Data envelopment analysis with missing values: an approach using neural network

B. Dalvand, F. Hosseinzadeh Lotfi, G. R. Jahanshahloo

Department of Mathematics, Science and Research Branch, Islamic Azad University, Tehran, Iran

Abstract
The data envelopment analysis (DEA) models developed with the assumption that input and output data from all of the decision making units (DMUs) to be evaluated are available. So, the need to apply an appropriate approach so that it handles cases includes DMUs whose some data are missing, has been an important issue. In this paper, we consider the case of missing values in one component of the output vector of a certain unit. We first apply a DEA-base clustering method to know the cluster that this unit belongs to and then predicted the missing value by training the neural network algorithm with this cluster. Finally, we also apply EM algorithm and Monte Carlo simulation to compare obtained results by an illustrative example.

Keywords:
Data Envelopment Analysis (DEA), Missing data, DEA-based clustering, Neural network

1. Introduction

To evaluate the performance of a set of homogeneous Decision Making Units (DMUs) with multiple inputs and multiple outputs, by using data envelopment analysis (DEA), input and output data from all of units are required. If any observation of a DMU in the group is missing, for any reason, then this DMU must be ignored from the group in order to evaluate all other DMUs. In so doing, the number of DMUs is decreased and the resulting efficiencies will be biased. Therefore, it is desirable to keep those DMUs in the group and apply an appropriate approach so that it handles this case. Facing with missing value could be included two situations: a) The situation where a DMU consumes resources to produce an output, but fails to do so, or else the amount of output exists, but is unknown; b) the situation where the DMU intentionally does not produce that output. One that has been addressed in the literature is the situation (a). One prescribed solution to face with this problem is that the missing data are replaced by a value (e.g. by taking the average value of other units). Kuosmanen (2001) uses dummy entries (zero for the outputs and sufficiently large number for inputs). Smirlis et. al (2006) presented an approach based on interval DEA and used interval estimations for the missing values applying deterministic techniques. Kao and Liu (2007) proposed a DEA approach based on fuzzy theory and replaced the missing values with intervals by using observed data. Park et al. (2008) proposed a method for estimating parameters in logistic linear models involving missing data and used the Monte Carlo EM algorithm. It is worth mentioning that the EM (Expectation-Maximization) algorithm is a common algorithm which used in statistical estimation when some of the random variables involved are considered missing or incomplete.
In recent years, using the neural network as an artificial intelligence approach has been widely applied to forecast because of its ability to extract useful knowledge from vast data.
So far, the missing data have been analyzed in many ways, but the use of data mining and the neural networks to predict the missing values has not been discussed in DEA literature.
In this paper, the issue is that one component of output vector is missing for a certain DMU. We first apply a clustering method based on DEA model and then the missing values are replaced by the predicted value using the neural network algorithm.
The paper is organized in 5 sections. Section 2 mentions a clustering method. In section 3 we first address a clustering method and then predict the missing value by training the neural network. Section4 illustrates an example. Finally, Section 5 gives conclusions.

2. DEA-based clustering method with a missing output

Cluster analysis is a method for classifying like groups of a data set into the same cluster and unlike groups into different clusters. To cluster the data with input and output items, Po et al. (2009) employed the piecewise production functions derived from the DEA method and developed a DEA-based clustering approach. In fact, the basic idea of their approach is using the piecewise production functions and conducting a cluster analysis for a group of DMUs. In this method, each supporting hyperplanes of Production Possibility Set (PPS) represents one cluster. After finding the projection of all DMUs on the efficient frontier, each unit knows the cluster that it belongs to. To do this, input-based or output-based DEA model is solved for each DMU.

Manuscript received February 5, 2017
Manuscript revised February 20, 2017

3. The proposed method

In this part, we use the idea of clustering mentioned in section 2 and suggest a method based on solving a linear programming to obtain the projection point of DMUs on the efficient frontier when some units have missing value in one component of output vector.

Consider a set of n DMUs whose all data are available. The input and output vector of each DMUj (j=1,…, n) is Xj=(x1j,…,xmj) and Yj=(y1j,…,ysj) respectively. Suppose DMU(n+1) with missing kth output (yk(n+1)). So, to assess DMUo (o=1,…, n) we solve the following output-based model which just maximizes the kth output.

$$
\begin{aligned}
&\theta^* = \text{Max } \theta \\
&\text{s.t.} \quad \sum_{j=1}^{n} \lambda_j x_{ij} \le x_{io} \quad i = 1,\dots,m \\
&\sum_{j=1}^{n} \lambda_j y_{rj} \ge y_{ro} \quad r = 1,\dots,s \;\; \& \; r \ne k \\
&\sum_{j=1}^{n} \lambda_j y_{kj} \ge \theta y_{ko} \qquad (1) \\
&\sum_{j=1}^{n} \lambda_j = 1 \\
&\lambda_j \ge 0 \qquad j = 1,\dots,n
\end{aligned}
$$

Theorem 1: Suppose the vector $(\lambda_1^*,\dots,\lambda_n^*\ \theta^*$ be the optimal solution of the Model (1). DMU_o is an efficient DMU if $\theta^* = 1$ and an inefficient DMU if $\theta^* > 1$.

Proof: We just prove that DMU_o is an efficient DMU if $\theta^* = 1$; another case can be proved similarly.

By contradiction, we suppose that DMU_o is an inefficient unit while $\theta^* = 1$. So, there is a point belongs to PPS such as $DMU' = (x_{1o},\dots,x_{mo},y_{1o},\dots,y'_{ko},\dots,y_{so})$ so that $y'_{ko} > y_{ko}$. Since DMU' belongs to PPS, there is the vector $(\lambda'_1,\dots,\lambda'_n)$ so that:

$$
\begin{aligned}
&\sum_{j=1}^{n} \lambda'_j x_{ij} \le x_{io} \quad i = 1,\dots,m \\
&\sum_{j=1}^{n} \lambda'_j y_{rj} \ge y_{ro} \quad r = 1,\dots,s \;\; \& \; r \ne k \\
&\sum_{j=1}^{n} \lambda'_j y_{kj} \ge y'_{ko} \\
&\sum_{j=1}^{n} \lambda'_j = 1 \\
&\lambda'_j \ge 0 \qquad j = 1,\dots,n
\end{aligned}
$$

Let $\theta' = \dfrac{y'_{ko}}{y_{ko}}$. Therefore, the solution $(\lambda'_1,\dots,\lambda'_n,\theta')$ is a feasible solution for Model (1) in which $\theta' > 1$ which contradicts the assumption.

After solving Model (1), the projection point of DMU_o, say, $\overline{DMU}_o = (x_{1o},\dots,x_{mo},y_{1o},\dots,\theta^* y_{ko},\dots,y_{so})$, is on the efficient frontier. We solve this model for all DMUs to find their location on the efficient frontier. Depends on the projection of which DMUs is lying on a same hyperplane, all DMUs will be classified into different clusters. It is worth mentioning that there are several studies that had developed methods for identifying facet members of the Pareto-optimal frontier (e. g., Huang et al. (1997), and Cooper et al. (2007), Jahanshahloo et al. (2007)).

Now, the missing output $y_{k(n+1)}$ is treated as a variable and the following model is solved to obtain the projection of $DMU_{(n+1)}$ on the efficient frontier constructed by given DMUs (DMU_1,…,DMU_n).

$$
\begin{aligned}
&y^*_{k(n+1)} = \text{Max } y_{k(n+1)} \\
&\text{s.t.} \quad \sum_{j=1}^{n} \mu_j x_{ij} \le x_{i(n+1)} \quad i = 1,\dots,m \\
&\sum_{j=1}^{n} \mu_j y_{rj} \ge y_{r(n+1)} \quad r = 1,\dots,s \;\; \& \; r \ne k \qquad (2) \\
&\sum_{j=1}^{n} \mu_j y_{kj} \ge y_{k(n+1)} \\
&\sum_{j=1}^{n} \mu_j = 1 \\
&\mu_j \ge 0 \qquad j = 1,\dots,n \\
&y_{k(n+1)} \ge 0
\end{aligned}
$$

Suppose the vector $(\mu_1^*,\dots,\mu_n^*\ y^*_{k(n+1)}$ be the optimal solution of the Model (2). So, the projection point of DMU_{n+1}, named $\overline{DMU}_{n+1} = (x_{1(n+1)},\dots,x_{m(n+1)},y_{1(n+1)},\dots,y^*_{k(n+1)},\dots,y_{s(n+1)})$, is on the efficient frontier and is lying at least on a defining hyperplane of PPS.

After finding the locations corresponding DMU_j (j={1,…,n}) and DMU_{n+1} on the efficient frontier through Models (1) and (2), respectively, DMU_{n+1} knows the cluster that it belongs to. Actually, our reason for using the idea of clustering is that to know the units that have more compatibility with DMU_{n+1}.

In the next step, the neural network is trained with members of the related cluster to predict the missing output $y_{k(n+1)}$.

Now, the procedure proposed in this paper can be summarized in the following four steps:

Step 1. Solve Model (1) for each DMU. Each DMU is clustered according to the location of its projection point on the efficient frontier.

Step 2. Solve Model (2) for DMU_{n+1}.

Step 3. Let $J = \{j_1, ..., j_t\}$ ($J \subseteq \{1, ..., n\}$) as the set of index of DMUs whose projection point along with $\overline{DMU}_{n+1}$ are lying on a same defining hyperplane of PPS. So, DMU_{n+1} and DMUs $\{DMU_{j_1}, ..., DMU_{j_t}\}$ are classified into the same cluster.

Step 4. Train the neural network with the members of the set $\{DMU_{n+1}, DMU_{j_1}, ..., DMU_{j_t}\}$ to have the amount predicted for the missing output $y_{k(n+1)}$.

Remark 1. The location of DMU_{n+1} may be at the intersection point of the frontier. Consequently, this DMU belongs to more than one cluster. In such case, we suggest two following strategies that the manager can select one of these options in its sole discretion:

a) Train the neural network with the members of each cluster separately. Then among the different obtained results, the one is more compatible with the system be considered.

b) Consider en masse of all members of clusters and train the neural network with them.

4. Illustrative example

In this section, we consider an example, consists of 81 DMUs with two inputs and three outputs whose data is given in Table 1. However, the 3th output of DMU81 is missing.

Table 1. Input–output data and the result of Model (1)

DMU	I1	I2	O1	O2	O3	θ^*
1	112	194	1007	382	700	1
2	66	274.5	1003.5	231.5	840	1.28
3	75	225	81	82	863	1.68
4	75	250	497	212	999	1.45
5	105	22	565	185	795	1
6	95	215	323	134	847	1.86
7	94	343	908	259	1118	1.18
8	120	169	697	304	317	3.36
9	102	320	667	63	576	2.74
10	101	276	1138	71	553	2.57
11	123	247	795	222	1549	1
12	85	263	800	160	1337	1.17
13	109	132	751	161	718	2.19
14	78	180	890.7	220	540	2.46
15	94	370	783	247	1110	1.29
16	101	257	808	193	1157	1.35
17	79	253	507	238	1322	1.08
18	80.1	198	776.9	254.8	520	2.66
19	99	247	263	180	960	1.64
20	109	267	1141	100	373	3.80
21	139	281	784	231	1065	1.42
22	87	222	653	124	1372	1.15
23	100	232	962	176	963	1.55
24	96	401	998	190	190	7.72
25	67	80	92	145	325	1
26	90	417	418	238	557	2.61
27	69	275	662	259	1368	1
28	111	325	1179	250	397	2.81
29	83	125	1125	241	760	1
30	99	205	899	303	637	1.74
31	89	229	950	201	610	2.38
32	102	281	588	190	609	2.58
33	88	124	150	162	915	1.72
34	105	285	571	158	1311	1.20
35	109	159	353	177	664	2.37
36	80.6	267	893.2	226.6	890	1.54
37	94	298	510	225	824	1.80
38	63	274	1345	204	778	1
39	84	331	1339	358	475	1
40	119	325	808	161	966	1.62
41	80	201	46	212	1314	1.14
42	113	269	598	228	612	2.48
43	99	179	1047	69	405	3.40
44	120	150	658	250	178	7.38
45	66	319	1001	139	1006	1.06
46	94	250	1345	202	546	1
47	78	245	764	298	1284	1
48	144	318	346	167	799	1.97
49	111	294	766	305	868	1.42
50	120	276	775	134	735	2.14
51	81	343	687	173	87	17.64
52	90	319	1086	133	1107	1.26
53	93	267	419	233	754	1.95
54	115	198	746	140	476	3.31
55	93	201	248	180	920	1.71
56	109	339	761	179	52	30.32
57	65	129	705	215	353	1
58	92	248	207	208	31	49.28
59	84	103	785	181	1577	1
60	72	326	874	104	499	2.56
61	79.2	180	754.1	231.7	1260	1.14
62	82	198	825	241	450	3.10
63	81	201	830	217	301	4.78
64	76	250	567	92	1254	1.16
65	116	293	406	244	734	2.00
66	103	85	719	245	386	2.05
67	93	76	298	97	1167	1.12
68	98	255	1240	178	1375	1
69	93	178	1019	248	890	1.28
70	69	280	804	177	24	52.90
71	93	300	1086	183	458	3.12
72	84	314	207	213	1240	1.21
73	82	198	1042	243	954	1.16
74	79	283	300	89	488	3.09
75	89	257	846	314	729	1.56

76	65	311	105	327	944	1
77	113	290	644	259	761	1.86
78	106	317	929	113	1210	1.25
79	87.5	260.7	1271.5	185.8	430	2.78
80	98	244	636	120	928	1.69
81	80	200	845	230	?	-

Applying the suggested procedure, the amount of θ^*_j corresponding DMU_j {j=1,…,n} is reported in Table 1 and we have $y^*_{3(81)} = 1381.38$ through Model (2) for DMU_{81}. It must be noted that we apply the algorithm suggested in Jahanshahloo et al. [16] and MATLAB software to find the strong defining hyperplanes of PPS. We find out that DMU_{81} along with efficient DMUs {27, 38, 47, 59, 68, 79} and the projection point of inefficient DMUs {2, 14, 18, 36, 61, 62, 63} are lying on a same defining hyperplane, i. e. *J*={2, 14, 18, 27, 36, 38, 47, 59, 61, 62, 63, 68, 79}. After training the neural network with this DMUs, the amount predicted for the 3*th* output of DMU_{81} is 902.57. Furthermore, we consider a sample set includes DMU_{81} and 13 DMUs from these 80 units which are selected at random. Training the neural network with this random set predicts the amount 1480.24 for the missing output.

Here, we also apply EM algorithm and Monte Carlo simulation as two well-known statistical approaches for set J and random set separately to compare the obtained results with those from neural network. The probability density obtained through Monte Carlo simulation procedure for set J and random set is shown in Fig 1 and Fig 2 respectively.

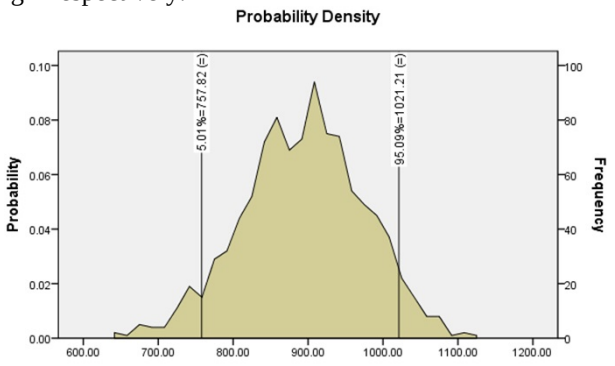

Fig. 1 Probability Density corresponding set J

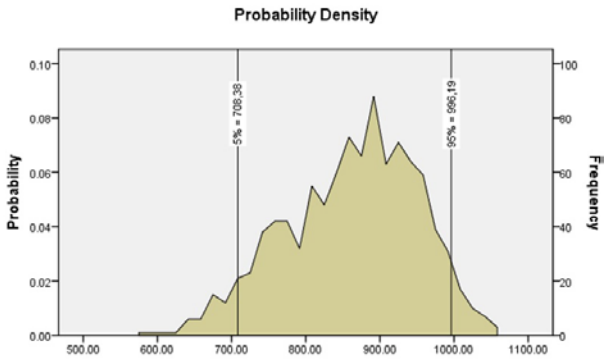

Fig.2 Probability Density corresponding random set

Table 2 gives the summary results in regards of the three procedures for both set J and random set.

Table. 2 The summary results

	Set J	Random set
Neural network	902.57	1480.24
EM algorithm	915	849.4
Monte Carlo simulation	with a probability of 90%: 757.82 - 1021.21	with a probability of 90%: 708.38 - 996.19

With respect to the above results, the estimated amount for 3*th* output of DMU_{81} through the three procedures for set J are very close, while there is a significant gap between neural network result and those from EM algorithm and Monte Carlo simulation considering random set. So, it shows the validity of the proposed method in finding the most appropriate sample set for estimating missing value. In other words, applying the suggested procedure for clustering units lead to the introduction of the most appropriate sample set whose members have more compatibility with the unit with missing value.

5. Conclusion

The main idea of this current paper was to contribute to the use of DEA model and the neural network to address the missing data problem. The issue that has not been discussed in DEA literature. To do so, we first introduced a DEA model for classifying all units into different clusters and then predicted the amount of missing output for a certain DMU by training the neural network with the members of the cluster this DMU belongs to. In fact, the suggested method introduces a set whose members have more compatibility with the unit with missing value. We also elaborated that by considering the corresponding cluster, the estimated amount through

statistical procedures EM algorithm and Monte Carlo simulation are closed to the one from neural network, Whereas, by considering a random set there is not such compatibility between results.

References

[1] Cooper, W.W., Ruiz, J.L., Sirvent, I., 2007. Choosing weights from alternative optimal solutions of dual multiplier models in DEA. European Journal of Operational Research 180, 443–458.

[2] Huang, Z., Li, S.X., Rousseau, J.J., 1997. Determining rates of change in data envelopment analysis. Journal of the Operational Research Society 48, 591–599.

[3] Jahanshahloo, G.R., Lotfi, F.H., Rezai, H.Z., Balf, F.R., 2007. Finding strong defining hyperplanes of production possibility set. European Journal of Operational Research 177, 42– 54.

[4] Kao, C., Liu, S.T., Data Envelopment Analysis With Missing Data, A Reliable Solution Method, Modeling Data Irregularities and Structural Complexities in Data Envelopment Analysis, 291-304.

[5] Kuosmanen, T., 2001. Modelling blank data entries in Data Envelopment Analysis, Econometrics 0210001, Economics Working Paper Archive at WUSTL, available from: http://econwpa.wustl.edu/eprints/em/papers/0210/0210001.abs.

[6] MacQueen, J. 1967. Some methods for classification and analysis of multivariate observations, Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability, 1, pp. 281–297, University of California Press, Berkeley, CA.

[7] Park, J.S, Qian, G. Q., Jun, Y. , 2008. Monte Carlo EM algorithm in logistic linear models involving non-ignorable missing data, Applied Mathematics and Computation 197, 440–450.

[8] Po, R.W., Guh, Y.Y., Yang, M.S., 2009. A new clustering approach using data envelopment analysis, European Journal of Operational Research 199, 276-284.

[9] Smirlis, Y.G., Maragos , E. K. , Despotis, D. K., 2006. Data envelopment analysis with missing values: An interval DEA approach, Applied Mathematics and Computation 177 , 1–10.