Nonlinear Deterministic Frontier Model Using Genetic Programming

Chin-Yi Chen[1,2], Jih-Jeng Huang[3], and Gwo-Hshiung Tzeng[1,3]

[1] Graduate Institute of Management of Technology,
National Chiao Tung University, Hsinchu, Taiwan
[2] Department of Business Administration, Chung Yuan Christian University,
Chung Li, Taiwan
[3] Department of Business and Entrepreneurial Administration,
Kainan University, Taoyuan, Taiwan
iris@cycu.edu.tw; jjhuang@mail.knu.edu.tw;
ghtzeng@mail.knu.edu.tw

Abstract. In economics, several parametric regression-based models have been proposed to measure the technical efficiency of decision making units (DMUs). However, the problem of misspecification restricts the use of these methods. In this paper, symbolic regression is employed to obtain the approximate optimal production function automatically using genetic programming (GP). Monte Carlo simulation is used to compare the performance of data envelopment analysis (DEA), deterministic frontier analysis (DFA) and GP-based DFA with respect to three different production functions and sample sizes. The simulated results indicated that the proposed method has better performance than that of others with respect to nonlinear production functions.

Keywords: Technical efficiency; symbolic regression; genetic programming (GP); Monte Carlo simulation; data envelopment analysis (DEA).

1 Introduction

Traditionally, data envelopment analysis (DEA) and regression-based methods, such as deterministic and stochastic models, are widely used to measure the technical efficiency of decision making units (DMUs). The mainly difference between DEA and regression-based methods is that DEA is a non-parametric approach while regression-based methods are parametric approaches. Several papers have been proposed to compare DEA with regression-based methods, with respect to efficiency, flexibility, robust, assumption and sample size [1-3].

DEA has been suggested to be abandoned for measuring technical efficiency, due to the disadvantages of sensitive to outlier and ignoring measurement error [4,5]. However the most critical problem of using regression-based methods is the problem of misspecification [6-7]. That is, it is necessary to specify a particular production function (e.g., Cobb-Douglas or translog form) before measuring the frontier of DMUs, and different production functions may yield different results. However, it is

Y. Shi et al. (Eds.): MCDM 2009, CCIS 35, pp. 753–760, 2009.

hard to specify a correct production function in advance, because of the complex relation between input and output variables. Therefore, this paper attempts to provide a flexible and robust way for finding the production function automatically so that the linear/nonlinear relation between input and output variables can be considered.

The problem of misspecification for regression-based models has recently been attracted much attention [6-7]. Although artificial neural network (ANN) was employed to find the nonlinear production function [7-9], it only performs better in large sample size [10]. This shortcoming restricts its application for measuring the technical efficiency of DMUs in realistic problems.

In this paper, symbolic regression is employed to obtain the approximate optimal production function automatically using the concept of genetic programming (GP). The main advantage of GP is that it can generate an optimal production function without specifying a particular production function, whether linear/nonlinear, to describe the relation between input and output variables of DMUs. In addition, unlike the restriction of ANN, GP can work well regardless of the sample size of DMUs [10].

In order to justify the performance of the proposed method, Monte Carlo simulation is used to compare symbolic regression with DEA and deterministic frontier analysis (DFA), with respect to three different production functions, including linear, Cobb-Douglas and nonlinear functions, and sample sizes, including 25, 50 and 100 samples. Then, spearman's rank correlation is used to compare the performance of models. The results reveal that the proposed model is very suitable for describing the nonlinear relation between input and output variables.

The rest of this paper is organized as follows. Reviews of regression-based frontier models are introduced in Section 2. The way to use symbolic regression and GP for measuring the efficiency of MDUs is described in Section 3. In section 4, Monte Carlo simulation is used to compare the performance of DEA, DFA and GP-based DFA. Discussions and conclusions are presented in the last section.

2 Regression-Based Frontier Models

Several regression-based frontier models have been proposed to evaluate the technical efficiency of DMUs. These methods can be roughly classified into DFA and stochastic frontier analysis (SFA). The only difference between DFA and SFA is the assumption of the residuals. In DFA, all deviations (ε_i) are assumed to be the random error. However, in SFA, the error term is composed by the measurement error (u_i) and random error (v_i), such that $\varepsilon_i = v_i - u_i$. Hence, if we assume that $u_i = 0$, SFA is reduced to DFA.

In this paper, the problem of misspecification is highlighted for regression-based frontier models. Since both DFA and SFA suffer the problem of misspecification, DFA is used to compare with the proposed model, due to its simplicity of operation. Next, we discuss the DFA model and the processes of deriving technical efficiency score.

A production frontier model can be represented as

$$y_i = f(\boldsymbol{x}_i, \boldsymbol{\beta}) + \boldsymbol{\varepsilon} \tag{1}$$

where y_i denotes the output of the *i*th firm, $\boldsymbol{x}_i$ is the input vector, $\boldsymbol{\beta}$ is the unknown parameter vector and $\boldsymbol{\varepsilon}$ is the random term. The production frontier model shows the maximum output which input vectors can achieve. Traditionally, the production function is assumed as Cobb-Douglas form and can be expressed as

$$y_i = \boldsymbol{A}\boldsymbol{x}_1^{\beta_1}\boldsymbol{x}_2^{\beta_2}\cdots\boldsymbol{x}_i^{\beta_i}\varepsilon_i, \quad \varepsilon_i \geq 0 \tag{2}$$

By taking logarithms of Eq. (2), we can rewrite the above equation as

$$\ln y_i = \ln \boldsymbol{A} + \ln \beta_1 \boldsymbol{x}_1 + \ln \beta_2 \boldsymbol{x}_2 + \cdots + \ln \beta_i \boldsymbol{x}_i - e_i, \text{ where } e_i = -\ln \varepsilon_i . \tag{3}$$

Next, we can use the ordinal least square (OLS) method to estimate the unknown parameter vector. Moreover, in order to ensure the error term to be nonnegative, the corrected OLS (COLS) is expressed as

$$\hat{e}_i^{COLS} = \hat{e}_i - \min\{\hat{e}_i\} \tag{4}$$

Finally, the efficiency can be obtained by

$$\varepsilon_i = \exp(-\hat{e}_i^{COLS}) \tag{5}$$

Although regression-based frontier models are often used to evaluate technical efficiency of DMUs in economics, they have the common problem of misspecification. These parametric models must first assume the form of production function in advance. Incorrect assumption will lead to improper ranking for DMUs. However, it is hard to quantify the production function in advance, especially when the relation between input and output variables is nonlinear.

In order to derive the production function more accurately and objectively, symbolic regression is used to obtain the approximate optimal linear/nonlinear production function automatically. Since the problem of symbolic regression is a NP-hard problem, genetic programming is considered to solve that problem. In addition, both concepts will be discussed in next section.

3 Symbolic Regression

Symbolic regression is proposed by Koza [11] to find an unknown regression function from a given sample set $\{(x, f(x) \mid x \in X)\}$ using GP and has been used in various applications [12-15]. The representation of GP can be viewed as a tree-based structure composed of the function set and terminal set. The function set is the collection of operators, functions or statements, such as arithmetic operators ($\{+,-,\times,\div\}$) or conditional statements (If …then…). On the other hand, the terminal set contains all inputs, constants and other zero-argument in the GP-tree. Consider expressing the equation $xy + 3/x$ for example, the GP tree can be represented as.

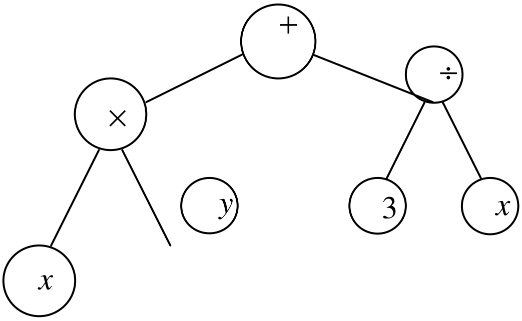

Fig. 1. The presentation of a GP-tree

Once we initialize a population of the GP tree, the following procedures are similar to genetic algorithms (GAs), including defining the fitness function, genetic operators, such as crossover, mutation and reproduction, and the termination criterion. Next, we introduce three main operators, crossover, mutation and reproduction, to show the procedures of finding the (approximate) optimal generation.

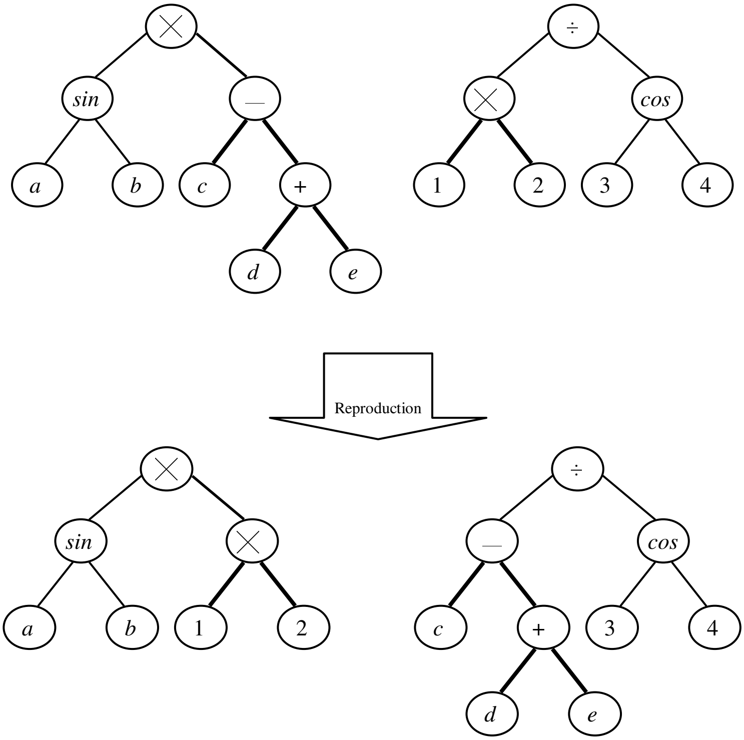

Fig. 2. The crossover operator of GP tree

For GP, the crossover operator is used to swap the sub-tree from the parents to reproduce the children using mating selection policy rather than exchange bit string in GAs. An example of crossover in GP is shown in Figure 2. Similar to GAs, GP uses the mutation operator in order to avoid falling into local optimal solution. The mutation operator is used to randomly choose a node in a sub-tree and replace it with a new created sub-tree randomly. Finally, a new generation can be reproduced from two parents using the reproduction operator to represent a better solution.

It should be highlighted that the function set and terminal set should be rich enough to represent the relation between independent and response variables. Moreover, in order to satisfy the principle of parsimony, the depth of the GP-tree should also be limited. In this paper, the depth of the GP-tree is limited to 10 levels. Next, we incorporate the concept of symbolic regression to find the approximate true production function and Monte Carlo simulation is used for testing the results in the next section.

4 Monte Carlo Simulation

In this section, Monte Carlo simulation is used for testing the efficiency of the GP-based frontier model. In order to test the ability of GP in fitting production function, three different models (i.e., linear, Cobb-Douglas and nonlinear functions) are specified to generate different data sets, including 25, 50 and 100 samples, as shown in Table 1. Each input variable (x_1 and x_2) is randomly generated by a uniform distribution within the interval 5 to 15 (i.e., $x_i \sim U(5,15)$, $i = 1,2$), respectively.

Table 1. The assumption of production functions in Monte Carlo simulation

Production Function Type	Model
Linear	$y = 3 + 2.5x_1 + 3.6x_2$
Cobb-Douglas	$y = 4x_1^{0.3}x_2^{0.7}$
Nonlinear	$y = 0.05x_1^3 + 0.01x_2^3$

In order to fit the production function curve automatically, the problem of symbolic regression is considered. Before using GP in symbolic regression, some parameters are assigned as shown in Table 2. Some special operations, such as *sin*, *cosine*, *log*, *exponent*, are used in order to fit nonlinear models more efficiently. In addition, the fitness function used in symbolic regression can be defined as

$$\text{Fitness function} = \sum_{i=1}^{m} | y_i - \hat{y}_i | \tag{6}$$

where *m* is the number of DMUs, y_i denotes the observe output of the *i*th firm and $\hat{y}_i$ is the predict output of the *i*th firm which is obtained by symbolic regression.

Table 2. The parameter settings in symbolic regression

Parameter	Value
Population Size	20
Fitness Function	$\sum_{i=1}^{m} \vert y_i - \hat{y}_i \vert$
Function Set	$\{ +, -, \times, \div, sin, cos, log, e \}$
Terminal Set	{random, 1, 2, 3, 4, 5, x_1, x_2 }
Maximum Number of Generation	200
Maximum Tree Depth	10
Crossover Rate	0.5
Mutation Rate	0.01

*{random} denotes the value randomly generated between [0,1].

Here, we demonstrate the fitness and tree-level of GP for Cobb-Douglas form with the sample size is equal to 25 as shown in Figures 3 and 4, respectively, other forms or sample sizes are similar to this situation.

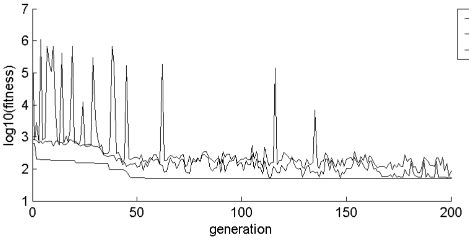

Fig. 3. The fitness of generations

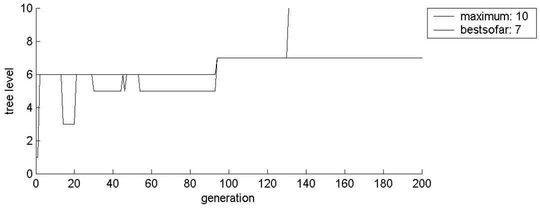

Fig. 4. The level of the GP-tree

Next, Spearman's rank correlation is used to test the correlation between the true and estimated technical efficiency of DEA, DFA and GP-based DFA models. We adopt input-oriented DEA with constant return to scale (CRS) in this simulation. On the other hand, Cobb-Douglas production form is assumed in DFA. Note that no further assumption is needed for the proposed method. These results of the above models are listed in Table 3.

Table 3. The comparisons of DEA, DFA and GP-DFA with Spearman's correlation

Production Function	*N*	**DEA**	**DFA**	**GP-DFA**
Linear Function	25	0.965	0.885	0.945
	50	0.840	0.850	0.932
	100	0.922	0.843	0.917
Average		**0.909**	**0.859**	**0.931**
Cobb-Douglas Function	25	0.988	0.9723	0.929
	50	0.979	0.8860	0.894
	100	0.985	0.9080	0.965
Average		**0.984**	**0.9221**	**0.922**
Nonlinear Function	25	0.826	0.1100	0.765
	50	0.577	0.1681	0.694
	100	0.649	0.1574	0.669
Average		**0.684**	**0.1452**	**0.709**

5 Discussion

From the results of Monte Carlo simulation, we can conclude that DFA is much suitable for correct or simple linear production functions, because DFA perform well only in the linear and Cobb-Douglas functions. However, it poses problems when the production function is nonlinear. Therefore, if the frontier of DMUs belongs to nonlinear, DFA is not suitable for dealing with this problem, because it is hard for researcher to specify the correct nonlinear production function. On the other hand, since DEA does not need to consider the form of production functions, it is more robust and outperforms to DFA in all kinds of the production functions. However, DEA suffers the problems of sensitive to outlier and ignoring the measurement error.

In contrast, unlike DFA has the problem of misspecification, GP-based DFA can search the approximate true production function automatically from simulated data sets. According to the simulated results, it can be seen that GP-based DFA performs well not only for linear functions but also for nonlinear functions. Even though GP-based DFA does not perform the best in all situations, it shows the great ability for dealing with nonlinear situations. In addition, the performance of GP-DFA is not affected by different sample sizes and is more robust than that of DEA and DFA.

The advantage of GP-based frontier models, e.g., GP-based DFA and GP-based SFA, is obvious. Without needing the prior knowledge, GP-based models can obtain the approximate true production function from empirical data sets. In addition, it can provide a solution for dealing with the problem of misspecification in traditional regression-based frontier models. Although only GP-based DFA is discussed in this paper, GP-based SFA should also share the advantages above and can be explored in further research.

6 Conclusion

Although many parametric methods have been proposed to measure the efficiency of DMUs, the form of production function is hard to determine in advance. It is dangerous to use these regression-based methods when the production function is mis-specified. In this paper, the problem of misspecification in regression-based frontier models is highlighted and symbolic regression is used here to fit approximate optimal production function automatically. Three models, including DEA, DFA and GP-based DFA, are compared using Monte Carlo simulation. From the simulated results, it can be shown that GP-based DFA is very suitable for dealing with the problem of measuring the technical efficiency of MUs, while the relation between input and output variables are unknown.

References

1. Cooper, W.W., Tone, K.: Measure of Inefficiency in Data Envelopment Analysis and Stochastic Estimation. European Journal of Operational Research 99, 72–88 (1997)
2. Ruggiero, J.: A New Approach for Technical Efficiency Estimation in Multiple Output Production. European Journal of Operational Research 111, 369–380 (1998)
3. Chen, T.Y.: A Comparison of Chance-Constrained DEA and Stochastic Frontier Analysis: Bank Efficiency in Taiwan. The Operational Research Society 53, 492–500 (2002)
4. Schmidt, P.: Production Frontier Function. Econometric Reviews 4, 289–328 (1985)
5. Greene, W.: The Econometric Approach to Efficiency Analysis. In: Fried, H.O., Lovell, C.A.K., Schmidt, S.S. (eds.) The Measurement of Productive Efficiency: Techniques and Applications, pp. 68–119. Oxford University Press, New York (1993)
6. Giannakas, K., Tran, K.C., Tzouvelekas, V.: On the Choice of Functional Form in Stochastic Frontier Modeling. Empirical Economics 25, 75–100 (2003)
7. Gonzalez, D.S., Castro, A.V.: Comparing Neural Networks and Efficiency Techniques in Non-linear Production Function. In: The Seventh European Workshop on Efficiency and Productivity Analysis at University of Oviedo, Spain (2001)
8. Wang, S.: Adaptive Non-parametric Efficiency Frontier Analysis a Neural-Network Based Model. Computers & Operations Research 30, 279–295 (2003)
9. Wang, S.: Nonparametric Econometric Modelling: A Neural Network Approach. European Journal of Operational Research 89, 581–592 (1996)
10. Castillo, F., Marshall, K., Green, J., Kordon, A.: A Methodology for Combining Symbolic Regression and Design of Experiments to Improve Empirical Model Building. In: Genetic and Evolutionary Computation Conference, pp. 1975–1985 (2003)
11. Koza, J.: Genetic Programming: On the Programming of Computers by Means of Natural Selection. MIT Press, Cambridge (1992)
12. Davidson, J.W., Savic, D.A., Walters, G.A.: Symbolic and Numerical Regression: Experiments and Applications. Information Sciences 150, 95–117 (2003)
13. Huang, J.J., Tzeng, G.H., Ong, C.S.: Two-stage genetic programming (2SGP) for the credit scoring model. Applied Mathematics and Computation 174, 1039–1053 (2006)
14. Ong, C.S., Huang, J.J., Tzeng, G.H.: Building credit scoring models using genetic programming. Expert Systems with Applications 29, 41–47 (2005)