Internal Layers Method in Presence of Special Data Structure in Data Envelopment Analysis

B. Daneshian

Dept. of Math, Islamic Azad University
Lahijan, Iran

F. Hosseinzadeh Lotfi

Dept. of Math, Science & Research Branch
Islamic Azad University, Tehran, Iran

H. Zhiani Rezai

Dept. of Math, Islamic Azad University
Mashhad, Iran

Abstract

Data Envelopment Analysis (DEA) measures the relative efficiency of DMUs (Decision Making Units) with multiple inputs and multiple outputs. Sometimes especial data structure of some DMUs enforces efficiency or inefficiency to others. This may make the DM (Decision Maker) unsatisfied; since, indeed, the result does not assist him (or her) to analyze the system. Internal Layers Method (LIM) uses the routine models with a different technique to overcome these deficiencies.

Keywords Data Envelopment Analysis, Efficiency
1 Introduction

Data Envelopment Analysis is a nonparametric method of evaluating efficiency of systems with multiple inputs and multiple outputs. Suppose a system contains \( n \) DMUs. For each \( DMU_j (j = 1, ..., n) \) input vector, \( x_j \in R^m_+ \), and output vector, \( y_j \in R^s_+ \), define as follows:

\[
x_j = (x_{1j}, ..., x_{mj})
\]

\[
y_j = (y_{1j}, ..., y_{sj})
\]

Rational input oriented model of CCR \(^2\) for evaluating efficiency of \( DMU_p (p \in \{1, ..., n\}) \) is

\[
\begin{align*}
\text{Max} & \quad \frac{u^t y_p}{v^t x_p} \\
\text{s.t.} & \quad \frac{u^t y_j}{v^t x_j} \leq 1, \; j = 1, \ldots, n, \\
& \quad u \geq 0, \; v \geq 0
\end{align*}
\]

where \( u \) is a \( s \)-vector and \( v \) is a \( m \)-vector of unknowns. Constraints of above model indicate competitive attribute of one DMU against others. Corresponding linear model of (1) is \(^3\)

\[
\begin{align*}
\text{Max} & \quad u^t y_p \\
\text{s.t.} & \quad v^t x_p = 1 \\
& \quad u^t y_j - v^t x_j \leq 0, \; j = 1, \ldots, n, \\
& \quad u \geq 0, \; v \geq 0
\end{align*}
\]

\( u_r (r = 1, ..., s) \) and \( v_i (i = 1, ..., m) \) can be mean weights of outputs and inputs. Then, (1) tries to find the maximum of weighted outputs over weighted inputs of \( DMU_p \) when weighted outputs over weighted inputs of all DMUs restricted to one and all weights must be nonnegative. Frankly, (1) and (2) are problems in weights space. \( DMU_p \) is called \textit{efficient} if in its assessment with (2)

\(^2\)Charnes, Cooper, Rodes. See [2].

\(^3\)For more details see [1] and [2].
(1) $u^* y_p = 1$.
(2) $(u^*, v^*) > 0$, where $(u^*, v^*)$ is optimal solution of (1) \(^4\).

Consider the system contains 9 DMUs with one input and two outputs (Table 1). Using model (2), it can be observed that the only efficient DMU is DMU1, and the others are all inefficient (Table 2). This is because of especial structure of data of DMU1. It has high outputs with the same input as other DMUs. Indeed, DMU1 dominates all others (See Fig. 1.).

2 Internal Layers Method

Sometimes, in presence of especial data structure (for example in presence of outliers) difference in performance of DMUs is so high that such DMUs impose zero weight on inputs or outputs of some other DMUs. Method of Internal Layers is categorical method; in which DMUs is grouped in such a way in each group only similar DMUs exist. Suppose a system contains $n$ DMUs. In first stage, efficient DMUs will be found. They are members of category One. We put these efficient members in $C_1$. Then all of members of $C_1$ are deleted from system and again efficient DMUs will be found. They are members of category Two. We put these efficient members in $C_2$. This process will continue until there is no DMU in system. DMUs in category $C_k (k \in \{1, \ldots, l\})$ make frontier of $k$-th layer. Members of $C_k$, are all efficient DMUs among the set $\{1, \ldots, n\} - \{C_1 \cup \ldots \cup C_{k-1}\}$. In the other world only members of $C_k$ are on the frontier of $k$-th layer. Consider the following model

$$\begin{align*}
\text{Max} & \quad u^t y_p \\
\text{s.t.} & \quad v^t x_p = 1 \\
& \quad u^t y_j - v^t x_j \leq 0 \quad , \quad j \in C_0 \\
& \quad u \geq 0, v \geq 0
\end{align*}$$

(3)

where $C_0$ is a subset of $\{1, \ldots, n\}$. In order to find members of each category, the following algorithm can be used.

Algorithm.

\(^4\)See [1].
Step 1. \( k = 1, C_0 = \{1, ..., n\} \)
Step 2. For each \( p \in C_0 \), if the model (3) indicates that \( DMU_p \) is efficient then \( p \in C_k \).
Step 3. \( C_0 = C_0 - C_k \)
Step 4. \( k = k + 1 \)
Step 5. If \( C_0 = \emptyset \) go to 2, otherwise stop.

In traditional DEA methods the improvement scenario is based on moving an inefficient DMU from an inner point toward frontier. Different models suggest different path and hence different ways \(^5\). But as can be seen, in special cases moving to efficient frontier is not an easy job. Here with layers in hand, for each DMU in \( C_k (k = \{2, ..., l\} \) the first improvement scenario, is going to the above layer, \( k - 1 \).

3 Illustration Example

Consider the following system contains 9 DMUs with one input and two outputs.

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x )</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>( y_1 )</td>
<td>35</td>
<td>17</td>
<td>6</td>
<td>5</td>
<td>9</td>
<td>12</td>
<td>6</td>
<td>9</td>
<td>4</td>
</tr>
<tr>
<td>( y_2 )</td>
<td>32</td>
<td>14</td>
<td>2</td>
<td>1</td>
<td>8</td>
<td>5</td>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 1.

Using the algorithm, for \( k = 1 \), the following result is yield

\(^5\)See [1], [2], [3]
Table 2.

As can be seen, except $DMU_1$ that is efficient, for other DMUs, the first weight is zero and the second weight the same. This indicates special data structure. Here $C_1 = \{1\}$ and $C_0$ changes to $\{2, \ldots, n\}$. Using the algorithm, for $k = 2$, the following result is yield.

<table>
<thead>
<tr>
<th>Eff.</th>
<th>$u_1$</th>
<th>$u_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1.00</td>
<td>0.029</td>
</tr>
<tr>
<td>3</td>
<td>0.35</td>
<td>0.059</td>
</tr>
<tr>
<td>4</td>
<td>0.24</td>
<td>0.059</td>
</tr>
<tr>
<td>5</td>
<td>0.57</td>
<td>0.00</td>
</tr>
<tr>
<td>6</td>
<td>0.71</td>
<td>0.059</td>
</tr>
<tr>
<td>7</td>
<td>0.35</td>
<td>0.059</td>
</tr>
<tr>
<td>8</td>
<td>0.53</td>
<td>0.059</td>
</tr>
<tr>
<td>9</td>
<td>0.24</td>
<td>0.059</td>
</tr>
</tbody>
</table>

Only $DMU_2$ is efficient and other DMUs, except $DMU_5$, have their second weight zero. Therefore $C_2 = \{2\}$ and $C_0$ changes to $\{3, \ldots, n\}$. Continuing the method, we will have $C_3 = \{5, 6\}, C_4 = \{7, 8\}, C_5 = \{3\}$ and $C_6 = \{4, 9\}$. Fig. 1. illustrates the layers.

Consider $DMU_5$ that is in $C_3$. For this DMU the first aim is going to second layer. Suppose because of limits in sources, the only possibility is
increment in the first output. It can be accomplished by solving the following linear program.

\[
\begin{align*}
\min & \quad \alpha \\
\text{s.t.} & \quad vx_{15} = 1 \\
& \quad u_1(y_{15} + \alpha) + u_2y_{25} - vx_{15} = 0 \\
& \quad u_1y_{1j} + u_2y_{2j} - vx_{1j} \leq 0, \quad j \in C_0 - \{5\} \\
& \quad u_1, u_2, v \geq 0
\end{align*}
\]

where \( C_0 = \{2, ..., n\} \)

4 Conclusion

Although DEA models give us good information about efficiency and other attributes of DMUs, but sometimes, because of special structure of data, unsatisfactory results will be yield. Here some manipulation in models is needed. Internal Layers Method is a technique to overcome the deficiency that is arisen from special structure of DMUs. It helps the Decision Maker to have a wider insight to the problem and DMUs attributes.
References


Received: December 25, 2005