Available online at http://ijdea.srbiau.ac.ir

Int. J. Data Envelopment Analysis (ISSN 2345-458X)

Vol.2, No.3, Year 2014 Article ID IJDEA-00236,7 pages

Research Article

International Journal of Data Envelopment Analysis Science and Research Branch (IAU)

A Neural Network Model to Solve DEA Problems

S. Dolatabadi[*a], H. Rezai Zhiani[a]

(*a*) *Department of Mathematics, Islamic Azad University, Mashhad Branch, Mashhad, Iran*

Received 25 April 2014, Revised 15 June 2014, Accepted 18 August 2014

Abstract

The paper deals with Data Envelopment Analysis (DEA) and Artificial Neural Network (ANN). We believe that solving for the DEA efficiency measure, simultaneously with neural network model, provides a promising rich approach to optimal solution. In this paper, a new neural network model is used to estimate the inefficiency of DMUs in large datasets.

Keywords: Data Envelopment Analysis (DEA), Neural Networks.

1. Introduction

Data envelopment analysis (DEA), occasionally called frontier analysis, was first put forward by Charnes, Cooper and Rhodes in 1978 [1]. It is a performance measurement technique which can be used for evaluating the relative efficiency of decision-making units (DMU's) in organizations. The major advantages of DEA are its allowing the relative efficiency to change over time and requiring no prior assumption on the best solution frontier; therefore, lots of businesses or organizations have applied DEA to find their operating performances for further making decisions on the efficiency improvement. Those inefficient DMUs can be identified and proposed to make up their input resources and/or generated benefits. The identification has been performed widely by linear programming (LP) technique [2,3,4]; yet, the serious dependence on the number of DMUs causes the LP technique to a longer DEA computation time. To overcome this limitation, an alternative approach seems to be needed.

DEA for a large dataset with many inputs/outputs would require huge computer resources in terms of memory and CPU time. This paper proposes a neural network Data Envelopment Analysis to address

[*] Corresponding author: dolatabadii@gmail.com

this problem for the very large scale datasets now emerging in practice. Neural network requirements for computer memory and CPU time are far less than that needed by conventional DEA methods and can therefore be a useful tool in measuring the efficiency of large datasets.

This paper explores an alternative algorithm using a neural network to estimate the efficiency of DMUs (and inefficiency of DMUs) in large datasets. This method offers considerable computational savings. We use a neural network model as a solution tool in measuring the efficiency (and inefficiency) of large datasets simultaneously.

The paper unfolds as follows. The DEA models are explained in Section 2. Section 3 describes a new neural network algorithm for DEA (NNDEA). Finally, simple numerical examples are provided for the sake of illustration.

2. Data envelopment analysis (DEA) models

DEA is a non-parametric approach for measuring relative efficiency that produces a single aggregate measure of relative efficiency among comparable units (called DMUs) that is a function of the inputs and outputs of processes operating at the DMUs. DEA defines relative efficiency as the ratio of the sum of weighted outputs to the sum of weighted inputs:

$$\text{DEA efficiency}=\frac{\text{Sum of weighted outputs}}{\text{Sum of weighted inputs}}$$

The more output produced for a given amount of resources, the more efficient (i.e., less wasteful) is the process. The problem is how to weight each of the individual input and output variables, expressed in their natural units; solving for these weights is the fundamental essence of DEA.

The following derivations of the DEA linear programming formulation follow that of Charnes et al. [1]. The efficiency of the *r*th DMU, w_r , can be obtained by solving the following linear DEA formulation.

Model: Charnes et al. Modified Data Envelopment Analysis Model (DEA Model)

$$Max \qquad w_r = \sum_{j=1}^{J} u_j O_{jr} \tag{1}$$

$$s.t.$$

$$\sum_{i=1}^{I} v_i I_{ir} = 1 \tag{2}$$

$$\sum_{j=1}^{J} u_j O_{jk} - \sum_{i=1}^{I} v_i I_{ik} \le 0 \qquad k = 1,\dots,K \tag{3}$$

$$u_j, v_i \ge \varepsilon \qquad i = 1,\dots,I \ , \ j = 1,\dots,J$$

where ε is an infinitesimal value and

i= 1,..., I inputs used at DMU,

j= 1,..., J outputs produced at DMU,

k= 1,..., r, . . . , K DMUs.

Parameters*:*

O_{jk} = amount of the jth output for the kth DMU,

I_{ik} = amount of the ith input for the kth DMU.

Decision variables:

u_j = the weight assigned to the jth output,

v_i = the weight assigned to the ith input.

The typical DEA solution process consists of sequentially solving DEA model for each DMU. The DEA model solution process needs to be modified to allow for the DEA efficiencies of all the DMUs to be calculated in one linear program. We define a new variable d_r as the level of inefficiency of DMU r ($d_r = 1 - w_r$). The DEA model to solve all for all DMUs simultaneously is expanded in the following manner [6]:

Model: Simultaneous DEA (SDEA)

$$Max \quad \sum_r (1-d_r) = \sum_r w_r \tag{4}$$

$$s.t.$$

$$\sum_{i=1}^{I} v_{ri} I_{ir} = 1 \qquad \forall r, \tag{5}$$

$$\sum_{j=1}^{J} u_{rj} O_{jr} + d_r = 1 \qquad \forall r, \tag{6}$$

$$\sum_{j=1}^{J} u_{rj} O_{jk} - \sum_{i=1}^{I} v_{ri} I_{ik} \leq 0 \qquad \forall k; \forall r; k \neq r \tag{7}$$

$$u_{rj}, v_{ri} \geq \varepsilon \qquad \forall j, i, r$$

Where

Decision variables:

u_{rj} = the weight assigned to the jth output for DMU r,

v_{ri} = the weight assigned to the ith input for DMU r.

To allow for the simultaneous solution of the DEA model for all DMUs, the objective function (4) now maximizes the sum of the efficiencies. The constraints in (5) require the sum of DMU r's weighted inputs to be equal to 1. The constraints in (6) define efficiency as the sum of DMU r's weighted outputs. The constraints in (7), require the sum of each set of weighted outputs to be less than the corresponding sum of weighted inputs.

3. New NN-DEA Model

In this section we demonstrate a neural network model to estimate the inefficiency of DMUs (and efficiency of DMUs) in large datasets.

We transform SDEA model to a neural network model [5]. In general, if the penalty method is applied to solve SDEA model, we can obtain an unconstrained optimization problem:

$$\text{Min } P(x) = -E(x) + \frac{L}{2}\sum_{r=1}^{K}\Big(\sum_{i=1}^{6}(h_{ir}^{+}(x_r))^2 + \sum_{k=1,k\neq r}^{K}(h_{6+k,r}^{+}(x_r))^2\Big) \tag{8}$$

where L is a positive number, K is the total number of DMUs, $r=1,\dots,K$ and

$$V_r = [v_{r1},\dots,v_{rI}],\ U_r = [u_{r1},\dots,u_{rJ}]$$
$$x_r = [V_r, U_r, d_r]$$
$$X = [x_r]^T$$
$$I_r = [I_{1r},\dots,I_{Ir}]^T,\ O_r = [O_{1r},\dots,O_{Jr}]^T$$
$$E(X) = \sum_{r=1}^{K}(1-d_r)$$
$$h_{1r} = V_r.I_r - 1$$
$$h_{2r} = 1 - V_r.I_r$$
$$h_{3r} = U_r.O_r + d_r - 1$$
$$h_{4r} = 1 - U_r.O_r - d_r$$
$$h_{5r} = \varepsilon - U_r$$
$$h_{6r} = \varepsilon - V_r$$
$$h_{6+k,r} = U_r.O_k - V_r.I_k \qquad k \neq r$$
$$h_{ir}^{+}(x_r) = Max\{0, h_{ir}(x_r)\} = \begin{cases} 0 & h_{ir}(x_r) < 0 \\ h_{ir}(x_r) & h_{ir}(x_r) \geq 0 \end{cases} \quad (i=1,\dots,6),\ (r=1,\dots K)$$
$$h_{6+k,r}^{+}(x_r) = Max\{0, h_{6+k,r}(x_r)\} = \begin{cases} 0 & h_{6+k,r}(x_r) < 0 \\ h_{6+k,r}(x_r) & h_{6+k,r}(x_r) \geq 0 \end{cases} \quad (k=1,\dots,K),\ (r=1,\dots K), r \neq k$$

Since that in SDEA model objective function (*E(X)*) and all constraint are convex, demonstrate following NN-DEA model:

$$E^{'}(t)=-\frac{\partial E(X(t))}{\partial X}+L\sum_{r=1}^{K}\Big(\sum_{i=1}^{6}h_{ir}^{+}(x_r(t))\frac{\partial h_{ir}(x_r(t))}{\partial x_r}+\sum_{k=1,k\neq r}^{K}h_{6+k,r}^{+}(x_r(t))\frac{\partial h_{6+k,r}(x_r(t))}{\partial x_r}\Big) \tag{9}$$

where

$$\frac{\partial E}{\partial X}=\begin{bmatrix}\frac{\partial E}{\partial v_{11}} & \frac{\partial E}{\partial v_{21}} & \cdots & \frac{\partial E}{\partial v_{K1}}\\ \frac{\partial E}{\partial v_{12}} & \frac{\partial E}{\partial v_{22}} & \cdots & \frac{\partial E}{\partial v_{K1}}\\ \vdots & \vdots & \vdots & \vdots\\ \frac{\partial E}{\partial v_{1I}} & \frac{\partial E}{\partial v_{2I}} & \cdots & \frac{\partial E}{\partial v_{KI}}\\ \frac{\partial E}{\partial u_{11}} & \frac{\partial E}{\partial u_{21}} & \cdots & \frac{\partial E}{\partial u_{K1}}\\ \frac{\partial E}{\partial u_{12}} & \frac{\partial E}{\partial u_{22}} & \cdots & \frac{\partial E}{\partial u_{K2}}\\ \vdots & \vdots & \vdots & \vdots\\ \frac{\partial E}{\partial u_{1J}} & \frac{\partial E}{\partial u_{2J}} & \cdots & \frac{\partial E}{\partial u_{KJ}}\\ \frac{\partial E}{\partial d_{1}} & \frac{\partial E}{\partial d_{2}} & \cdots & \frac{\partial E}{\partial d_{K}}\end{bmatrix} \quad \text{and} \quad \frac{\partial h_{mr}}{\partial x_r}=\begin{bmatrix}\frac{\partial h_{mr}}{\partial v_{r1}}\\ \frac{\partial h_{mr}}{\partial v_{r2}}\\ \vdots\\ \frac{\partial h_{mr}}{\partial v_{rI}}\\ \frac{\partial h_{mr}}{\partial u_{r1}}\\ \frac{\partial h_{mr}}{\partial u_{r2}}\\ \vdots\\ \frac{\partial h_{mr}}{\partial u_{rJ}}\\ \frac{\partial h_{mr}}{\partial d_{r}}\end{bmatrix} \quad \begin{matrix}(m=1,2,3,4,5,6,\ldots,6+K-1)\\ (r=1,2,3,\ldots,K)\end{matrix}$$

Proposition 1. If for any L (8) has an optimal solution, and if for system (9) we can find a state variable $x(t)$ such that the neural network (9) is asymptotically stable at x^*, then the optimal solution to (8) will be the equilibrium state of (9). [5]

Proposition 2. Under the penalty method, $P(x)$ of (8) is a Lyapunov function of system (9). [5]

4. Numerical Example

Consider table 1 for 3 DMUs.

Table 1. Input/Output Data For Example

DMU	I_1	I_2	O
1	1	1	1
2	2	1	1
3	3	2	1

with respect to SDEA model we have following model

$$
\begin{aligned}
&\text{Max} \;\; (1-d_1)+(1-d_2)+(1-d_3) \\
&s.t. \\
&\quad v_{11}+v_{12}=1 \\
&\quad 2v_{21}+v_{22}=1 \\
&\quad 3v_{31}+2v_{32}=1 \\
&\quad u_{11}+d_1=1 \\
&\quad u_{21}+d_2=1 \\
&\quad u_{31}+d_3=1 \\
&\quad u_{11}-(2v_{11}+v_{12})\le 0 \\
&\quad u_{11}-(3v_{11}+2v_{12})\le 0 \\
&\quad u_{21}-(v_{21}+v_{22})\le 0 \\
&\quad u_{21}-(3v_{21}+2v_{22})\le 0 \\
&\quad u_{31}-(v_{31}+v_{32})\le 0 \\
&\quad u_{31}-(2v_{31}+v_{32})\le 0 \\
&\quad u_{11},u_{21},u_{31},v_{11},v_{12},v_{21},v_{22},v_{13},v_{23}\ge \varepsilon
\end{aligned}
$$

For measure inefficiency (and efficiency) of DMUs we apply NN-DEA model (9) and use Euler method for solving our neural network. Thus we have following neural network model for this example:

$$E'(t)=-\frac{\partial E(X(t))}{\partial X}+L\sum_{r=1}^{3}\Big(\sum_{i=1}^{6}h_{ir}^{+}(x_r(t))\,\frac{\partial h_{ir}(x_r(t))}{\partial x_r}+\sum_{k=1,k\neq r}^{3}h_{6+k,r}^{+}(x_r(t))\,\frac{\partial h_{6+k,r}(x_r(t))}{\partial x_r}\Big)$$

We select $L=1000$, $\varepsilon=0.000001$ and use Euler method with $n = 3000$ iteration for solving above neural network and obtain state variable $X(t)$. The state variable $X^*(t)$ (or optimal solution of SDEA model) is $d_1^*=0,\ d_2^*=0.0000001,\ d_3^*=0.5000005$ and

$V_1^*=[0.9999990, 0.000001]$, $V_2^*=[0.000001, 0.9999980]$, $V_3^*=[0.000001, 0.4999985]$
$U_1^*=[1.000000]$, $U_2^*=[0.9999990]$, $U_3^*=[0.4999995]$

Thus DMU1 and DMU2 is efficient and DMU3 is inefficient.

5. Conclusions

DEA is a non-parametric method that is widely used for measuring the efficiency and productivity of Decision Making Units. DEA for a large dataset with many input/output variables and/or many DMUs would require huge computer resources in terms of memory and CPU time. This paper used a new neural network for DEA to introduce an alternative algorithm and approach to estimating the efficiency of DMUs in large datasets. We use a new NN-DEA to find the optimal efficiency among multiple DMUs of a DEA problem.

6. References

[1] Charnes, A., Cooper, W.W, and Rhodes, E., Measuring the efficiency of decision making units, European Journal of Operations Research, vol.2, no.6, 1978, pp.429-44.

[2] Seiford, L.M. and Thrall, R.M., Recent developments in DEA: the mathematical Programming approach to frontier analysis. Journal of Econometrics, vol.46, 1990, pp.7-38.

[3] Bowlin, W.F., Measuring Performance: An Introduction to Data Envelopment Analysis (DEA), Journal of Cost Analysis, Fall, 1998, pp.3-27.

[4] Bouchard, G., Girard, S., Iouditski, A., and Nazin, A., Some linear programming methods for frontier estimation, Applied Stochastic Models in Business and Industry, vol.21, no.2, 2005, pp.175-185.

[5] Effati, S., and Jafarzadeh, M., A new nonlinear neural network for solving a class of constrained parametric optimization problems, Applied Mathematics and Computation. 186, 2007, p. 814-819.

[6] Klimberg, R.K., and Ratick, S.J., Modeling data envelopment analysis (DEA) efficient location-allocation decisions, Computers and Operations Research, 35, 2008, p. 457-74