.4 and Table 5.5, all of the true, standard and affine transform models produced not available (NA) for OSE. This is because OSE is not well defined for any of the models when $b_1 + b_2 > 1$ (See Chapter 3.4.2 for a more detailed explanation).

Table 5.4: *Comparison between standard and affine transformed models without measurement error when* $b_1 = 0.3, b_2 = 0.9, \mu_1 = \mu_2 = 0$

Estimators						
$a = 1, b_1 = 0.3, b_2 = 0.9, \mu_1 = \mu_2 = 0, \sigma_u^2 = 0.04$						
	True		Standard		Affine Transform	
	N=50	N=500	N=50	N=500	N=50	N=500
a	1.000	1.000	1.009	1.009	1.054	1.017
b_1	0.300	0.300	0.300	0.300	0.289	0.299
b_2	0.900	0.900	0.900	0.900	0.889	0.898
μ_1	0.000	0.000	0.000	0.000	0.045	0.002
μ_2	0.000	0.000	0.000	0.000	0.016	0.001
$OTE(min)$	0.852	0.848	0.846	0.840	0.845	0.838
$OTE(mean)$	0.859	0.859	0.852	0.851	0.851	0.849
$OTE(max)$	0.866	0.866	0.860	0.859	0.860	0.856
$OSE(min)$	NA	NA	NA	NA	NA	NA
$OSE(mean)$	NA	NA	NA	NA	NA	NA
$OSE(max)$	NA	NA	NA	NA	NA	NA
η_1	1.200	1.200	1.200	1.200	1.196	1.198

Bias					MSE				
$a = 1, b_1 = 0.3, b_2 = 0.9, \mu_1 = \mu_2 = 0, \sigma_u^2 = 0.04$					$a = 1, b_1 = 0.3, b_2 = 0.9, \mu_1 = \mu_2 = 0, \sigma_u^2 = 0.04$				
	Standard		Affine Transform			Standard		Affine Transform	
	N=50	N=500	N=50	N=500		N=50	N=500	N=50	N=500
a	0.009	0.009	0.054	0.017	a	0.000	0.000	0.001	0.000
b_1	0.000	0.000	-0.011	-0.001	b_1	0.000	0.000	0.000	0.000
b_2	0.000	0.000	-0.011	-0.002	b_2	0.000	0.000	0.000	0.000
μ_1	0.000	0.000	0.045	0.002	μ_1	0.000	0.000	0.002	0.000
μ_2	0.000	0.000	0.016	0.001	μ_2	0.000	0.000	0.001	0.000

Table 5.5: *Comparison between standard and affine transformed models without measurement error when* $b_1 = 0.3, b_2 = 0.9, \mu_1 = 1.5, \mu_2 = 3.5$

Estimators						
$a = 1, b_1 = 0.3, b_2 = 0.9, \mu_1 = 1.5, \mu_2 = 3.5$						
$\sigma_u^2 = 0.04$						
	True		Standard		Affine Transform	
	N=50	N=500	N=50	N=500	N=50	N=500
a	1.000	1.000	0.011	0.010	1.003	1.012
b_1	0.300	0.300	0.556	0.562	0.302	0.300
b_2	0.900	0.900	2.763	2.778	0.903	0.900
μ_1	1.500	1.500	0.000	0.000	1.479	1.499
μ_2	3.500	3.500	0.000	0.000	3.497	3.500
$OTE(min)$	0.851	0.848	0.624	0.606	0.843	0.837
$OTE(mean)$	0.858	0.859	0.640	0.629	0.850	0.848
$OTE(max)$	0.864	0.866	0.654	0.649	0.857	0.856
$OSE(min)$	NA	NA	NA	NA	NA	NA
$OSE(mean)$	NA	NA	NA	NA	NA	NA
$OSE(max)$	NA	NA	NA	NA	NA	NA
η_1	6.201	6.743	3.426	3.340	6.208	6.692

Bias					MSE				
$a = 1, b_1 = 0.3, b_2 = 0.9, \mu_1 = 1.5, \mu_2 = 3.5$ $\sigma_u^2 = 0.04$					$a = 1, b_1 = 0.3, b_2 = 0.9, \mu_1 = 1.5, \mu_2 = 3.5$ $\sigma_u^2 = 0.04$				
	Standard		Affine Transform			Standard		Affine Transform	
	N=50	N=500	N=50	N=500		N=50	N=500	N=50	N=500
a	-0.989	-0.990	0.003	0.012	a	0.140	0.140	0.001	0.000
b_1	0.256	0.262	0.002	0.000	b_1	0.011	0.010	0.000	0.000
b_2	1.863	1.878	0.003	0.000	b_2	0.511	0.505	0.000	0.000
μ_1	-1.500	-1.500	-0.021	-0.001	μ_1	0.321	0.321	0.004	0.000
μ_2	-3.500	-3.500	-0.003	0.000	μ_2	1.750	1.750	0.000	0.000

Fourth, the simulation results have justified the main aim of the thesis: to identify a meaningful measure of level of scale efficiency by using the affine transformed model. For all the tables above, the standard model yields NA for OSE for the reasons discussed in Chapter 3.4, while the affine transformed model does a good job. When the true location parameters are equal to zero, even through there is no meaningful OSE in the true model, since the estimator for μ_m does not exactly equal zero in the affine transformed model, it still yields the estimated value of OSE. When the location parameters are not equal to zero, both the true and affine transformed model yield a meaningful measure of OSE.

Finally, the careful reader may notice that in Table 5.3 and Table 5.5where the true location parameters are not all set to zeros, the affine transformed estimates for them are always downward biased. This is because we set a constraint for the location estimates where $\hat{\mu}_m < \min(x_{mit})$, and the DGP for x_{mit} is Uniform$(\mu_m, 10)$, hence we immediately know that $\hat{\mu}_m < \mu_m$. When the location estimate has a downward bias, the estimate for b_m will in contrast always have a upward bias, which can be seen in Table 5.3 and Table

5.5. This is not the case when the DGP for location parameters are zeros as there is also a lower bound for μ_m where $\mu_m > 0$ in order to satisfy the no free lunch assumption (Section 2.2).

5.3.2 Two Inputs With Measurement error

In this section, the true DGP is

$$q_i = a(x_{1i} - \mu_1)^{b_1}(x_{2i} - \mu_2)^{b_2}\exp(-u_i)\exp(v_i)$$

where inputs, stochastic noise and inefficiency terms are generated the way in Section 5.2 and parameters $a, b_1, b_2, \mu_1, \mu_2$ are specified in the tables below. For the reasons we discussed in Section 4.4, the stochastic noise here is treated as measurement error in output.

The main aim here is to compare the performance of the affine transformed model and the standard model when there are measurement errors in output. Bias and MSE of the estimators have been reported. The minimum, mean and max of OTE and OSE and the elasticity of scale of the first observation in the simulated dataset are also reported.

Table 5.6 to Table 5.9 compare the performance of the estimators of the standard and affine transformed models when there are measurement errors. Based on the estimation results from these tables, it is safe to draw almost the same conclusions as in DFA framework (Section 5.3.1). The conclusions are 1) as sample size goes up, the bias and MSE of the estimates of MLET converge to zeros; 2) restricted MLE perform better than unrestricted MLE when the restrictions imposed are true, and unrestricted MLE perform better than restricted MLE when the restriction are wrong; 3) the affine transformed MLE are robust to the change of the value of parameter b; 4) MLE for the standard model fails to obtain a meaningful measure of OSE while MLE for the affine transformed model succeeds; 5) the direction of the bias of affine transformed MLE is not predictable here since stochastic noise is involved in the true model.

Table 5.6: *Comparison between standard and affine transformed models with measurement error when* $b_1 = 0.3, b_2 = 0.4, \mu_1 = \mu_2 = 0$

Estimators						
$a = 1, b_1 = 0.3, b_2 = 0.4, \mu_1 = \mu_2 = 0$						
$\sigma_v^2 = 0.01, \sigma_u^2 = 0.04$						
	True		Standard		Affine Transform	
	N=50	N=500	N=50	N=500	N=50	N=500
a	1.000	1.000	0.979	0.994	1.040	1.014
b_1	0.300	0.300	0.302	0.300	0.287	0.298
b_2	0.400	0.400	0.402	0.400	0.385	0.398
μ_1	0.000	0.000	0.000	0.000	0.064	0.003
μ_2	0.000	0.000	0.000	0.000	0.037	0.004
$OTE(min)$	0.834	0.818	0.869	0.838	0.878	0.829
$OTE(mean)$	0.858	0.858	0.884	0.871	0.860	0.825
$OTE(max)$	0.878	0.888	0.895	0.895	0.877	0.859
$OSE(min)$	NA	NA	NA	NA	0.890	0.890
$OSE(mean)$	NA	NA	NA	NA	0.432	0.221
$OSE(max)$	NA	NA	NA	NA	0.461	0.264
η_1	0.700	0.700	0.704	0.700	0.688	0.699

Bias					MSE				
$a = 1, b_1 = 0.3, b_2 = 0.4, \mu_1 = \mu_2 = 0$					$a = 1, b_1 = 0.3, b_2 = 0.4, \mu_1 = \mu_2 = 0$				
$\sigma_v^2 = 0.01, \sigma_u^2 = 0.04$					$\sigma_v^2 = 0.01, \sigma_u^2 = 0.04$				
	Standard		Affine Transform			Standard		Affine Transform	
	N=50	N=500	N=50	N=500		N=50	N=500	N=50	N=500
a	-0.021	-0.006	0.040	0.014	a	0.001	0.000	0.000	0.000
b_1	0.002	0.000	-0.013	-0.002	b_1	0.000	0.000	0.000	0.000
b_2	0.002	0.000	-0.015	-0.002	b_2	0.000	0.000	0.000	0.000
μ_1	0.000	0.000	0.064	0.003	μ_1	0.000	0.000	0.000	0.000
μ_2	0.000	0.000	0.037	0.004	μ_2	0.000	0.000	0.000	0.000

Table 5.7: *Comparison between standard and affine transformed models with measurement error when* $b_1 = 0.3, b_2 = 0.4, \mu_1 = 1.5, \mu_2 = 3.5$

Estimators						
$a = 1, b_1 = 0.3, b_2 = 0.4, \mu_1 = 1.5, \mu_2 = 3.5$						
$\sigma_v^2 = 0.01, \sigma_u^2 = 0.04$						
	True		Standard		Affine Transform	
	N=50	N=500	N=50	N=500	N=50	N=500
a	1.000	1.000	0.124	0.121	0.958	0.975
b_1	0.300	0.300	0.516	0.520	0.308	0.304
b_2	0.400	0.400	1.113	1.111	0.412	0.420
μ_1	1.500	1.500	0.000	0.000	1.460	1.495
μ_2	3.500	3.500	0.000	0.000	3.470	3.476
$OTE(min)$	0.847	0.825	0.780	0.728	0.869	0.822
$OTE(mean)$	0.858	0.859	0.793	0.774	0.876	0.849
$OTE(max)$	0.868	0.889	0.807	0.817	0.886	0.875
$OSE(min)$	0.856	0.813	NA	NA	0.840	0.798
$OSE(mean)$	0.870	0.869	NA	NA	0.855	0.855
$OSE(max)$	0.882	0.921	NA	NA	0.868	0.906
η_1	2.869	1.888	1.629	1.631	2.622	1.875

Bias					MSE				
$a = 1, b_1 = 0.3, b_2 = 0.4, \mu_1 = 1.5, \mu_2 = 3.5$					$a = 1, b_1 = 0.3, b_2 = 0.4, \mu_1 = 1.5, \mu_2 = 3.5$				
$\sigma_v^2 = 0.01, \sigma_u^2 = 0.04$					$\sigma_v^2 = 0.01, \sigma_u^2 = 0.04$				
	Standard		Affine Transform			Standard		Affine Transform	
	N=50	N=500	N=50	N=500		N=50	N=500	N=50	N=500
a	-0.876	-0.879	-0.042	-0.025	a	0.085	0.086	0.001	0.000
b_1	0.216	0.220	0.008	0.004	b_1	0.006	0.005	0.000	0.000
b_2	0.713	0.711	0.012	0.020	b_2	0.059	0.056	0.000	0.000
μ_1	-1.500	-1.500	-0.040	-0.005	μ_1	0.250	0.250	0.005	0.000
μ_2	-3.500	-3.500	-0.030	-0.024	μ_2	1.361	1.361	0.003	0.000

Table 5.8: *Comparison between standard and affine transformed models with measurement error when* $b_1 = 0.3, b_2 = 0.9, \mu_1 = \mu_2 = 0$

Estimators						
$a = 1, b_1 = 0.3, b_2 = 0.9, \mu_1 = \mu_2 = 0$						
$\sigma_v^2 = 0.01, \sigma_u^2 = 0.04$						
	True		Standard		Affine Transform	
	N=50	N=500	N=50	N=500	N=50	N=500
a	1.000	1.000	0.979	0.994	1.048	0.951
b_1	0.300	0.300	0.302	0.300	0.281	0.303
b_2	0.900	0.900	0.902	0.900	0.890	0.908
μ_1	0.000	0.000	0.000	0.000	0.074	0.000
μ_2	0.000	0.000	0.000	0.000	0.020	0.001
$OTE(min)$	0.837	0.715	0.860	0.744	0.856	0.743
$OTE(mean)$	0.860	0.858	0.876	0.857	0.876	0.857
$OTE(max)$	0.883	0.962	0.893	0.935	0.892	0.934
$OSE(min)$	NA	NA	NA	NA	NA	NA
$OSE(mean)$	NA	NA	NA	NA	NA	NA
$OSE(max)$	NA	NA	NA	NA	NA	NA
η_1	1.200	1.200	1.204	1.200	1.199	1.204

Bias					MSE				
$a = 1, b_1 = 0.3, b_2 = 0.9, \mu_1 = \mu_2 = 0$					$a = 1, b_1 = 0.3, b_2 = 0.9, \mu_1 = \mu_2 = 0$				
$\sigma_v^2 = 0.01, \sigma_u^2 = 0.04$					$\sigma_v^2 = 0.01, \sigma_u^2 = 0.04$				
	Standard		Affine Transform			Standard		Affine Transform	
	N=50	N=500	N=50	N=500		N=50	N=500	N=50	N=500
a	-0.021	-0.006	0.048	-0.049	a	0.001	0.000	0.000	0.000
b_1	0.002	0.000	-0.019	0.003	b_1	0.000	0.000	0.000	0.000
b_2	0.002	0.000	-0.010	0.008	b_2	0.000	0.000	0.000	0.000
μ_1	0.000	0.000	0.074	0.000	μ_1	0.000	0.000	0.000	0.000
μ_2	0.000	0.000	0.020	0.001	μ_2	0.000	0.000	0.000	0.000

Table 5.9: *Comparison between standard and affine transformed models with measurement error when* $b_1 = 0.3, b_2 = 0.9, \mu_1 = 1.5, \mu_2 = 3.5$

Estimators						
$a = 1, b_1 = 0.3, b_2 = 0.9, \mu_1 = 1.5, \mu_2 = 3.5$						
$\sigma_v^2 = 0.01, \sigma_u^2 = 0.04$						
	True		Standard		Affine Transform	
	N=50	N=500	N=50	N=500	N=50	N=500
a	1.000	1.000	0.021	0.020	0.961	0.969
b_1	0.300	0.300	0.509	0.511	0.312	0.305
b_2	0.900	0.900	2.408	2.395	0.913	0.937
μ_1	1.500	1.500	0.000	0.000	1.425	1.494
μ_2	3.500	3.500	0.000	0.000	3.498	3.477
$OTE(min)$	0.850	0.813	0.692	0.590	0.861	0.776
$OTE(mean)$	0.858	0.859	0.719	0.696	0.870	0.840
$OTE(max)$	0.868	0.907	0.736	0.776	0.877	0.877
$OSE(min)$	NA	NA	NA	NA	NA	NA
$OSE(mean)$	NA	NA	NA	NA	NA	NA
$OSE(max)$	NA	NA	NA	NA	NA	NA
η_1	3.720	3.062	2.917	2.906	3.804	3.119

Bias					MSE				
$a = 1, b_1 = 0.3, b_2 = 0.9, \mu_1 = 1.5, \mu_2 = 3.5$					$a = 1, b_1 = 0.3, b_2 = 0.9, \mu_1 = 1.5, \mu_2 = 3.5$				
$\sigma_v^2 = 0.01, \sigma_u^2 = 0.04$					$\sigma_v^2 = 0.01, \sigma_u^2 = 0.04$				
	Standard		Affine Transform			Standard		Affine Transform	
	N=50	N=500	N=50	N=500		N=50	N=500	N=50	N=500
a	-0.979	-0.980	-0.039	-0.031	a	0.106	0.107	0.001	0.000
b_1	0.209	0.211	0.012	0.005	b_1	0.006	0.005	0.000	0.000
b_2	1.508	1.495	0.013	0.037	b_2	0.258	0.249	0.000	0.000
μ_1	-1.500	-1.500	-0.075	-0.006	μ_1	0.250	0.250	0.005	0.000
μ_2	-3.500	-3.500	-0.002	-0.023	μ_2	1.360	1.361	0.000	0.000

5.4 Conclusion

This chapter first uses a single input simulation to illustrate how to estimate the affine transformed model using CNLS. A numerical and graphical example is presented to demonstrate how to construct the measure of scale efficiency. Several simulations with multiple inputs are then performed to compare the statistical performance of the affine transformed model and the standard model. The results of the simulation show that bias and MSE values of estimators obtained using affine transformed model are substantially lower than the standard model when location parameters are not all set to zero. Most importantly, the results justify that the affine transformed model can be applied to identify a well-defined measure of scale efficiency while the standard model cannot. The next chapter will apply the affine transformed model to an empirical agriculture dataset and the main results will be reported.

Chapter 6

Empirical Illustration

6.1 Introduction

This chapter describes an empirical application of the stochastic frontier estimation of the affine transformed model discussed in Chapter 4 to panel data collected from rice producers in the Philippines. Section 6.2 explains the data collection process and describes the input and output variables included in the estimation model. Section 6.3 lists the assumptions imposed in the data structure for estimation purposes. Section 6.4 reports the estimation results. Section 6.5 concludes.

6.2 Rice Data

The International Rice Research Institute (IRRI) collected the data from 43 smallholder rice producers in the Tarlac region of the Philippines between 1990 and 1997. Details of the survey can be found in Pandey et al. (1999). The data were supplied to, and made publicly available by Coelli et al. (2005). The data are used to construct observations on the variables presented in Table 6.1. Output quantities are measured by tonnes of freshly threshed rice per year. Input quantities are area planted, labour used on the farm, fertilisers and other inputs (See Coelli et al. (2005) for a more detailed explanation). Input prices are adjusted for subsidies, taxes, and direct payments under government commodity programs. Thus, they reflect the net values of inputs to farmers and are used to construct the Lowe aggregator functions (commonly attributed to Lowe (1823)).

Table 6.1: *Variables in the Rice Dataset*

Variable	Variable (original paper)	Description
Farm	FMERCODE	farm
Year	YEARDUM	year
q	PROD	output (tonnes of freshly threshed rice)
x1	AREA	area planted (hectares)
x2	LABOR	labour (man-days of family and hired labour)
x3	NPK	fertiliser (kg of active ingredients)
x4	OTHER	other inputs (Laspeyres index = 100 for Firm 17 in 1991)
w1	AREAP	rental price of land (pesos per hectare)
w2	LABORP	labour price (pesos per hired man-day)
w3	NPKP	fertiliser price (pesos per kg of active ingredient)
w4	OTHERP	price of other inputs (implicit price index)

6.3 Assumptions

For estimation purposes, this section makes three sets of assumptions. First, outputs are assumed to be weakly disposable, and the output set is assumed to be closed and bounded. These assumptions ensure that (1) the ODF is a well-defined representation of the period-t technology, and (2) the relationship between the variables involved in the production process can be written in the form of equation (4.1). The noise component in equation (4.1) accounts for the measurement error in output. Second, it is assumed that $D_O^t(x_{it}, q_{it}) = q_{it}/F^t(x_{it})$ where $F^t(x_{it}) = A\Pi_m(x_{mit} - \mu_m)^{b_m}$. These assumptions imply that equation (4.1) can be rewritten in the form of equation (4.4). Finally, the log-area, log-other inputs, log-labour and log-NPK variables are assumed to be uncorrelated with the error in equation (4.4) as all of the inputs are chosen at the start of the production process. Notice that as this is a well-known Philippines dataset and we are not seriously interested in the characteristics of the rice producers, hence no specific diagnostic tests (e,g,. autocorrelation, heteroskedasticity and endogeneity) have been conducted. Again the main object in this chapter is just to illustrate the feasibility of the affine transformed model in the real world.

6.4 Results

Under the assumptions of ML1-ML3, consistent ML estimates are obtained in Table 6.2. The standard errors indicate that most of the coefficients are significantly different from zero. Notice that the location estimates for area (μ_1) and fertiliser (μ_4) are both equal to zero, which again shows the standard model is a special case of an affine transformed model. Furthermore, it is feasible to obtain $\hat{\mu}_4 = 0$ as farmer can produce rice without any fertiliser. The reason $\hat{\mu}_1 = 0$ maybe due to the fact that 1) the data used here only contains small households in developing country where they may produce rice within very

limited areas; 2) the approximating function used here is only valid in a range where the observed data fall hence we do not really want to generate the prediction that is far away from the mean.

All of the other estimated coefficients seem plausible. For example, the positive estimated coefficients of the log-inputs indicate that increases in inputs lead to increases in outputs, and the estimated elasticity of scale indicates that the technology exhibits decreasing returns to scale for farm 1 at period t. In addition, OTE varies from 0.317 to 0.949 and OSE varies from 0.128 to 1 (rounding result), which are both in the range from zero to one, as expected. Furthermore, when ML1-ML3 are true, as the ML estimators are consistent, asymptotically efficient and asymptotically normal, the likelihood ratio (LR) test is also asymptotically valid. Thus we can use the LR test to verify the null hypothesis that all of the location parameters are zero. The test has been conducted in R (see Appendix B.4) and the null hypothesis was rejected at the 5% level of significance. Therefore we can conclude that the affine transformed model is more appropriate than the standard model based on this particular dataset.

Table 6.2: *ML estimation with measurement error in output*

MLE Estimators			
N=334			
a	Constant	0.490**	(0.218)
b_1	Area	0.306***	(0.074)
b_2	Labour	0.274***	(0.038)
b_3	Fertiliser	0.231***	(0.042)
b_4	Other inputs	0.051***	(0.021)
μ_1	Intercept 1	0.000	(0.082)
μ_2	Intercept 2	7.997***	(0.003)
μ_3	Intercept 3	7.215**	(3.130)
μ_4	Intercept 4	0.000	(2.310)
$OTE(min)$		0.317	
$OTE(mean)$		0.727	
$OTE(max)$		0.949	
$OSE(min)$		0.128	
$OSE(mean)$		0.923	
$OSE(max)$		1.000	
η_1	Elasticity of scale (Farm 1)	0.885	

*,**,*** significance at the 10, 5 and 1 percent level, respectively.

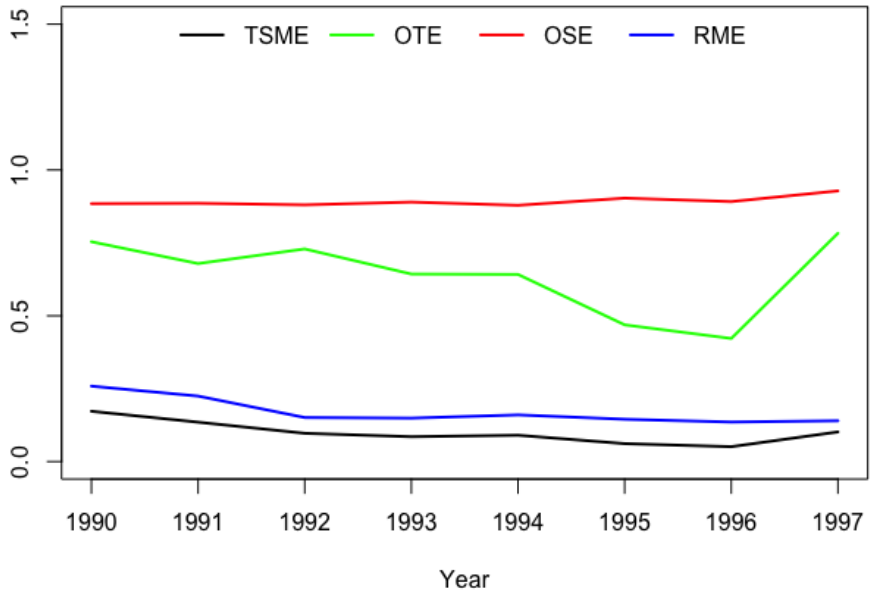
Next, let us look at the components of TFP of the first rice producer in different years.

The ML parameter estimates and the Jondrow et al. (1982) estimator of $E(u_{it} \mid e_{it})$ were used to estimate OTE, OSE, RME and TSME components on Table 6.3.

Table 6.3: *TSME decomposition of farm 1 from 1990-1997*

Year	OTE	OSE	RME	TSME
1990	0.754	0.885	0.259	0.173
1991	0.679	0.886	0.225	0.135
1992	0.729	0.88	0.151	0.097
1993	0.643	0.89	0.149	0.085
1994	0.642	0.879	0.16	0.09
1995	0.469	0.903	0.145	0.062
1996	0.422	0.892	0.136	0.051
1997	0.783	0.928	0.14	0.102

Figure 6.1: *Demonstration of TSME decomposition of farm 1 from 1990-1997*



The ML efficiency estimates reported in Table 6.3 illustrate how assumptions ML1 and ML2 can be used to disentangle the effects of technical inefficiency and statistical noise. In the case of farm 1, for example, the ML estimates indicate that the farm 1's OTE steadily declined from 0.754 in 1990 to 0.422 in 1996 and suddenly increased from 0.422 in 1996 to 0.783 in 1997. A more complete picture of TFP change in farm 1 is presented in Figure 6.1. This figure presents the estimates for measures of productivity and efficiencies for farm 1 from 1990 to 1997. This figure indicates that the main long-term driver of TFP decrease of the farm 1 has been residual-mix efficiency regress, and that an important short to medium term driver of TFP change has been technical efficiency. Scale efficiency appears to be stably increasing but have had very limited effects on TFP in this particular farm.

6.5 Conclusion

This chapter has applied the affine transformed model developed in Chapter 4 to a well-known Philippines rice dataset. Maximum likelihood estimators are used to estimate the parameters of the farm production frontier. The estimates of the parameters are then used to identify the measures of OTE, OSE, RME and TSME over time. Interpretations of the estimation results and implications for individual farmers have also been discussed. The next chapter will summarise the main findings of this thesis and points out its limitations as well as the directions for potential future research.

Chapter 7

Conclusion

7.1 Summary

As Kendrick (1961, p.1) aptly writes it, "The story of productivity, the ratio of output to input, is at the heart of the record of man's efforts to raise himself from poverty". This quote implies how important it is to understand productivity. As the level of scale efficiency can impact the magnitude of productivity, obtaining a meaningful measure of scale efficiency is also essential. This thesis has contributed to the existing productivity and efficiency literature by providing a new model that can identify the level of scale efficiency in a parametric frontier estimation framework, particularly in the case where measurement errors in outputs are involved. This thesis has also shown how TFP can be decomposed into economically-meaningful components. To illustrate, simulations have been applied to assess the statistical properties of the estimators. Finally, the paper has applied the maximum likelihood method to estimate the parameters of a rice production frontier. The estimated parameters were then used to identify the technical efficiency, scale efficiency, residual-mix efficiency components of TFP efficiency. The concepts and methods presented in this paper are relevant in most empirical contexts.

7.2 Limitations and opportunities for future research

The major limitation of this thesis is that it only shows the way to construct the measure of scale efficiency particularly to one kind of functional form. The analytical solutions for various kinds of measures of efficiency obtained in Chapter 3 and Chapter 4 are only valid when the functional form is assumed to be an affine transformed Cobb-Douglas function. Nevertheless, in practice, researchers may want to impose different functional form assumptions (e.g., generalised Leontief function, Translog production function and generalised Box-Cox function). Another limitation of this thesis is that the benefits of the affine transformed model are pronounced based on a particular simulation dataset

and a particular empirical dataset. There is, however, no guarantee that it would be true in all practical settings. Additionally, this study does not include production environment into the model. As O'Donnell (2016) suggested, characteristics of the natural and market environments are factors that can influence the output-input combinations which firms choose (e.g., farmers are unproductive when crop failures are due to low rainfall or earthquake). A more comprehensive model can be built to deal with characteristics of the production environment upfront, by accounting for the environmental factors in the period-and-environment-specific production possibilities set. The last limitation of the thesis is the difficulty to draw statistical inferences for measures of efficiency in a finite sample scenario when maximum likelihood estimation method is used. In light of these limitations, here are some of the opportunities for future experiments and researches.

As the functional form in this thesis is limited to an affine transformed Cobb-Douglas function, other flexible-functional forms and second-order series approximations of the production function could also be used to derive the formulas for measures of scale efficiency as well as other measures of efficiency. A natural question then arises: how to pick a suitable functional form in practice? In the production literature, there are many ways to choose a "good" functional form for the production process. Fuss et al. (1978) listed a set of criteria for selecting appropriate functional forms including 1) the functional form should be parsimonious in parameters; 2) the functional form should contain plausible implications and be easy for researchers to make interpretations; 3) computational requirement of the functional form should be balanced with the thoroughness of the empirical analysis. For a more detailed explanation of advantages and disadvantages of different flexible functional forms, see a survey paper by Griffin et al. (1987) and Thompson (1988). A more recent paper by Chambers et al. (2013) also discusses the process of how to choose a flexible functional form based on solutions to functional equations when homogeneity and translation invariance assumptions are satisfied.

Another direction for future research could entail constructing the formula for the input-oriented measures of scale efficiency rather than the output-oriented measure used in this thesis. In a more general case, an arbitrary choice between input and output oriented measures of efficiency can be avoided by using directional distance functions. Directional distance functions were introduced by Chambers et al. (1996) as a way of exhaustively characterising technologies with respect to both inputs and prices. Moreover, instead of dealing with the characterisation of technology that uses the input-output information (commonly referred to as the primal approach), the future researcher may also want to work with the cost, revenue and profit functions (commonly referred to as the dual approach) when the primal information (inputs and outputs) is not available but input prices are easy to obtain. There are several other benefits of the Shepherd (1953) duality theory that are often mentioned (see Varian (1992) and Färe and Primont (1995) for further discussions). After obtaining the dual form of the affine transformed model, the researcher could easily construct not only the measure of scale efficiency, but also the measures of cost, revenue and profit efficiency (see O'Donnell (2012) for a more detailed explanation).

Finally, even though we can draw reasonable statistical inferences regarding the estimators of the parameters of the affine transform model, due to the nonlinear relationship between the measures of efficiency and the parameters, it is hard to draw statistical inferences for those measures using MLE. A potential solution to this problem could be the use of the smooth bootstrap method, which estimates the standard error and the confidence interval of the efficiency measures. The future researcher could implement the smooth bootstrap in SFA similar to the method proposed by Simar and Zelenyuk (2007), who have summarised the steps for smooth bootstrap method using data envelopment analysis and free disposal hull analysis techniques. Another solution could be the use of the Bayesian method, which estimates the stochastic frontier model and conduct a standard Bayesian inference (see Griffiths et al. (2014) for a more detailed explanation).

Chapter 8

Bibliography

Afriat, S. N. (1972). Efficiency estimation of production functions, *International Economic Review* **13**(3): 568–598.

Aigner, D. J. and Chu, S. (1968). On estimating the industry production function, *The American Economic Review* **58**(4): 826–839.

Aigner, D., Lovell, C. K. and Schmidt, P. (1977). Formulation and estimation of stochastic frontier production function models, *Journal of econometrics* **6**(1): 21–37.

Balk, B. M. (2001). Scale efficiency and productivity change, *Journal of Productivity Analysis* **15**(3): 159–183.

Balk, B. M. (2013). *Industrial price, quantity, and productivity indices: the micro-economic theory and an application*, Springer Science, New York.

Battese, G. E. and Coelli, T. J. (1988). Prediction of firm-level technical efficiencies with a generalized frontier production function and panel data, *Journal of econometrics* **38**(3): 387–399.

Beattie, B. R. and Aradhyula, S. (2015). A note on threshold factor level and Stone-Geary technology, *Journal of Agricultural and Applied Economics* **47**(4): 482–493.

Chambers, R., Färe, R., Grosskopf, S. and Vardanyan, M. (2013). Generalized quadratic revenue functions, *Journal of Econometrics* **173**(1): 11–21.

Chambers, R. G., Chung, Y. and Färe, R. (1996). Benefit and distance functions, *Journal of economic theory* **70**(2): 407–419.

Coelli, T. J., Rao, D. S. P., O'Donnell, C. J. and Battese, G. E. (2005). *An introduction to efficiency and productivity analysis*, Springer Science, New York.

Debreu, G. (1951). The coefficient of resource utilization, *Econometrica: Journal of the Econometric Society* **19**(3): 273–292.

Dummit, D. S. and Foote, R. M. (2004). *Abstract algebra*, Vol. 3, Wiley Hoboken.

Färe, R. and Primont, D. (1995). *Multi-output production and duality: theory and applications*, Kluwer Academic Publishers, New York.

Farrell, M. J. (1957). The measurement of productive efficiency, *Journal of the Royal Statistical Society. Series A (General)* **120**(3): 253–290.

Fuss, M., McFadden, D., Mundlak, Y. et al. (1978). *A survey of functional forms in the economic analysis of production*, Vol. 1, Chapter 4, McMaster University Archive for the History of Economic Thought.

Geary, R. C. (1950). A note on "a constant-utility index of the cost of living", *The Review of Economic Studies* **18**(1): 65–66.

Goldfeld, S. M., Quandt, R. E. and Smallwood, D. E. (1972). *Nonlinear methods in econometrics*, North-Holland, Amsterdam.

Griffin, R. C., Montgomery, J. M. and Rister, M. E. (1987). Selecting functional form in production function analysis, *Western Journal of Agricultural Economics* pp. 216–227.

Griffiths, W., Zhang, X. and Zhao, X. (2014). Estimation and efficiency measurement in stochastic production frontiers with ordinal outcomes, *Journal of Productivity Analysis* **42**(1): 67–84.

Jondrow, J., Lovell, C. K., Materov, I. S. and Schmidt, P. (1982). On the estimation of technical inefficiency in the stochastic frontier production function model, *Journal of econometrics* **19**(2-3): 233–238.

Kendrick, J. W. (1961). *Front matter, Productivity Trends in the United States*, Princeton University Press.

Koopmans, T. C. et al. (1951). *Activity analysis of production and allocation*, John Wiley & Sons, New York.

Lovell, C. (1993). *The Measurement of Productive Efficiency: Techniques and Applications*, Oxford University Press: Oxford.

Lowe, J. (1823). *The Present State of England in Regard to Agriculture, Trade and Finance*, London: Longman, Hurst, Rees, Orme and Brown.

Meeusen, W. and van den Broeck, J. (1977). Efficiency estimation from cobb-douglas production functions with composed error, *International economic review* **18**(2): 435–444.

O'Donnell, C. (2016). Using information about technologies, markets and firm behaviour to decompose a proper productivity index, *Journal of Econometrics* **190**(2): 328–340.

O'Donnell, C. J. (2008). An aggregate quantity-price framework for measuring and decomposing productivity and profitability change, *Centre for Efficiency and Productivity Analysis Working Papers No.WP07/2008* .

O'Donnell, C. J. (2012). Nonparametric estimates of the components of productivity and profitability change in us agriculture, *American Journal of Agricultural Economics* **94**(4): 873–890.

Pandey, S., Masicat, P., Velasco, L. and Villano, R. (1999). Risk analysis of a rainfed rice production system in Tarlac, Central Luzon, Philippines, *Experimental Agriculture* **35**(2): 225–237.

Shepherd, R. W. (1953). *Cost and Production Functions*, Princeton University Press, Princeton.

Shepherd, R. W. (1970). *The Theory of Cost and Production Functions*, Princeton University Press, Princeton.

Simar, L. and Zelenyuk, V. (2007). Statistical inference for aggregates of farrell-type efficiencies, *Journal of Applied Econometrics* **22**(7): 1367–1394.

Stone, R. (1954). Linear expenditure systems and demand analysis: an application to the pattern of british demand, *The Economic Journal* **64**(255): 511–527.

Thompson, G. D. (1988). Choice of flexible functional forms: review and appraisal, *Western Journal of Agricultural Economics* pp. 169–183.

Varian, H. R. (1992). *Microeconomic Analysis*, New York, W. W. Norton.

Winsten, C. (1957). Discussion on Mr. Farrell's paper, *Journal of the Royal Statistical Society* **120**: 282–284.

Appendix A

Analytical Solution Derivation

A.1 Maximise TFP Point

A.1.1 One Input

The maximisation problem is:

$$\begin{aligned} \max\ & f(x) = \frac{a(x-\mu)^b}{x} \\ \text{subject to } & 1 > \min(x) > x \geq 0, \mu \geq 0, b > 0, a > 0 \end{aligned}$$

First Order Condition (FOC) :

$$\begin{aligned} f'(x) &= a(-x^{-2}(x-\mu)^b + bx^{-1}(x-\mu)^{b-1}) \\ \text{set } f'(x) &= 0 \\ 0 &= a(-x^{-2}(x-\mu)^b + bx^{-1}(x-\mu)^{b-1}) \\ bx^{-1}(x-\mu)^{b-1} &= x^{-2}(x-\mu)^b \\ bx &= x-\mu \\ x &= \mu(1-b)^{-1} \\ x^* &= \mu(1-b)^{-1} \end{aligned}$$

Now let us check whether the critical point is a local minimum or local maximum by calculating the second order derivative. If the second order derivative of the stationary point is smaller than zero, then the original function is concave at the stationary point, and x^* will be the local maximum, which is what we want. If the second order derivative

is a positive number, then the original function at x^* is convex, and x^* will be the local minimum.

Second Order Condition :

$$\begin{aligned} f''(x) &= a(2x^{-3}(x-\mu)^b - bx^{-2}(x-\mu)^{b-1} - bx^{-2}(x-\mu)^{b-1} + b(b-1)x^{-1}(x-\mu)^{b-2}) \\ &= a(x-\mu)^{b-2}x^{-3}(2(x-\mu)^2 - bx(x-\mu) - bx(x-\mu) + b(b-1)x^2) \end{aligned}$$

(at the stationary point) :

$$\begin{aligned} &= a(x-\mu)^{b-2}x^{-3}(2(x-\mu)^2 - 2bx(x-\mu) + b(b-1)x^2) \\ &= A * (2(x-\mu)^2 - 2bx(x-\mu) + b(b-1)x^2) \\ &= 2(x-\mu)^2 - 2bx(x-\mu) + b(b-1)x^2 \\ &= 2(\frac{b\mu}{1-b})^2 - 2(\frac{b\mu}{1-b})^2 + b(b-1)x^2 \\ &= b(b-1)(\frac{\mu}{1-b})^2 \end{aligned}$$

As the constraint for b is $(0, 1)$, implying that $b(b-1)(\mu/(1-b))^2$ will always be negative, so x^* will be the local maximum as desired. Since the function evaluated at both endpoints ($x = \mu$ & $x = +\infty$) are smaller than the value evaluated at the critical point, the local maximum is indeed the global maximum.

A.1.2 Two Inputs

The maximisation problem is:

$$\begin{aligned} &\max f(\mathbf{x_{it}}) = A(x_1 - \mu_1)^{b_1}(x_2 - \mu_2)^{b_2}/X(x_1, x_2) \\ &\text{subject to } min(x_m) > \mu_m \geq 0, 1 > b_1 + b_2 > 0, b_m > 0, A > 0 \end{aligned} \tag{A.1}$$

where $X(x_1, x_2) = a_1x_{1it} + a_2x_{2it}$, a_1, a_2 are weights for input one and two , and $m = 1$ or 2 . As a result, the above problem became

$$\max_{x_1,x_2} f(x) = \frac{a(x_1 - \mu_1)^{b_1}(x_2 - \mu_2)^{b_2}}{a_1x_1 + a_2x_2}$$

We could take the derivative of $f(x)$ in terms of x_1 and x_2 to get the stationary point

First Order Condition (FOC) :

$$\begin{aligned}
f_1'(x) &= a(x_2-\mu_2)^{b_2}\{b_1(x_1-\mu_1)^{b_1-1}(a_1x_1+a_2x_2)^{-1} \\
&\quad -(x_1-\mu_1)^{b_1}a_1(a_1x_1+a_2x_2)^{-2}\} \\
&\text{set } f_1'(x)=0 \\
0 &= a(x_2-\mu_2)^{b_2}\{b_1(x_1-\mu_1)^{b_1-1}(a_1x_1+a_2x_2)^{-1} \\
&\quad -(x_1-\mu_1)^{b_1}a_1(a_1x_1+a_2x_2)^{-2}\} \\
0 &= b_1(x_1-\mu_1)^{b_1-1}(a_1x_1+a_2x_2)^{-1}-(x_1-\mu_1)^{b_1}a_1(a_1x_1+a_2x_2)^{-2} \\
0 &= \frac{b_1}{(x_1-\mu_1)(a_1x_1+a_2x_2)}-\frac{a_1}{(a_1x_1+a_2x_2)^2} \\
0 &= b_1(a_1x_1+a_2x_2)-a_1(x_1-\mu_1)
\end{aligned}$$

similarly we have

$$\begin{aligned}
f_2'(x) &= 0 \\
0 &= b_2(a_1x_1+a_2x_2)-a_2(x_2-\mu_2)
\end{aligned}$$

two equations two unknownswe can solve

$$\begin{aligned}
x_1^* &= \frac{a_2\mu_2b_1}{a_1(1-b_1-b_2)}+\frac{\mu_1(1-b_2)}{1-b_1-b_2} \\
&= \frac{b_1}{a_1}\frac{a_1\mu_1+a_2\mu_2}{1-b_1-b_2}+\mu_1 \\
x_2^* &= \frac{a_1\mu_1b_2}{a_2(1-b_1-b_2)}+\frac{\mu_2(1-b_1)}{1-b_1-b_2} \\
&= \frac{b_2}{a_2}\frac{a_1\mu_1+a_2\mu_2}{1-b_1-b_2}+\mu_2
\end{aligned}$$

hence

$$\begin{aligned}
X^* &= a_1x_1^*+a_2x_2^* \\
&= \frac{a_2\mu_2b_1}{1-b_1-b_2}+\frac{a_1\mu_1(1-b_2)}{1-b_1-b_2}+\frac{a_1\mu_1b_2}{1-b_1-b_2}+\frac{a_2\mu_2(1-b_1)}{1-b_1-b_2} \\
&= \frac{a_1\mu_1+a_2\mu_2}{1-b_1-b_2}
\end{aligned}$$

Noticing here if $b_1+b_2\geq 1$, then the aggregator input will be either positive infinity or a negative number, which is not meaningful in real life. It is easy to check the object function is concave by calculating the hessian matrix. Hence we conclude the stationary point we have obtained is the global maximum.

A.1.3 Multiple Inputs

The maximisation problem is:

$$\max f(\mathbf{x_{it}}) = \frac{A * \Pi_{m=1}^{M}(x_{mit} - \mu_m)^{b_m}}{\Sigma_{m=1}^{M} a_m x_{mit}}$$
$$\text{subject to } \min(x_{mit}) > \mu_m \geq 0, 1 > \Sigma_{m=1}^{M} b_m > 0, b_m > 0, A > 0$$

We could take the partial derivative of $f(x)$ to get the stationary point

First Order Condition (FOC) :

$$f_j'(x) = \Pi_{i \neq j} a(x_i - \mu_i)^{b_i} \left(\frac{a b_j (x_j - \mu_j)^{b_j - 1}}{\Sigma_{i=1}^{k} a_i x_i} - \frac{a_j a (x_j - \mu_j)^{b_j}}{(\Sigma_{i=1}^{k} a_i x_i)^2}\right)$$

$$\text{set } f_j'(x) = 0$$

$$0 = \frac{a b_j (x_j - \mu_j)^{b_j - 1}}{\Sigma_{i=1}^{k} a_i x_i} - \frac{a_j a (x_j - \mu_j)^{b_j}}{(\Sigma_{i=1}^{k} a_i x_i)^2}$$

$$0 = b_j \Sigma_{i=1}^{k} a_i x_i - a_j * (x_j - \mu_j)$$

hence

$$a_j x_j = b_j \Sigma_{i=1}^{k} a_i x_i + a_j \mu_j$$

hence

$$\Sigma_{j=1}^{k} a_j x_j = (\Sigma_{j=1}^{k} b_j) \Sigma_{i=1}^{k} a_i x_i + \Sigma_{j=1}^{k} (a_j \mu_j)$$

as

$$X^* = \Sigma_{i=1}^{k} a_i x_i$$
$$= \Sigma_{j=1}^{k} a_j x_j$$
$$X^* = (\Sigma_{j=1}^{k} b_j) X + \Sigma_{j=1}^{k} (a_j \mu_j)$$

Hence

$$X^* = \frac{\Sigma_{j=1}^{k} (a_j \mu_j)}{1 - (\Sigma_{j=1}^{k} b_j)}$$

Furthermore

$$x_j^* = \frac{b_j \Sigma_{i=1}^{k} a_i x_i}{a_j} + \mu_j$$
$$= \frac{b_j X}{a_j} + \mu_j$$
$$= \frac{b_j}{a_j} \frac{\Sigma_{j=1}^{k} (a_j \mu_j)}{1 - (\Sigma_{j=1}^{k} b_j)} + \mu_j$$
$$q^* = \Pi_{j=1}^{k} a (x_j^* - \mu_j)^{b_j}$$

hence we can find analytical solution for the maximum TFP point (x_{it}^*, q_{it}^*) in multiple inputs case. Notice that the two inputs maximisation problem is just a special case of this

more general optimisation problem when we have $k = 2$.

A.2 Optimum Scale Point (Two Inputs)

The maximisation problem is:

$$\max f(x_1) = \frac{A(x_1 - \mu_1)^{b_1}(kx_1 - \mu_2)^{b_2}}{a_1 x_1 + a_2 k x_1}$$
$$\text{subject to } \; min(x_{mit}) > \mu_m \geq 0, b_1 + b_2 > 0, b_m > 0, A > 0, k > 0$$

First Order Condition (FOC) :

$$f'(x_1) = \frac{A}{a_1 + a_2 k} * b_1(x_1 - \mu_1)^{b_1} x_1^{-2}(kx_1 - \mu_2)^{b_2}\{\frac{b_1}{x_1 - \mu_1}x_1 + \frac{kb_2}{kx_1 - \mu_2}x_1 - 1\}$$

set $f'(x_1) = 0$

$$0 = (x_1 - \mu_1)^{b_1} x_1^{-2}(kx_1 - \mu_2)^{b_2}\{\frac{b_1}{x_1 - \mu_1}x_1 + \frac{kb_2}{kx_1 - \mu_2}x_1 - 1\}$$

$$0 = \frac{b_1}{x_1 - \mu_1}x_1 + \frac{kb_2}{kx_1 - \mu_2}x_1 - 1$$

$$= x_1^2 k(b_1 + b_2 - 1) + x_1(\mu_2 + \mu_1 k - b_1\mu_2 - kb_2\mu_1) - \mu_1\mu_2$$

set

$$\bar{a} = k(b_1 + b_2 - 1)$$
$$\bar{b} = \mu_2 + \mu_1 k - b_1\mu_2 - kb_2\mu_1 = \mu_1 k(1 - b_2) + \mu_2(1 - b_1)$$
$$\bar{c} = -\mu_1\mu_2$$

use quadratic formula we have

$$x_1^* = \frac{-\bar{b} \pm \sqrt{\bar{b}^2 - 4\bar{a}\bar{c}}}{2\bar{a}}$$

and

$$x_2^* = kx_1^*$$

It is clear to see from the below deduction that $\bar{b}^2 - 4\bar{a}\bar{c}$ will always be non-negative under $k > 0$, $\mu_i \geq 0$ and $b_i > 0$ restriction so the roots of the quadratic will equation always exist.

$$
\begin{aligned}
\bar{b}^2 - 4\bar{a}\bar{c} &= [\mu_1 k(1-b_2) + \mu_2(1-b_1)]^2 + 4\mu_1\mu_2 k(b_1+b_2-1) \\
&= [\mu_1 k(1-b_2) - \mu_2(1-b_1)]^2 + 4\mu_1\mu_2 k[(1-b_1)(1-b_2) + (b_1+b_2-1)] \\
&= [\mu_1 k(1-b_2) - \mu_2(1-b_1)]^2 + 4\mu_1\mu_2 k[1-b_1-b_2+b_1b_2+b_1+b_2-1] \\
&= [\mu_1 k(1-b_2) - \mu_2(1-b_1)]^2 + 4\mu_1\mu_2 k b_1 b_2
\end{aligned}
$$

Appendix B

R code

This appendix provides the R code used for the Monte Carlo simulation in Section 5 and empirical illustration in Section 6.

B.1 Single Input and Single Output

```
### OLS & NLS; M=50 ; mu1 = mu2 = 0#####
rm(list=ls())
a=1
lna=log(a)
b1=0.7
mu1=2
sigmau =0.2
M= 50 # sample size
N= 1000 # number of repetition
set.seed(101)
max = 10 # maximum of x1 & x2
nlsmatrix = matrix(NA,N,3)
OTEtrue = matrix(NA,N,M)
OTEnls = matrix(NA,N,M)
etanls = matrix(NA,N,M)
etatrue = matrix(NA,N,M)
OSEtrue = matrix(NA,N,M)
OSEnls = matrix(NA,N,M)
for (i in 1:N){
  x1 = runif(M,mu1*1.0001,max)
  u = abs(rnorm(M, mean = 0, sd = sigmau))
  q1=a*(x1-mu1)^b1
  y=log(q1)-u
```

```
q=exp(y)

# NLS
TF = try(nls(y~astar+b1*log(x1-mu1),
             start=c(astar=0.1,b1=0.3,mu1=0),
             algorithm="port", lower=c(-Inf,0,0),
             upper=c(Inf,Inf,0.999999*min(x1)),
             control = list(maxiter = 5000)), silent = T)
if (class(TF)=="try-error"){

} else {

  nls = TF
  lna=coef(nls)[1]+max(residuals(nls))
  nlsmatrix[i,] = cbind(exp(lna),coef(nls)[2],coef(nls)[3])

  # OTE & True OTE
  OTE=exp(residuals(nls)-max(residuals(nls)))
  OTEt = exp(-u)    #True OTE
  OTEtrue[i,] = OTEt
  OTEnls[i,] = OTE

  # eta & True eta
  etanls[i,] = coef(nls)[2] * x1[1]/(x1[1]-coef(nls)[3])
  etatrue[i,] = b1 * x1[1]/(x1[1]-mu1)

  # OSE & True OSE
  qbar=q/OTE
  qbarslope=qbar/x1
  xstar=coef(nls)[3]/(1-coef(nls)[2])
  qbarslopestar=exp(0+b1*log(xstar-mu1))/xstar
  OSE = qbarslope/qbarslopestar
  OSEnls[i,]=OSE

  qbar=q/OTEt
  qbarslope=qbar/x1
  xstar=mu1/(1-b1)
  qbarslopestar=exp(lna+coef(nls)[2]*
                    log(xstar-coef(nls)[3]))/xstar
  OSEt = qbarslope/qbarslopestar
  OSEtrue[i,]=OSEt
```

```
  }}

# transform
nlsmatrix = na.omit(nlsmatrix)
OTEtrue = na.omit(OTEtrue)
OTEnls = na.omit(OTEnls)
etanls = na.omit(etanls)
etatrue = na.omit(etatrue)
OSEtrue = na.omit(OSEtrue)
OSEnls = na.omit(OSEnls)

# Combine
OTEtruereuslt=colMeans(OTEtrue)
OTEnlsreuslt=colMeans(OTEnls)
etanlsreuslt=colMeans(etanls)
etatruereuslt=colMeans(etatrue)
OSEtruereuslt=colMeans(OSEtrue)
OSEnlsreuslt=colMeans(OSEnls)


# nls results combine
nlsmatrixresult = t(colMeans(nlsmatrix))
nlsresult = cbind(nlsmatrixresult,min(OTEnlsreuslt),
                  mean(OTEnlsreuslt),max(OTEnlsreuslt),
                  min(OSEnlsreuslt),mean(OSEnlsreuslt),
                  max(OSEnlsreuslt),etanlsreuslt[1])

# True value combine
TrueValue=cbind(a,b1,mu1)
trueresult = cbind(TrueValue,min(OTEtruereuslt),
                   mean(OTEtruereuslt),max(OTEtruereuslt),
                   min(OSEtruereuslt),mean(OSEtruereuslt),
                   max(OSEtruereuslt),etatruereuslt[1])

# compare
compare = rbind(trueresult,nlsresult)
compare


# Bias & MSE
biasols = colMeans(olsmatrix-TVmatrix)
```

```
MSEols = colSums((olsmatrix - TVmatrix)^2)/length(olsmatrix)

biasnls = colMeans(na.omit(nlsmatrix-TVmatrix))
MSEnls = colSums((na.omit(nlsmatrix - TVmatrix))^2)/length(nlsmatrix)

# Output
TrueValue=TrueValue[1,]
obs1 = cbind(TrueValue, olsresult, nlsresult, biasols, MSEols,
             biasnls, MSEnls)
olsresult-biasols-TrueValue
nlsresult-biasnls-TrueValue
```

B.2 Two Inputs Without Measurement Error

```
### OLS & NLS; M=50 ; mu1 = mu2 = 0#####
rm(list=ls())
a=1
lna=log(a)
b1=0.3
b2=0.4
mu1=0
mu2=0
sigmav= 0.1
sigmau =0.2
lamda= sigmau/sigmav
sigma=sigmav^2+sigmau^2
M= 50 # sample size
N= 1000 # number of repetition
set.seed(101)
max = 10 # maximum of x1 & x2
olsmatrix = matrix(0,N,7)
nlsmatrix = matrix(NA,N,7)
OTEtrue = matrix(0,N,1)
etatrue = matrix(0,N,1)
for (i in 1:N){
  x1 = runif(M,mu1*1.0001,max)
  x2 = runif(M,mu2*1.0001,max)
  k =x2/x1
  v = rnorm(M, mean = 0, sd = sigmav)
  u = abs(rnorm(M, mean = 0, sd = sigmau))
  q1=a*(x1-mu1)^b1*(x2-mu2)^b2
  y=log(q1)-u
```

```
q=exp(y)
# OLS
x = cbind(log(x1),log(x2))
ols=lm(y~x)
aols=coef(ols)[1]+max(residuals(ols))
OTE=exp(residuals(ols)-max(residuals(ols)))
OTE1=OTE[1]
OTEt = exp(-u)    #True OTE
OTEt1=OTEt[1]     #True OTE for observation1
eta1 = coef(ols)[2] * mean(x1)/(mean(x1)-mu1)
eta2 = coef(ols)[3] * mean(x2)/(mean(x2)-mu2)
eta = eta1 + eta2 #eta estimates
olsmatrix[i,] = cbind(exp(aols),coef(ols)[2],
                      coef(ols)[3],0,0,OTE1,eta)

# True eta and OTE1
OTEtrue[i,] =OTEt1
etatrue[i,] = b1 * mean(x1)/(mean(x1)-mu1)+b2*
  mean(x2)/(mean(x2)-mu2)

# NLS
TF = try(nls(y~astar+b1*log(x1-mu1)+b2*log(x2-mu2),
             start=c(astar=0.1,b1=0.3,b2=0.4,mu1=0,mu2=0),
             algorithm="port", lower=c(-Inf,0,0,0,0),
             upper=c(Inf,Inf,Inf,0.999999*min(x1),
                     0.999999*min(x2)),
             control = list(maxiter = 5000)), silent = T)
if (class(TF)=="try-error"){

} else {

  nls = TF
  lna=coef(nls)[1]+max(residuals(nls))
  OTE=exp(residuals(nls)-max(residuals(nls)))
  OTE1=OTE[1]
  eta1 = coef(nls)[2] * mean(x1)/(mean(x1)-coef(nls)[4])
  eta2 = coef(nls)[3] * mean(x2)/(mean(x2)-coef(nls)[5])
  eta = eta1 + eta2
  nlsmatrix[i,] = cbind(exp(lna),coef(nls)[2],
                        coef(nls)[3],coef(nls)[4],
                        coef(nls)[5],OTE1,eta)
```

```
  }}
# ols results combine
olsmatrix = subset(olsmatrix, olsmatrix[,1] !=0)
olsresult = colMeans(olsmatrix)

# True
OTEtrue = subset(OTEtrue,OTEtrue!=0)
OTE1true = colMeans(OTEtrue)
etatrue = subset(etatrue,etatrue!=0)
etameantrue = colMeans(etatrue)
TrueValue=cbind(a,b1,b2,mu1,mu2,OTE1true,etameantrue)

# nls results combine
nlsmatrix1 = subset(nlsmatrix, nlsmatrix[,1] !=0)
nlsresult = colMeans(nlsmatrix1)

# compare
compare = rbind(TrueValue,olsresult,nlsresult)
compare
TVmatrix = cbind(a,b1,b2,mu1,mu2,OTEtrue,etatrue)

# Bias & MSE
biasols = colMeans(olsmatrix-TVmatrix)
MSEols = colSums((olsmatrix - TVmatrix)^2)/length(olsmatrix)

biasnls = colMeans(na.omit(nlsmatrix-TVmatrix))
MSEnls = colSums((na.omit(nlsmatrix - TVmatrix))^2)/length(nlsmatrix)

# Output
TrueValue=TrueValue[1,]
obs1 = cbind(TrueValue,olsresult,nlsresult,
             biasols,MSEols,biasnls,MSEnls)
olsresult-biasols-TrueValue
nlsresult-biasnls-TrueValue

### OLS & NLS; M=500 ; mu1 = mu2 = 0#####
rm(list=setdiff(ls(), "obs1"))
a=1
lna=log(a)
b1=0.3
b2=0.4
```

```
mu1=0
mu2=0
sigmav= 0.1
sigmau =0.2
lamda= sigmau/sigmav
sigma=sigmav^2+sigmau^2
M= 500 # sample size
N= 1000 # number of repetition
set.seed(101)
max = 10 # maximum of x1 & x2
olsmatrix = matrix(0,N,7)
nlsmatrix = matrix(NA,N,7)
OTEtrue = matrix(0,N,1)
etatrue = matrix(0,N,1)
for (i in 1:N){
  x1 = runif(M,mu1*1.0001,max)
  x2 = runif(M,mu2*1.0001,max)
  k =x2/x1
  v = rnorm(M, mean = 0, sd = sigmav)
  u = abs(rnorm(M, mean = 0, sd = sigmau))
  q1=a*(x1-mu1)^b1*(x2-mu2)^b2
  y=log(q1)-u
  q=exp(y)
  # OLS
  x = cbind(log(x1),log(x2))
  ols=lm(y~x)
  aols=coef(ols)[1]+max(residuals(ols))
  OTE=exp(residuals(ols)-max(residuals(ols)))
  OTE1=OTE[1]
  OTEt = exp(-u)    #True OTE
  OTEt1=OTEt[1]     #True OTE for observation1
  eta1 = coef(ols)[2] * mean(x1)/(mean(x1)-mu1)
  eta2 = coef(ols)[3] * mean(x2)/(mean(x2)-mu2)
  eta = eta1 + eta2 #eta estimates
  olsmatrix[i,] = cbind(exp(aols),coef(ols)[2],
                        coef(ols)[3],0,0,OTE1,eta)

  # True eta and OTE1
  OTEtrue[i,] =OTEt1
  etatrue[i,] = b1 * mean(x1)/(mean(x1)-mu1)+b2 *
    mean(x2)/(mean(x2)-mu2)
```

```
  # NLS
  TF = try(nls(y~astar+b1*log(x1-mu1)+b2*log(x2-mu2),
               start=c(astar=0.1,b1=0.3,b2=0.4,mu1=0,mu2=0),
               algorithm="port", lower=c(-Inf,0,0,0,0),
               upper=c(Inf,Inf,Inf,0.999999*min(x1),0.999999*min(x2)),
               control = list(maxiter = 5000)), silent = T)
  if (class(TF)=="try-error"){

  } else {

    nls = TF
    lna=coef(nls)[1]+max(residuals(nls))
    OTE=exp(residuals(nls)-max(residuals(nls)))
    OTE1=OTE[1]
    eta1 = coef(nls)[2] * mean(x1)/(mean(x1)-coef(nls)[4])
    eta2 = coef(nls)[3] * mean(x2)/(mean(x2)-coef(nls)[5])
    eta = eta1 + eta2
    nlsmatrix[i,] = cbind(exp(lna),coef(nls)[2],coef(nls)[3],
                          coef(nls)[4],coef(nls)[5],OTE1,eta)


  }}
# ols results combine
olsmatrix = subset(olsmatrix, olsmatrix[,1] !=0)
olsresult = colMeans(olsmatrix)

# True
OTEtrue = subset(OTEtrue,OTEtrue!=0)
OTE1true = colMeans(OTEtrue)
etatrue = subset(etatrue,etatrue!=0)
etameantrue = colMeans(etatrue)
TrueValue=cbind(a,b1,b2,mu1,mu2,OTE1true,etameantrue)

# nls results combine
nlsmatrix1 = subset(nlsmatrix, nlsmatrix[,1] !=0)
nlsresult = colMeans(nlsmatrix1)

# compare
compare = rbind(TrueValue,olsresult,nlsresult)
compare
TVmatrix = cbind(a,b1,b2,mu1,mu2,OTEtrue,etatrue)
```

```
# Bias & MSE
biasols = colMeans(olsmatrix-TVmatrix)
MSEols = colSums((olsmatrix - TVmatrix)^2)/length(olsmatrix)

biasnls = colMeans(na.omit(nlsmatrix-TVmatrix))
MSEnls = colSums((na.omit(nlsmatrix - TVmatrix))^2)/length(nlsmatrix)

# Output
TrueValue=TrueValue[1,]
obss = cbind(TrueValue, olsresult, nlsresult, biasols,
             MSEols, biasnls, MSEnls)
obs1 = rbind(obs1, obss)
olsresult-biasols-TrueValue
nlsresult-biasnls-TrueValue


### OLS & NLS; M=50 ; mu1 = 1.5 mu2 = 3.5#####
rm(list=setdiff(ls(), "obs1"))
a=1
lna=log(a)
b1=0.3
b2=0.4
mu1=1.5
mu2=3.5
sigmav= 0.1
sigmau =0.2
lamda= sigmau/sigmav
sigma=sigmav^2+sigmau^2
M= 50 # sample size
N= 1000 # number of repetition
set.seed(101)
max = 10 # maximum of x1 & x2
olsmatrix = matrix(0,N,7)
nlsmatrix = matrix(NA,N,7)
OTEtrue = matrix(0,N,1)
etatrue = matrix(0,N,1)
for (i in 1:N){
  x1 = runif(M,mu1*1.0001,max)
  x2 = runif(M,mu2*1.0001,max)
  k =x2/x1
  v = rnorm(M, mean = 0, sd = sigmav)
  u = abs(rnorm(M, mean = 0, sd = sigmau))
```

```
q1=a*(x1-mu1)^b1*(x2-mu2)^b2
y=log(q1)-u
q=exp(y)
# OLS
x = cbind(log(x1),log(x2))
ols=lm(y~x)
aols=coef(ols)[1]+max(residuals(ols))
OTE=exp(residuals(ols)-max(residuals(ols)))
OTE1=OTE[1]
OTEt = exp(-u)    #True OTE
OTEt1=OTEt[1]     #True OTE for observation1
eta1 = coef(ols)[2]
eta2 = coef(ols)[3]
eta = eta1 + eta2 #eta estimates
olsmatrix[i,] = cbind(exp(aols),coef(ols)[2],
                      coef(ols)[3],0,0,OTE1,eta)

# True eta and OTE1
OTEtrue[i,] =OTEt1
etatrue[i,] = b1 * mean(x1)/(mean(x1)-mu1)+b2 * mean(x2)/(mean(x2)-mu2)

# NLS
TF = try(nls(y~astar+b1*log(x1-mu1)+b2*log(x2-mu2),
             start=c(astar=0.1,b1=0.3,b2=0.4,mu1=0,mu2=0),
             algorithm="port", lower=c(-Inf,0,0,0,0),
             upper=c(Inf,Inf,Inf,0.999999*min(x1),0.999999*min(x2)),
             control = list(maxiter = 5000)), silent = T)
if (class(TF)=="try-error"){

} else {

  nls = TF
  lna=coef(nls)[1]+max(residuals(nls))
  OTE=exp(residuals(nls)-max(residuals(nls)))
  OTE1=OTE[1]
  eta1 = coef(nls)[2] * mean(x1)/(mean(x1)-coef(nls)[4])
  eta2 = coef(nls)[3] * mean(x2)/(mean(x2)-coef(nls)[5])
  eta = eta1 + eta2
  nlsmatrix[i,] = cbind(exp(lna),coef(nls)[2],coef(nls)[3],
                        coef(nls)[4],coef(nls)[5],OTE1,eta)
```

```
  }}
# ols results combine
olsmatrix = subset(olsmatrix, olsmatrix[,1] !=0)
olsresult = colMeans(olsmatrix)

# True
OTEtrue = subset(OTEtrue,OTEtrue!=0)
OTE1true = colMeans(OTEtrue)
etatrue = subset(etatrue,etatrue!=0)
etameantrue = colMeans(etatrue)
TrueValue=cbind(a,b1,b2,mu1,mu2,OTE1true,etameantrue)

# nls results combine
nlsmatrix1 = subset(nlsmatrix, nlsmatrix[,1] !=0)
nlsresult = colMeans(nlsmatrix1)

# compare
compare = rbind(TrueValue,olsresult,nlsresult)
compare
TVmatrix = cbind(a,b1,b2,mu1,mu2,OTEtrue,etatrue)

# Bias & MSE
biasols = colMeans(olsmatrix-TVmatrix)
MSEols = colSums((olsmatrix - TVmatrix)^2)/length(olsmatrix)

biasnls = colMeans(na.omit(nlsmatrix-TVmatrix))
MSEnls = colSums((na.omit(nlsmatrix - TVmatrix))^2)/length(nlsmatrix)

# Output
TrueValue=TrueValue[1,]
obs2 = cbind(TrueValue,olsresult,nlsresult,biasols,MSEols,biasnls,MSEnls)
olsresult-biasols-TrueValue
nlsresult-biasnls-TrueValue

### OLS & NLS; M=500 ; mu1 = 1.5 mu2 = 3.5#####
rm(list= ls()[!(ls() %in% c('obs1','obs2'))])
a=1
lna=log(a)
b1=0.3
b2=0.4
mu1=1.5
mu2=3.5
```

```
sigmav= 0.1
sigmau =0.2
lamda= sigmau/sigmav
sigma=sigmav^2+sigmau^2
M= 500 # sample size
N= 1000 # number of repetition
set.seed(101)
max = 10 # maximum of x1 & x2
olsmatrix = matrix(0,N,7)
nlsmatrix = matrix(NA,N,7)
OTEtrue = matrix(0,N,1)
etatrue = matrix(0,N,1)
for (i in 1:N){
  x1 = runif(M,mu1*1.0001,max)
  x2 = runif(M,mu2*1.0001,max)
  k =x2/x1
  v = rnorm(M, mean = 0, sd = sigmav)
  u = abs(rnorm(M, mean = 0, sd = sigmau))
  q1=a*(x1-mu1)^b1*(x2-mu2)^b2
  y=log(q1)-u
  q=exp(y)
  # OLS
  x = cbind(log(x1),log(x2))
  ols=lm(y~x)
  aols=coef(ols)[1]+max(residuals(ols))
  OTE=exp(residuals(ols)-max(residuals(ols)))
  OTE1=OTE[1]
  OTEt = exp(-u)    #True OTE
  OTEt1=OTEt[1]     #True OTE for observation1
  eta1 = coef(ols)[2]
  eta2 = coef(ols)[3]
  eta = eta1 + eta2 #eta estimates
  olsmatrix[i,] = cbind(exp(aols),coef(ols)[2],coef(ols)[3],0,0,OTE1,eta)

  # True eta and OTE1
  OTEtrue[i,] =OTEt1
  etatrue[i,] = b1 * mean(x1)/(mean(x1)-mu1)+b2 *
    mean(x2)/(mean(x2)-mu2)

  # NLS
  TF = try(nls(y~astar+b1*log(x1-mu1)+b2*log(x2-mu2),
               start=c(astar=0.1,b1=0.3,b2=0.4,mu1=0,mu2=0),
```

```
                algorithm="port", lower=c(-Inf,0,0,0,0),
                upper=c(Inf,Inf,Inf,0.999999*min(x1),0.999999*min(x2)),
                control = list(maxiter = 5000)), silent = T)
  if (class(TF)=="try-error"){

  } else {

    nls = TF
    lna=coef(nls)[1]+max(residuals(nls))
    OTE=exp(residuals(nls)-max(residuals(nls)))
    OTE1=OTE[1]
    eta1 = coef(nls)[2] * mean(x1)/(mean(x1)-coef(nls)[4])
    eta2 = coef(nls)[3] * mean(x2)/(mean(x2)-coef(nls)[5])
    eta = eta1 + eta2
    nlsmatrix[i,] = cbind(exp(lna),coef(nls)[2],
                          coef(nls)[3],coef(nls)[4],coef(nls)[5],OTE1,eta)


  }}
# ols results combine
olsmatrix = subset(olsmatrix, olsmatrix[,1] !=0)
olsresult = colMeans(olsmatrix)

# True
OTEtrue = subset(OTEtrue,OTEtrue!=0)
OTE1true = colMeans(OTEtrue)
etatrue = subset(etatrue,etatrue!=0)
etameantrue = colMeans(etatrue)
TrueValue=cbind(a,b1,b2,mu1,mu2,OTE1true,etameantrue)

# nls results combine
nlsmatrix1 = subset(nlsmatrix, nlsmatrix[,1] !=0)
nlsresult = colMeans(nlsmatrix1)

# compare
compare = rbind(TrueValue,olsresult,nlsresult)
compare
TVmatrix = cbind(a,b1,b2,mu1,mu2,OTEtrue,etatrue)

# Bias & MSE
biasols = colMeans(olsmatrix-TVmatrix)
MSEols = colSums((olsmatrix - TVmatrix)^2)/length(olsmatrix)
```

```
biasnls = colMeans(na.omit(nlsmatrix-TVmatrix))
MSEnls = colSums((na.omit(nlsmatrix - TVmatrix))^2)/length(nlsmatrix)

# Output
roundup = function(x,n){sign(x)*trunc(abs(x)*10^n + 0.5)/10^n}
TrueValue=TrueValue[1,]
obss = cbind(TrueValue, olsresult, nlsresult, biasols, MSEols, biasnls, MSEnls)
obs2 = rbind(obs2, obss)
roundup(obs1,3)
roundup(obs2,3)

olsresult-biasols-TrueValue
nlsresult-biasnls-TrueValue



write.csv(rbind(obs1, obs2), file="DFA1.csv")
```

B.3 Two Inputs With Measurement Error

```
### MLES & MLET; M=50 ; mu1 = mu2 = 0####
rm(list=ls())
a=1
lna=log(a)
b1=0.3
b2=0.9
mu1=0
mu2=0
sigmav= 0.1
sigmau =0.2
lamda= sigmau/sigmav
sigma=sigmav^2+sigmau^2
M= 50 # sample size
N= 1000 # number of repetition
set.seed(101)
max = 10 # maximum of x1 & x2
mlesmatrix = matrix(0,N,9)
mletmatrix = matrix(NA,N,9)
OTEtrue = matrix(0,N,1)
etatrue = matrix(0,N,1)
```

```
for (i in 1:N){
  x1 = runif(M,mu1*1.0001,max)
  x2 = runif(M,mu2*1.0001,max)
  k =x2/x1
  v = rnorm(M, mean = 0, sd = sigmav)
  u = abs(rnorm(M, mean = 0, sd = sigmau))
  #library("fdrtool")
  # u= rhalfnorm(M, theta=5*sqrt(pi/2))
  # theta = sqrt(pi/2)/sigma2
  q1=a*(x1-mu1)^b1*(x2-mu2)^b2
  y=log(q1)-u+v
  q=exp(y)

  # MLE (standard)#####
  start = c(lna-0.4,b1+0.04,b2+0.08,sigma+0.4,lamda+0.3)
  x = cbind(x1,x2)
  func = function(parm, y, x) {
    a1 = parm[1]
    b1 = parm[2]
    b2 = parm[3]
    sig = parm[4]
    lamda1 = parm[5]
    N = nrow(x)
    f = a1 + b1*log(x[,1]) +b2*log(x[,2])
    logl = N*log(2/sqrt(2*pi*sig^2))-1/(2*sig^2)*
      sum((y-f)^2)+sum(log(pnorm(-lamda1*(y-f)/sig)))
    return(logl)
  }
  ml = optim(start, func, control=list(fnscale=-1, maxit = 10000),
             y = y, x = x, method="BFGS")
  # mu1=ml$par[6]
  # mu2=ml$par[7]
  # b1=ml$par[2]
  # b2=ml$par[3]
  # a=exp(ml$par[1])

  # construct the technique efficiency OTE
  lamdasq=(ml$par[5])^2
  sigmasq=(ml$par[4])^2
  sigvsq=sigmasq/(lamdasq+1)
  sigusq=sigmasq-sigvsq
  error=y-ml$par[1]-log(x1)*ml$par[2]-log(x2)*ml$par[3]
```

```
mustar=-error*sigusq/sigmasq
sigmastarsq=sigvsq*sigusq/sigmasq
sigmastar=sqrt(sigmastarsq)
# using Jondorow's predictor of u
predict_u=mustar+sigmastar*(dnorm(mustar/sigmastar)/
                                    pnorm(mustar/sigmastar))
OTE=exp(-predict_u)
OTE1=OTE[1]
OTEt = exp(-u)    #True OTE
OTEt1=OTEt[1]
eta1 = ml$par[2] * mean(x1)/(mean(x1)-mu1)
eta2 = ml$par[3] * mean(x2)/(mean(x2)-mu2)
eta = eta1 + eta2
# True eta and OTE1
OTEtrue[i,] =OTEt1
etatrue[i,] = b1 * mean(x1)/(mean(x1)-mu1)+b2 *
  mean(x2)/(mean(x2)-mu2)

# MLES result
mlesmatrix[i,] = cbind(exp(ml$par[1]),ml$par[2],
                           ml$par[3],0,0,sigvsq,sigusq,OTE1,eta)


# True eta and OTE1
OTEtrue[i,] =OTEt1
etatrue[i,] = b1 * mean(x1)/(mean(x1)-mu1)+b2 *
  mean(x2)/(mean(x2)-mu2)

# MLE (affine)#####
start = c(lna+0.4,b1+0.04,b2+0.08,0,lamda+0.3,0,0)
x = cbind(x1,x2)
func = function(parm, y, x) {
  a1 = parm[1]
  b1 = parm[2]
  b2 = parm[3]
  sig = parm[4]
  lamda1 = parm[5]
  mu1 = parm[6]
  mu2 = parm[7]
  N = nrow(x)
  f = a1 + b1*log(x[,1]-mu1) +b2*log(x[,2]-mu2)
  logl = N*log(2/sqrt(2*pi*sig^2))-1/(2*sig^2)*
```

```
    sum((y-f)^2)+sum(log(pnorm(-lamda1*(y-f)/sig)))
  return(logl)
}
# only for mu = 0
TF = try(optim(start,func,control=list(fnscale=-1, maxit = 1000),
                 y = y, x = x, method="L-BFGS-B",
                 lower=c(-Inf,0,0,0.01,0,0,0),
                 upper = c(Inf,Inf,Inf,Inf,Inf,0.99999*min(x1),
                           0.99999*min(x2))),silent = T)
if (class(TF)=="try-error"){

} else {
  ml = TF
  # mu1=ml$par[6]
  # mu2=ml$par[7]
  # b1=ml$par[2]
  # b2=ml$par[3]
  # a=exp(ml$par[1])

  # construct the technique efficiency OTE
  lamdasq=(ml$par[5])^2
  sigmasq=(ml$par[4])^2
  sigvsq=sigmasq/(lamdasq+1)
  sigusq=sigmasq-sigvsq
  error=y-ml$par[1]-log(x1-ml$par[6])*
    ml$par[2]-log(x2-ml$par[7])*ml$par[3]
  mustar=-error*sigusq/sigmasq
  sigmastarsq=sigvsq*sigusq/sigmasq
  sigmastar=sqrt(sigmastarsq)
  # using Jondorow's predictor of u
  predict_u=mustar+sigmastar*
    (dnorm(mustar/sigmastar)/pnorm(mustar/sigmastar))
  OTE=exp(-predict_u)
  OTE1=OTE[1]
  OTEt = exp(-u)    #True OTE
  OTEt1=OTEt[1]
  eta1 = ml$par[2] * mean(x1)/(mean(x1)-ml$par[6])
  eta2 = ml$par[3] * mean(x2)/(mean(x2)-ml$par[7])
  eta = eta1 + eta2
  # MLET result
  mletmatrix[i,] = cbind(exp(ml$par[1]),
                         ml$par[2],ml$par[3],ml$par[6],
```

```
                                ml$par[7],sigvsq,sigusq,OTE1,eta)

  }
}
# mles results combine
mlesmatrix = subset(mlesmatrix, mlesmatrix[,1] !=0)
mlesresult = colMeans(mlesmatrix)

# True
OTEtrue = subset(OTEtrue,OTEtrue!=0)
OTE1true = colMeans(OTEtrue)
etatrue = subset(etatrue,etatrue!=0)
etameantrue = colMeans(etatrue)
TrueValue=cbind(a,b1,b2,mu1,mu2,sigmav^2,sigmau^2,OTE1true,etameantrue)

# mlet results combine
mletmatrix1 = na.omit(mletmatrix)
mletresult = colMeans(mletmatrix1)

# compare
compare = rbind(TrueValue,mlesresult,mletresult)
compare
TVmatrix = cbind(a,b1,b2,mu1,mu2,sigmav^2,sigmau^2,OTEtrue,etatrue)

# Bias & MSE
biasmles = colMeans(mlesmatrix-TVmatrix)
MSEmles = colSums((mlesmatrix - TVmatrix)^2)/length(mlesmatrix)

biasmlet = mletresult-TrueValue
MSEmlet = colSums((na.omit(mletmatrix - TVmatrix))^2)/length(mletmatrix)

# Output
TrueValue=TrueValue[1,]
obs1 = cbind(TrueValue,mlesresult,
             mletresult,biasmles,MSEmles,unname(biasmlet)[1,],MSEmlet)
mlesresult-biasmles-TrueValue
mletresult-biasmlet-TrueValue


### MLES & MLET; M=500 ; mu1 = mu2 = 0####
rm(list=setdiff(ls(), "obs1"))
a=1
```

```
lna=log(a)
b1=0.3
b2=0.9
mu1=0
mu2=0
sigmav= 0.1
sigmau =0.2
lamda= sigmau/sigmav
sigma=sigmav^2+sigmau^2
M= 500 # sample size
N= 1000 # number of repetition
set.seed(101)
max = 10 # maximum of x1 & x2
mlesmatrix = matrix(0,N,9)
mletmatrix = matrix(NA,N,9)
OTEtrue = matrix(0,N,1)
etatrue = matrix(0,N,1)

for (i in 1:N){
  x1 = runif(M,mu1*1.0001,max)
  x2 = runif(M,mu2*1.0001,max)
  k =x2/x1
  v = rnorm(M, mean = 0, sd = sigmav)
  u = abs(rnorm(M, mean = 0, sd = sigmau))
  #library("fdrtool")
  # u= rhalfnorm(M, theta=5*sqrt(pi/2))
  # theta = sqrt(pi/2)/sigma2
  q1=a*(x1-mu1)^b1*(x2-mu2)^b2
  y=log(q1)-u+v
  q=exp(y)

  # MLE (standard)#####
  start = c(lna-0.4,b1+0.04,b2+0.08,sigma+0.4,lamda+0.3)
  x = cbind(x1,x2)
  func = function(parm, y, x) {
    a1 = parm[1]
    b1 = parm[2]
    b2 = parm[3]
    sig = parm[4]
    lamda1 = parm[5]
    N = nrow(x)
    f = a1 + b1*log(x[,1]) +b2*log(x[,2])
```

```
  logl = N*log(2/sqrt(2*pi*sig^2))-1/(2*sig^2)*
    sum((y-f)^2)+sum(log(pnorm(-lamda1*(y-f)/sig)))
  return(logl)
}
ml =optim(start,func,control=list(fnscale=-1, maxit = 10000),
          y = y, x = x, method="BFGS")
# mu1=ml$par[6]
# mu2=ml$par[7]
# b1=ml$par[2]
# b2=ml$par[3]
# a=exp(ml$par[1])

# construct the technique efficiency OTE
lamdasq=(ml$par[5])^2
sigmasq=(ml$par[4])^2
sigvsq=sigmasq/(lamdasq+1)
sigusq=sigmasq-sigvsq
error=y-ml$par[1]-log(x1)*ml$par[2]-log(x2)*ml$par[3]
mustar=-error*sigusq/sigmasq
sigmastarsq=sigvsq*sigusq/sigmasq
sigmastar=sqrt(sigmastarsq)
# using Jondorow's predictor of u
predict_u=mustar+sigmastar*(dnorm(mustar/sigmastar)/
                            pnorm(mustar/sigmastar))
OTE=exp(-predict_u)
OTE1=OTE[1]
OTEt = exp(-u)    #True OTE
OTEt1=OTEt[1]
eta1 = ml$par[2] * mean(x1)/(mean(x1)-mu1)
eta2 = ml$par[3] * mean(x2)/(mean(x2)-mu2)
eta = eta1 + eta2
# True eta and OTE1
OTEtrue[i,] =OTEt1
etatrue[i,] = b1 * mean(x1)/(mean(x1)-mu1)+b2 * mean(x2)/
  (mean(x2)-mu2)

# MLES result
mlesmatrix[i,] = cbind(exp(ml$par[1]),
                       ml$par[2],ml$par[3],0,0,sigvsq,sigusq,OTE1,eta)


# True eta and OTE1
```

```
OTEtrue[i,] =OTEt1
etatrue[i,] = b1*mean(x1)/(mean(x1)-mu1)+b2 * mean(x2)/(mean(x2)-mu2)

# MLE (affine)#####
start = c(lna+0.4,b1+0.04,b2+0.08,0,lamda+0.3,0,0)
x = cbind(x1,x2)
func = function(parm, y, x) {
  a1 = parm[1]
  b1 = parm[2]
  b2 = parm[3]
  sig = parm[4]
  lamda1 = parm[5]
  mu1 = parm[6]
  mu2 = parm[7]
  N = nrow(x)
  f = a1 + b1*log(x[,1]-mu1) +b2*log(x[,2]-mu2)
  logl = N*log(2/sqrt(2*pi*sig^2))-1/(2*sig^2)*
    sum((y-f)^2)+sum(log(pnorm(-lamda1*(y-f)/sig)))
  return(logl)
}
# only for mu = 0
TF = try( optim(start, func, control=list(fnscale=-1, maxit = 1000),
                y = y, x = x, method="L-BFGS-B",
                lower=c(-Inf,0,0,0.01,0,0,0),
                upper = c(Inf,Inf,Inf,Inf,Inf,0.999999*min(x1),
                          0.999999*min(x2))),silent = T)
if (class(TF)=="try-error"){

} else {
  ml = TF
  # mu1=ml$par[6]
  # mu2=ml$par[7]
  # b1=ml$par[2]
  # b2=ml$par[3]
  # a=exp(ml$par[1])

  # construct the technique efficiency OTE
  lamdasq=(ml$par[5])^2
  sigmasq=(ml$par[4])^2
  sigvsq=sigmasq/(lamdasq+1)
  sigusq=sigmasq-sigvsq
  error=y-ml$par[1]-log(x1-ml$par[6])*ml$par[2]-
```

```
      log(x2-ml$par[7])*ml$par[3]
    mustar=-error*sigusq/sigmasq
    sigmastarsq=sigvsq*sigusq/sigmasq
    sigmastar=sqrt(sigmastarsq)
    # using Jondorow's predictor of u
    predict_u=mustar+sigmastar*(dnorm(mustar/sigmastar)/
                                pnorm(mustar/sigmastar))
    OTE=exp(-predict_u)
    OTE1=OTE[1]
    OTEt = exp(-u)    #True OTE
    OTEt1=OTEt[1]
    eta1 = ml$par[2] * mean(x1)/(mean(x1)-ml$par[6])
    eta2 = ml$par[3] * mean(x2)/(mean(x2)-ml$par[7])
    eta = eta1 + eta2
    # MLET result
    mletmatrix[i,] = cbind(exp(ml$par[1]),ml$par[2],
                           ml$par[3],ml$par[6],ml$par[7],
                           sigvsq,sigusq,OTE1,eta)

  }
}
# mles results combine
mlesmatrix = subset(mlesmatrix, mlesmatrix[,1] !=0)
mlesresult = colMeans(mlesmatrix)

# True
OTEtrue = subset(OTEtrue,OTEtrue!=0)
OTE1true = colMeans(OTEtrue)
etatrue = subset(etatrue,etatrue!=0)
etameantrue = colMeans(etatrue)
TrueValue=cbind(a,b1,b2,mu1,mu2,sigmav^2,sigmau^2,OTE1true,etameantrue)

# mlet results combine
mletmatrix1 = na.omit(mletmatrix)
mletresult = colMeans(mletmatrix1)

# compare
compare = rbind(TrueValue,mlesresult,mletresult)
compare
TVmatrix = cbind(a,b1,b2,mu1,mu2,sigmav^2,sigmau^2,OTEtrue,etatrue)

# Bias & MSE
```

```
biasmles = colMeans(mlesmatrix-TVmatrix)
MSEmles = colSums((mlesmatrix - TVmatrix)^2)/length(mlesmatrix)

biasmlet = mletresult-TrueValue
MSEmlet = colSums((na.omit(mletmatrix - TVmatrix))^2)/length(mletmatrix)

# Output
TrueValue=TrueValue[1,]
obss = cbind(TrueValue,mlesresult,
             mletresult,biasmles,MSEmles,unname(biasmlet)[1,],MSEmlet)
obs1 = rbind(obs1,obss)
mlesresult-biasmles-TrueValue
mletresult-biasmlet-TrueValue



### MLES & MLET; M=50 ; mu1 = 1.5 mu2 = 3.5####
rm(list=setdiff(ls(), "obs1"))
a=1
lna=log(a)
b1=0.3
b2=0.9
mu1=1.5
mu2=3.5
sigmav= 0.1
sigmau =0.2
lamda= sigmau/sigmav
sigma=sigmav^2+sigmau^2
M= 50 # sample size
N= 1000 # number of repetition
set.seed(101)
max = 10 # maximum of x1 & x2
mlesmatrix = matrix(0,N,9)
mletmatrix = matrix(NA,N,9)
OTEtrue = matrix(0,N,1)
etatrue = matrix(0,N,1)

for (i in 1:N){
  x1 = runif(M,mu1*1.0001,max)
  x2 = runif(M,mu2*1.0001,max)
  k =x2/x1
  v = rnorm(M, mean = 0, sd = sigmav)
  u = abs(rnorm(M, mean = 0, sd = sigmau))
```

```
#library("fdrtool")
# u= rhalfnorm(M, theta=5*sqrt(pi/2))
# theta = sqrt(pi/2)/sigma2
q1=a*(x1-mu1)^b1*(x2-mu2)^b2
y=log(q1)-u+v
q=exp(y)

# MLE (standard)#####
start = c(lna-0.4,b1+0.04,b2+0.08,sigma+0.4,lamda+0.3)
x = cbind(x1,x2)
func = function(parm, y, x) {
  a1 = parm[1]
  b1 = parm[2]
  b2 = parm[3]
  sig = parm[4]
  lamda1 = parm[5]
  N = nrow(x)
  f = a1 + b1*log(x[,1]) +b2*log(x[,2])
  logl = N*log(2/sqrt(2*pi*sig^2))-
    1/(2*sig^2)*sum((y-f)^2)+sum(log(pnorm(-lamda1*(y-f)/sig)))
  return(logl)
}
ml = optim(start, func, control=list(fnscale=-1, maxit = 10000),
           y = y, x = x, method="BFGS")
# mu1=ml$par[6]
# mu2=ml$par[7]
# b1=ml$par[2]
# b2=ml$par[3]
# a=exp(ml$par[1])

# construct the technique efficiency OTE
lamdasq=(ml$par[5])^2
sigmasq=(ml$par[4])^2
sigvsq=sigmasq/(lamdasq+1)
sigusq=sigmasq-sigvsq
error=y-ml$par[1]-log(x1)*ml$par[2]-log(x2)*ml$par[3]
mustar=-error*sigusq/sigmasq
sigmastarsq=sigvsq*sigusq/sigmasq
sigmastar=sqrt(sigmastarsq)
# using Jondorow's predictor of u
predict_u=mustar+sigmastar*(dnorm(mustar/sigmastar)/
                              pnorm(mustar/sigmastar))
```

```
OTE=exp(-predict_u)
OTE1=OTE[1]
OTEt = exp(-u)    #True OTE
OTEt1=OTEt[1]
eta1 = ml$par[2]
eta2 = ml$par[3]
eta = eta1 + eta2
# True eta and OTE1
OTEtrue[i,] =OTEt1
etatrue[i,] = b1 * mean(x1)/(mean(x1)-mu1)+b2 * mean(x2)/
  (mean(x2)-mu2)

# MLES result
mlesmatrix[i,] = cbind(exp(ml$par[1]),ml$par[2],ml$par[3],0,0,
                       sigvsq,sigusq,OTE1,eta)


# True eta and OTE1
OTEtrue[i,] =OTEt1
etatrue[i,] = b1 * mean(x1)/(mean(x1)-mu1)+b2 * mean(x2)/
  (mean(x2)-mu2)

# MLE (affine)#####
start = c(lna+0.4,b1+0.04,b2+0.08,0,lamda+0.3,0,0)
x = cbind(x1,x2)
func = function(parm, y, x) {
  a1 = parm[1]
  b1 = parm[2]
  b2 = parm[3]
  sig = parm[4]
  lamda1 = parm[5]
  mu1 = parm[6]
  mu2 = parm[7]
  N = nrow(x)
  f = a1 + b1*log(x[,1]-mu1) +b2*log(x[,2]-mu2)
  logl = N*log(2/sqrt(2*pi*sig^2))-1/(2*sig^2)*
    sum((y-f)^2)+sum(log(pnorm(-lamda1*(y-f)/sig)))
  return(logl)
}
# only for mu = 0
TF = try( optim(start, func, control=list(fnscale=-1, maxit = 1000),
                y = y, x = x, method="L-BFGS-B",
```

```
                    lower=c(-Inf,0,0,0.01,0,0,0),
                    upper = c(Inf,Inf,Inf,Inf,Inf,
                              0.99*min(x1),0.99*min(x2))),silent = T)
  if (class(TF)=="try-error"){

  } else {
    ml = TF
    # mu1=ml$par[6]
    # mu2=ml$par[7]
    # b1=ml$par[2]
    # b2=ml$par[3]
    # a=exp(ml$par[1])

    # construct the technique efficiency OTE
    lamdasq=(ml$par[5])^2
    sigmasq=(ml$par[4])^2
    sigvsq=sigmasq/(lamdasq+1)
    sigusq=sigmasq-sigvsq
    error=y-ml$par[1]-log(x1-ml$par[6])*ml$par[2]-
      log(x2-ml$par[7])*ml$par[3]
    mustar=-error*sigusq/sigmasq
    sigmastarsq=sigvsq*sigusq/sigmasq
    sigmastar=sqrt(sigmastarsq)
    # using Jondorow's predictor of u
    predict_u=mustar+sigmastar*(dnorm(mustar/sigmastar)/
                                  pnorm(mustar/sigmastar))
    OTE=exp(-predict_u)
    OTE1=OTE[1]
    OTEt = exp(-u)    #True OTE
    OTEt1=OTEt[1]
    eta1 = ml$par[2] * mean(x1)/(mean(x1)-ml$par[6])
    eta2 = ml$par[3] * mean(x2)/(mean(x2)-ml$par[7])
    eta = eta1 + eta2
    # MLET result
    mletmatrix[i,] = cbind(exp(ml$par[1]),ml$par[2],ml$par[3],
                           ml$par[6],ml$par[7],sigvsq,sigusq,OTE1,eta)

  }
}
# mles results combine
mlesmatrix = subset(mlesmatrix, mlesmatrix[,1] !=0)
mlesresult = colMeans(na.omit(mlesmatrix))
```

```
# True
OTEtrue = subset(OTEtrue,OTEtrue!=0)
OTE1true = colMeans(OTEtrue)
etatrue = subset(etatrue,etatrue!=0)
etameantrue = colMeans(etatrue)
TrueValue=cbind(a,b1,b2,mu1,mu2,sigmav^2,sigmau^2,OTE1true,etameantrue)

# mlet results combine
mletmatrix1 = na.omit(mletmatrix)
mletresult = colMeans(mletmatrix1)

# compare
compare = rbind(TrueValue,mlesresult,mletresult)
compare
TVmatrix = cbind(a,b1,b2,mu1,mu2,sigmav^2,sigmau^2,OTEtrue,etatrue)

# Bias & MSE
biasmles = colMeans(na.omit(mlesmatrix-TVmatrix))
MSEmles = colSums((na.omit(mlesmatrix-TVmatrix))^2)/length(mlesmatrix)

biasmlet = mletresult-TrueValue
MSEmlet = colSums((na.omit(mletmatrix - TVmatrix))^2)/length(mletmatrix)

# Output
TrueValue=TrueValue[1,]
obs2 = cbind(TrueValue,mlesresult,mletresult,
             biasmles,MSEmles,unname(biasmlet)[1,],MSEmlet)
mlesresult-biasmles-TrueValue
mletresult-biasmlet-TrueValue


### MLES & MLET; M=500 ; mu1 = 1.5 mu2 = 3.5####
rm(list= ls()[!(ls() %in% c('obs1','obs2'))])
a=1
lna=log(a)
b1=0.3
b2=0.9
mu1=1.5
mu2=3.5
sigmav= 0.1
sigmau =0.2
```

```
lamda= sigmau/sigmav
sigma=sigmav^2+sigmau^2
M= 500 # sample size
N= 1000 # number of repetition
set.seed(101)
max = 10 # maximum of x1 & x2
mlesmatrix = matrix(0,N,9)
mletmatrix = matrix(NA,N,9)
OTEtrue = matrix(0,N,1)
etatrue = matrix(0,N,1)

for (i in 1:N){
  x1 = runif(M,1.5*1.0001,max)
  x2 = runif(M,3.5*1.0001,max)
  k =x2/x1
  v = rnorm(M, mean = 0, sd = sigmav)
  u = abs(rnorm(M, mean = 0, sd = sigmau))
  #library("fdrtool")
  # u= rhalfnorm(M, theta=5*sqrt(pi/2))
  # theta = sqrt(pi/2)/sigma2
  q1=a*(x1-mu1)^b1*(x2-mu2)^b2
  y=log(q1)-u+v
  q=exp(y)

  # MLE (standard)#####
  start = c(lna-0.4,b1+0.04,b2+0.08,sigma+0.4,lamda+0.3)
  x = cbind(x1,x2)
  func = function(parm, y, x) {
    a1 = parm[1]
    b1 = parm[2]
    b2 = parm[3]
    sig = parm[4]
    lamda1 = parm[5]
    N = nrow(x)
    f = a1 + b1*log(x[,1]) +b2*log(x[,2])
    logl = N*log(2/sqrt(2*pi*sig^2))-1/(2*sig^2)*
      sum((y-f)^2)+sum(log(pnorm(-lamda1*(y-f)/sig)))
    return(logl)
  }
  ml = optim(start, func, control=list(fnscale=-1,
                                       maxit = 10000),
             y = y, x = x, method="BFGS")
```

```
# mu1=ml$par[6]
# mu2=ml$par[7]
# b1=ml$par[2]
# b2=ml$par[3]
# a=exp(ml$par[1])

# construct the technique efficiency OTE
lamdasq=(ml$par[5])^2
sigmasq=(ml$par[4])^2
sigvsq=sigmasq/(lamdasq+1)
sigusq=sigmasq-sigvsq
error=y-ml$par[1]-log(x1)*ml$par[2]-log(x2)*ml$par[3]
mustar=-error*sigusq/sigmasq
sigmastarsq=sigvsq*sigusq/sigmasq
sigmastar=sqrt(sigmastarsq)
# using Jondorow's predictor of u
predict_u=mustar+sigmastar*(dnorm(mustar/sigmastar)/
                                    pnorm(mustar/sigmastar))
OTE=exp(-predict_u)
OTE1=OTE[1]
OTEt = exp(-u)    #True OTE
OTEt1=OTEt[1]
eta1 = ml$par[2]
eta2 = ml$par[3]
eta = eta1 + eta2
# True eta and OTE1
OTEtrue[i,] =OTEt1
etatrue[i,] = b1 * mean(x1)/(mean(x1)-mu1)+b2 * mean(x2)/
  (mean(x2)-mu2)

# MLES result
mlesmatrix[i,] = cbind(exp(ml$par[1]),ml$par[2],ml$par[3],0,0,
                           sigvsq,sigusq,OTE1,eta)


# True eta and OTE1
OTEtrue[i,] =OTEt1
etatrue[i,] = b1 * mean(x1)/(mean(x1)-mu1)+b2 * mean(x2)/
  (mean(x2)-mu2)

# MLE (affine)#####
start = c(lna+0.4,b1+0.04,b2+0.08,0.01,lamda+0.3,0,0)
```

```
x = cbind(x1,x2)
func = function(parm, y, x) {
  a1 = parm[1]
  b1 = parm[2]
  b2 = parm[3]
  sig = parm[4]
  lamda1 = parm[5]
  mu1 = parm[6]
  mu2 = parm[7]
  N = nrow(x)
  f = a1 + b1*log(x[,1]-mu1) +b2*log(x[,2]-mu2)
  logl = N*log(2/sqrt(2*pi*sig^2))-1/(2*sig^2)*sum((y-f)^2)+
    sum(log(pnorm(-lamda1*(y-f)/sig)))
  return(logl)
}
# only for mu = 0
TF = try( optim(start, func, control=list(fnscale=-1, maxit = 1000),
                y = y, x = x, method="L-BFGS-B",
                lower=c(-Inf,0,0,0.01,0,0,0),
                upper = c(Inf,Inf,Inf,Inf,Inf,0.99*min(x1),
                          0.99*min(x2))),silent = T)
if (class(TF)=="try-error"){

} else {
  ml = TF
  # mu1=ml$par[6]
  # mu2=ml$par[7]
  # b1=ml$par[2]
  # b2=ml$par[3]
  # a=exp(ml$par[1])

  # construct the technique efficiency OTE
  lamdasq=(ml$par[5])^2
  sigmasq=(ml$par[4])^2
  sigvsq=sigmasq/(lamdasq+1)
  sigusq=sigmasq-sigvsq
  error=y-ml$par[1]-log(x1-ml$par[6])*ml$par[2]-
    log(x2-ml$par[7])*ml$par[3]
  mustar=-error*sigusq/sigmasq
  sigmastarsq=sigvsq*sigusq/sigmasq
  sigmastar=sqrt(sigmastarsq)
  # using Jondorow's predictor of u
```

```
      predict_u=mustar+sigmastar*(dnorm(mustar/sigmastar)/
                                    pnorm(mustar/sigmastar))
      OTE=exp(-predict_u)
      OTE1=OTE[1]
      OTEt = exp(-u)     #True OTE
      OTEt1=OTEt[1]
      eta1 = ml$par[2] * mean(x1)/(mean(x1)-ml$par[6])
      eta2 = ml$par[3] * mean(x2)/(mean(x2)-ml$par[7])
      eta = eta1 + eta2
      # MLET result
      mletmatrix[i,] = cbind(exp(ml$par[1]),ml$par[2],ml$par[3],
                             ml$par[6],ml$par[7],sigvsq,sigusq,OTE1,eta)

  }
}
# mles results combine
mlesmatrix = subset(mlesmatrix, mlesmatrix[,1] !=0)
mlesresult = colMeans(mlesmatrix)

# True
OTEtrue = subset(OTEtrue,OTEtrue!=0)
OTE1true = colMeans(OTEtrue)
etatrue = subset(etatrue,etatrue!=0)
etameantrue = colMeans(etatrue)
TrueValue=cbind(a,b1,b2,mu1,mu2,sigmav^2,sigmau^2,OTE1true,etameantrue)

# mlet results combine
mletmatrix1 = na.omit(mletmatrix)
mletresult = colMeans(mletmatrix1)

# compare
compare = rbind(TrueValue,mlesresult,mletresult)
compare
TVmatrix = cbind(a,b1,b2,mu1,mu2,sigmav^2,sigmau^2,OTEtrue,etatrue)

# Bias & MSE
biasmles = colMeans(mlesmatrix-TVmatrix)
MSEmles = colSums((mlesmatrix - TVmatrix)^2)/length(mlesmatrix)

biasmlet = mletresult-TrueValue
MSEmlet = colSums((na.omit(mletmatrix - TVmatrix))^2)/length(mletmatrix)
```

```
# Output
TrueValue=TrueValue[1,]
obss = cbind(TrueValue, mlesresult, mletresult, biasmles,
             MSEmles, unname(biasmlet)[1,], MSEmlet)
obs2 = rbind(obs2, obss)
mlesresult-biasmles-TrueValue
mletresult-biasmlet-TrueValue


roundup = function(x,n){sign(x)*trunc(abs(x)*10^n + 0.5)/10^n}
roundup(obs1,3)
roundup(obs2,3)
#write.csv(rbind(obs1, obs2), file="SFA2.csv")
```

B.4 Empirical Illustration

```
rm(list=ls())
rice <- read.csv("rice.csv")
attach(rice)
summary(rice)
y = log(q)
q=exp(y)

## DFA - OLS & NLS##################
# obtain a,b,mu
# 4 inputs
# t= Year-1989
# OLS#######

x = cbind(log(x1),log(x2),log(x3),log(x4)) # log(x4) is not significant
ols=lm(y~x)
llfr = logLik(ols)
astar=coef(summary(ols))[1,1]
a=exp(astar+max(residuals(ols)))
b1=coef(summary(ols))[2,1]
b2=coef(summary(ols))[3,1]
b3=coef(summary(ols))[4,1]
b4=coef(summary(ols))[5,1]
olsc=cbind(a,b1,b2,b3,b4,0,0,0,0)
# NLS#######

nls=nls(y~astar+b1*log(x1-mu1)+b2*log(x2-mu2)+b3*log(x3-mu3)+
```

```
    b4*log(x4-mu4),
  start=c(astar=astar,b1=b1,b2=b2,b3=b3,b4=b4,
          mu1=0,mu2=0,mu3=0,mu4=0), algorithm="port",
  lower=c(-Inf,0,0,0,0,0,0,0,0),
  upper=c(Inf,Inf,Inf,Inf,Inf,0.9999*min(x1),
          0.9999*min(x2),0.9999*min(x3),0.9999*min(x4)),
          control=list(maxiter=500))

sum(coef(nls)[2:5]) # check all b <1
a=exp(coef(nls)[1]+max(residuals(nls)))
b1=coef(nls)[2]
b2=coef(nls)[3]
b3=coef(nls)[4]
b4=coef(nls)[5]
mu1=coef(nls)[6]
mu2=coef(nls)[7]
mu3=coef(nls)[8]
mu4=coef(nls)[9]
nlsols=rbind(olsc,cbind(a,b1,b2,b3,b4,mu1,mu2,mu3,mu4))
nlsols

## SFA - MLE#####

start=c(astar=astar,b1=b1,b2=b2,b3=b3,b4=b4,2,2,mu1=0,mu2=0,mu3=0,mu4=0)
x = cbind(x1,x2,x3,x4)
func = function(parm, y, x) {
  a1 = parm[1]
  b1 = parm[2]
  b2 = parm[3]
  b3 = parm[4]
  b4 = parm[5]
  sig = parm[6]
  lamda1 = parm[7]
  mu1 = parm[8]
  mu2 = parm[9]
  mu3 = parm[10]
  mu4 = parm[11]
  N = nrow(x)
  f = a1 + b1*log(x[,1]-mu1)+b2*log(x[,2]-mu2)+
    b3*log(x[,3]-mu3)+b4*log(x[,4]-mu4)
  logl = N*log(2/sqrt(2*pi*sig^2))-1/(2*sig^2)*
    sum((y-f)^2)+sum(log(pnorm(-lamda1*(y-f)/sig)))
```

```
  return(logl)
}
ml = optim(start, func, control=list(fnscale=-1, maxit = 10000),
y = y,
            x = x,
            method="L-BFGS-B",hessian=T,lower=c(-Inf,0,0,0,0,0,0,0),
            upper=c(Inf,Inf,Inf,Inf,Inf,Inf,Inf,
                    0.999999*min(x1),0.999999*min(x2),
                    0.999999*min(x3),0.999999*min(x4)))
ml$par              # the best set of parameters found
# allcompare
OI<-solve(-ml$hessian)
se<-sqrt(diag(OI))
a = ml$par[1]
b1 = ml$par[2]
b2 = ml$par[3]
b3 = ml$par[4]
b4 = ml$par[5]
sigma=ml$par[6]
lamda=ml$par[7]
mu1 = ml$par[8]
mu2 = ml$par[9]
mu3 = ml$par[10]
mu4 = ml$par[11]

#LR test for affine transformed model
llfu = ml$value
LR = 2*(llfu-llfr)
df = 4
alpha = 0.05
crit = qchisq(1-2*alpha,df)
# qchisq(a,df) is the value that leaves an area of a in the LH tail
cbind(LR,crit)
# LR > crit so reject HO an conclude affine transformed model is prefered

mleresult = cbind(exp(ml$par[1]),ml$par[2],ml$par[3],
                  ml$par[4],ml$par[5],ml$par[8],ml$par[9],
                  ml$par[10],ml$par[11])

eta = b1 * x1/(x1-mu1)+b2 * x2/(x2-mu2)+b3 * x3/(x3-mu2)+b4 *
  x4/(x4-mu4)
```

```
# OTE
lamdasq=(ml$par[7])^2
sigmasq=(ml$par[6])^2
sigvsq=sigmasq/(lamdasq+1)
sigusq=sigmasq-sigvsq
error=y-(a + b1*log(x[,1]-mu1)+b2*log(x[,2]-mu2)+
            b3*log(x[,3]-mu3)+b4*log(x[,4]-mu4))
mustar=-error*sigusq/sigmasq
sigmastarsq=sigvsq*sigusq/sigmasq
sigmastar=sqrt(sigmastarsq)
# using Jondorow's predictor of u
predict_u=mustar+sigmastar*(dnorm(mustar/sigmastar)/
                                    pnorm(mustar/sigmastar))
OTEsfa=exp(-predict_u)
summary(OTEsfa)

# OSE
a1=mean(w1)
a2=mean(w2)
a3=mean(w3)
a4=mean(w4)
q1=exp(a + b1*log(x[,1]-mu1)+b2*log(x[,2]-mu2)+
          b3*log(x[,3]-mu3)+b4*log(x[,4]-mu4))

  # fix input
k1 =x2/x1
k2 = x3/x1
k3 = x4/x1

x1star=(1:length(x1))
qbarstarslop=(1:length(x1))
for (i in 1:length(x1))
{
  func = function(x1) {
    q = exp(a + b1*log(x1-mu1)+b2*log(k1[i]*x1-mu2)+
              b3*log(k2[i]*x1-mu3)+b4*log(k3[i]*x1-mu4))
    xbig=a1*x1+a2*x1*k1[i]+a3*x1*k2[i]+a4*x1*k3[i]
    slope =q/xbig
    return(slope)
  }
  ml = optim(c(10), func, control=list(fnscale=-1, maxit = 10000),
```

```
                method="BFGS")
  x1star[i]=ml$par
  qbarstarslop[i]=ml$value
}

OSEsfa=(q1/(a1*x1+a2*x2+a3*x3+a4*x4))/qbarstarslop
summary(OSEsfa)

# TSME & RME
maxTFP_x = (a1*mu1+a2*mu2+a3*mu3+a4*mu4)/(1-b1-b2-b3-b4)
maxTFP_q =exp(a)*(b1/a1*maxTFP_x)^b1*(b2/a2*maxTFP_x)^b2*
  (b3/a3*maxTFP_x)^b3*(b4/a4*maxTFP_x)^b4
maxTFP = maxTFP_q/maxTFP_x
ehat =log(q)- (a+b1*log(x1-mu1)+b2*log(x2-mu2)+b3*log(x3-mu3)+
  b4*log(x4-mu4))
vhat = ehat +predict_u
qnoiseless = q/exp(vhat)
TFP = qnoiseless/(a1*x1+a2*x2+a3*x3+a4*x4)
TSME = TFP/maxTFP
RME = TSME/(OTEsfa*OSEsfa)
RME2 = qbarstarslop/maxTFP

# combine
result = cbind(mleresult,min(OTEsfa),mean(OTEsfa),max(OTEsfa),
                    min(OSEsfa),mean(OSEsfa),max(OSEsfa),eta[1])
result
se
# write.csv(result,file="real data.csv")

# construct OTE, OSE for farm 1
roundup = function(x, n) {sign(x)*trunc(abs(x)*10^n + 0.5)/10^n}
results1 = data.frame(Year, roundup(OTEsfa,3),roundup(OSEsfa,3),
                        roundup(RME,3),roundup(TSME,3))
colnames(results1) = c("Year","OTE","OSE","RME","TSME")
results1[Farm == 1,]
plot(TSME[Farm == 1] ~ Year[Farm == 1],  type = "l",
     xlab = "Year", ylab = "", lwd = 2, col = "black",
     ylim = c(0.0, 1.5))
points(OTEsfa[Farm == 1] ~ Year[Farm == 1],  type = "l",
       lwd = 2, col = "green")
points(OSEsfa[Farm == 1] ~ Year[Farm == 1],  type = "l",
       lwd = 2, col = "red")
```

```
points(RME[Farm == 1] ~ Year[Farm == 1],   type = "l",
       lwd = 2, col = "blue")
legend("top", legend=c("TSME", "OTE", "OSE", "RME"),
       col=c("black","green","red", "blue","gold2"),
       lty = c("solid", "solid", "solid", "solid", "solid"),
       bty = "n", lwd = 2, horiz=TRUE)
```