A guided example of unsupervised Data Envelopment Analysis

April 26, 2023

This is a guided example of the steps performed in the unsupervised Data Envelopment Analysis (uDEA) algorithm, where a dataset is provided and solved by uDEA and the results compared with DEA.

The data in question for this example consists of $n = 20$ DMUs, with one input ($m = 1$) and one output ($s = 1$), in order to be able to represent it graphically. We report the data inputs and outputs, but internally the algorithm works with the netput notation, simply changing the sign of the value of the input (x) in each DMU. Most figures represent data in netput notation.

DMU	**x**	**y**
1	0.484505	0.669027
2	0.326993	0.445562
3	0.054679	0.155675
4	0.654055	0.322275
5	0.196588	0.239305
6	0.701487	0.596170
7	0.801368	0.376969
8	0.672302	0.407271
9	0.653045	0.725845
10	0.483559	0.123976
11	0.456264	0.302394
12	0.247688	0.345865
13	0.229938	0.436817
14	0.199769	0.394594
15	0.321987	0.425492
16	0.416869	0.337296
17	0.034270	0.103275
18	0.299813	0.525747
19	0.569931	0.673545
20	0.490719	0.576058

Table 1: DMU values

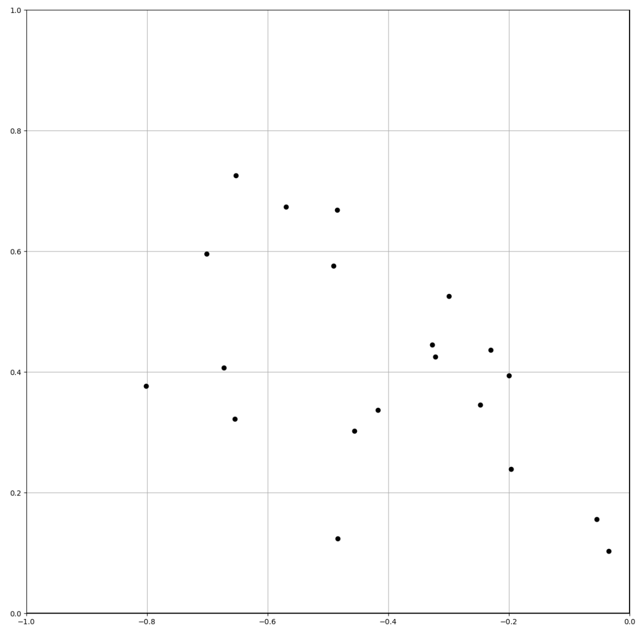

Figure 1: Data plot in netput notation

The uDEA algorithm performs the following steps:

1. The first step of the uDEA algorithm is to choose the number h of hyperplanes desired. We choose $h = n = 20$, that is, we choose to have a hyperplane corresponding to each DMU.

2. We solve the DEA program in its Directional Distance Function formulation with $\mathbf{g} = \mathbf{1}$, in its multiplier form (Model (1)) for each $\mathbf{z}_i$, obtaining the values of $\mathbf{p}_k$, q_k that determine the $h = n$ hyperplanes that will form the feature mapping $\boldsymbol{\phi}$. Furthermore, this step yields values μ_i for each DMU, which we use to calculate $\mu_{min} = \min_i\{\mu_i\}$.

$$
\begin{array}{lll}
\min\limits_{\mathbf{p}_k, q_k} & -\langle \mathbf{p}_k \cdot \mathbf{z}_i \rangle - q_k = -\mu_i & \\
\text{subject to} & \langle \mathbf{p}_k \cdot \mathbf{z}_r \rangle + q_k \leq 0 & \forall r \in \{1, \dots, n\} \\
& \langle \mathbf{p}_k \cdot \mathbf{1} \rangle = 1 & \\
& \mathbf{p}_k \geq \mathbf{0} &
\end{array} \tag{1}
$$

This program projects the DMU $\mathbf{z}_i$ being considered along direction $\mathbf{g} = \mathbf{1}$ until it reaches a hyperplane defined by parameters $\mathbf{p}_k, q_k$ which is the hyperplane closest to the DMU that contains every other DMU on the same half-plane as $\mathbf{z}_i$ (i.e. envelops the data).

We remark that the DEA efficient frontier is defined by the sections of the hyperplanes defined by $\mathbf{p}_k, q_k$ which are closest to the data in each region. The distance at which it can be projected before reaching this hyperplane is $-\mu_i$, so that those DMUs considered efficient by DEA have $\mu_i = 0$. However, different choices of Directional Vector yield different efficiency scores, so that these values for μ_i do not correspond to the DEA efficiencies calculated later. The choice used here corresponds to the l_∞ norm, whereas later we consider the directional vector corresponding to the Farrell output measure.

In particular, with the dataset presented above, solving Linear Model (1) yields the following values in Table 2. Figure 2 shows the projection of DMU $\mathbf{z}_2$ to its closest hyperplane, and Figure 3 represents every hyperplane obtained at this step. Using these values, we then define $\mu_{min} = \min_i\{\mu_i\} = -0.298312$.

Remark: we calculate 20 hyperplanes, one for each DMU, but they are not all distinct. In this example, there are 7 different hyerplanes defined, since multiple DMUs get projected to the same DEA hyperplane. In general, we obtain one distinct hyperplane for each piece of the DEA boundary to which a DMU is projected, as well as some freedom with those DMUs which lie at the corners of the DEA frontier.

DMU	**p_k**		**q_k**	**mu_i**
1	0.436867	0.563133	-0.165087	0.000000
2	0.559998	0.440002	-0.063435	-0.050502
3	0.622170	0.377830	-0.024799	0.000000
4	0.436867	0.563133	-0.165087	-0.269339
5	0.622170	0.377830	-0.024799	-0.056693
6	0.252123	0.747877	-0.378196	-0.109195
7	0.252123	0.747877	-0.378196	-0.298312
8	0.436867	0.563133	-0.165087	-0.229446
9	0.000000	1.000000	-0.725845	0.000000
10	0.583253	0.416747	-0.047930	-0.278300
11	0.559998	0.440002	-0.063435	-0.185888
12	0.622170	0.377830	-0.024799	-0.048225
13	0.583253	0.416747	-0.047930	0.000000
14	0.622170	0.377830	-0.024799	0.000000
15	0.559998	0.440002	-0.063435	-0.056530
16	0.559998	0.440002	-0.063435	-0.148470
17	1.000000	0.000000	0.034270	0.000000
18	0.436867	0.563133	-0.165087	0.000000
19	0.252123	0.747877	-0.378196	-0.018159
20	0.436867	0.563133	-0.165087	-0.055069

Table 2: Parameters corresponding to the calculated hyperplanes and offsets

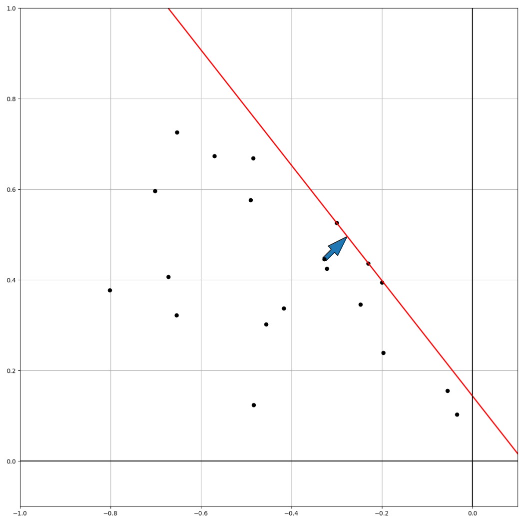

Figure 2: A figure indicating the direction of projection of a DMU (DMU 2) until the corresponding hypeprlane is reached

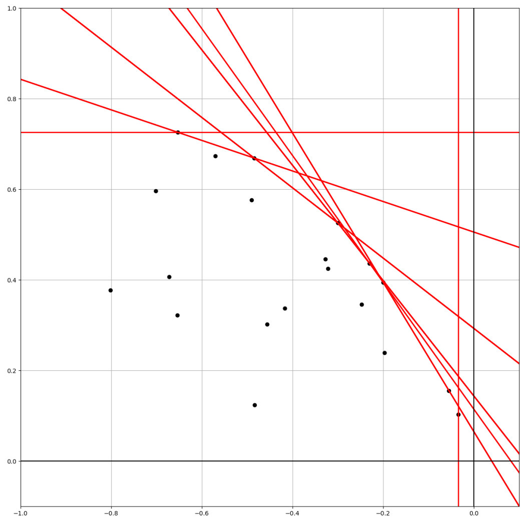

Figure 3: The DEA hyperplanes. These hyperplanes define the DEA technology. With an offset, they are involved in the transformation function of the uDEA algorithm

3. We then insert these values of $\mathbf{p}_k$ and q_k into the transformation function $\boldsymbol{\phi}$. Recall that the PWL transformation function $\boldsymbol{\phi}$ used in the uDEA algorithm is:

$$\boldsymbol{\phi}(\mathbf{z}) = \begin{cases} -z(k) & \text{for } k \in \{1, \ldots, m+s\} \\ -\max\{\mu,\ \langle \mathbf{p}_k \cdot \mathbf{z} \rangle + q_k\} & \text{for } k \in \{m+s+1, \ldots, m+s+h\} \end{cases} \tag{2}$$

In particular, for each $\mathbf{z}_i \in \mathcal{Z}$, $\boldsymbol{\phi}(\mathbf{z}_i)$ is a fixed vector of length $m+s+h = 22$ (1 input + 1 output + 20 DMUs).

The regions where $\langle \mathbf{p}_k \cdot \mathbf{z} \rangle + q_k = \mu$ are turning points or hinges for each of the corresponding components of the transformation function. The effect that this has on the decision function is that these are the regions where the efficient frontier of the technology can have turning points.

For example, when $\mu = 0$, we have $\boldsymbol{\phi}(\mathbf{z}_2) = [0.326993, -0.445562, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]$. In this case, we observe that, due to the first restriction in Model (1), every hyperplane component is negative, so that the maximum attains 0 in every hyperplane component. Thus, we will consider only negative values for μ.

As the values of μ get smaller, some of these hyperplane values get larger than μ, so that the values of various components start improving, until at the value μ_{min}, every will attain the corresponding hyperplane value (or almost every component, as those corresponding to the DMUs which attain μ_{min} in both entries of the maximum).

And for intermediate values, different hyperplane components take values different from mu, as they get "activated" accordingly. For example, when the offset is $\mu = 0.2\mu_{min} = -0.059662$, we have $\boldsymbol{\phi}(\mathbf{z}_2) = [0.326993, -0.445562, 0.057029, 0.050502, 0.059662, 0.057029, 0.059662, 0.059662, 0.059662, 0.057029, 0.059662, 0.052963, 0.050502, 0.059662, 0.052963, 0.059662, 0.050502, 0.050502, 0.059662, 0.057029, 0.059662, 0.057029]$. Here we can observe that, in Figure 4, the DMU $\mathbf{z}_2$ lies on the other side of 4 of the hyperplanes so that the coordinates corresponding to these hyperplanes are now distinct from $-\mu$. Thus, we see the effect of the hyperplane μ on the transformation function.

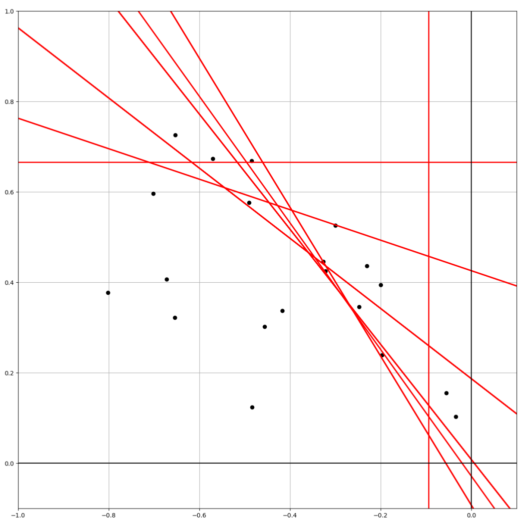

Figure 4: Hyperplanes at offset $0.2\mu_{min}$

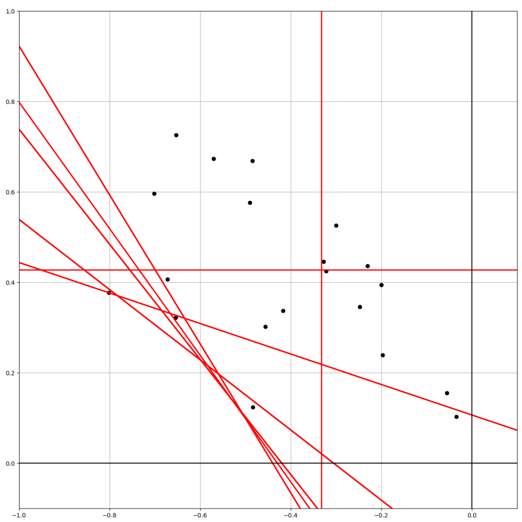

Figure 5: Hyperplanes at maximum offset μ_{min}

For example, when $\mu = \mu_{min} = -0.298312$, we have $\boldsymbol{\phi}(\mathbf{z}_2) = [0.326993, -0.445562, 0.057029, 0.050502, 0.059898, 0.057029, 0.059898, 0.127413, 0.127413, 0.057029, 0.280283, 0.052963, 0.050502, 0.059898, 0.052963, 0.059898, 0.050502, 0.050502, 0.292723, 0.057029, 0.127413, 0.057029]$. The hyperplanes offset by μ_{min} can be seen in Figure 5.

Graphically, we can also observe here the effect of μ in the components of $\boldsymbol{\phi}$ on the hyperplanes. For example, a value of $\mu = \mu_{min}$ corresponds to the hyperplanes being offset all the way to the furthest unit from the DEA efficient frontier, as ilustrated in Figure 5.

This leads to the use of the interval $[\mu_{min}, 0.1\mu_{min}]$ as a range of appropriate parameters for hyperparameter μ.

4. We then do a train-test split of the dataset $\mathcal{Z}$ with $\mathcal{Z}_{train}$ consisting of 70% of the data and $\mathcal{Z}_{test}$ the remaining 30%. This is in order to obtain various estimates of the technology according to different values of the hyperparameters μ and ν, which are then evaluated against the test set, from which a pair of best values is chosen. The role of μ has been discussed above, while ν is a SVM-inherited hyperparameter with the usual interpretation involving the relative weights of the terms of the objective function.

5. At this point, we have defined all values needed to set up the restrictions of the uDEA quadratic Program (3), which involves optimization variables $\mathbf{w}, \boldsymbol{\xi}, \rho$. This program also involves hyperparameters ν and μ, which will be tuned. Everything else in the program is fixed. We now proceed to solve this program using only the training dataset $\mathcal{Z}_{train}$.

$$\min_{\mathbf{w}\in\mathbb{R}^{m+s+h}, \boldsymbol{\xi}\in\mathbb{R}^n, \rho\in\mathbb{R}} \frac{1}{2}\|\mathbf{w}\|^2 + \frac{1}{\nu n}\sum_{i=1}^{n}\xi_i - \rho \tag{3}$$

$$\text{subject to } \langle \mathbf{w} \cdot \boldsymbol{\phi}(\mathbf{z}_i)\rangle \geq \rho - \xi_i \qquad \forall i \in \{1, \ldots, n\} \tag{3a}$$

$$\xi_i \geq 0 \qquad \forall i \in \{1, \ldots, n\} \tag{3b}$$

$$w_j \geq 0 \qquad \forall j \in \{1, \ldots, m+s+h\} \tag{3c}$$

$$\langle \mathbf{w} \cdot \boldsymbol{\phi}(\mathbf{0})\rangle = \rho \tag{3d}$$

6. Then, the hyperparameters ν and μ remain to be tuned. For this example, we choose the interval $\mu \in [\mu_{min}, 0.1\mu_{min}]$ for μ, and we fix $\nu = 0.001$.

   Thus, the potential values for μ are:

$$\mu = [\mu_{min} = -0.298312, -0.231191, -0.164071, -0.096951, -0.029831].$$

7. For each pair of hyperparameter values (μ, ν), we solve Model (3) using only the training set $\mathcal{Z}_{train}$, obtaining an optimal solution $(\mathbf{w}^*, \boldsymbol{\xi}^*, \rho^*)$ for each such combination. This yields a variety of estimates of the technology which we evaluate. Recall that the technology estimated by uDEA is defined by:

$$\hat{T}_{\text{uDEA}} = \{\mathbf{z} \in \mathbb{R}^m_- \times \mathbb{R}^s_+ : \langle \mathbf{w}^* \cdot \boldsymbol{\phi}(\mathbf{z}) \rangle \geq \rho^*\}.$$

8. In order to evaluate each estimate of the technology, we use the test set $\mathcal{Z}_{test}$. For each $\mathbf{z}_i$ in the test set, we solve Program (4) to calculate $\delta_{\text{uDEA}}(\mathbf{z}_i, \mathbf{g}_i)$ using Farrell's output-oriented measure of efficiency, that is $\mathbf{g}_i = (0, \ldots, 0, z_i(m+1), \ldots, z_i(m+s))$. This is a linearized version of the usual DDF formulation.

$$\begin{array}{lll} \max\limits_{\delta\in\mathbb{R},\boldsymbol{\sigma}\in\mathbb{R}^h} & M\delta - \sum\limits_{j=m+s+1}^{m+s+h} \sigma_j & \\ \text{subject to} & -\sum\limits_{j=1}^{m+s} w_j^*(z(j)+\delta g(j)) - \sum\limits_{j=m+s+1}^{m+s+h} w_j^*\sigma_j \geq \rho^* & \\ & \sigma_j \geq \mu & \forall j \in \{m+s+1, \ldots, m+s+h\} \\ & \sigma_j \geq \sum\limits_{k=1}^{m+s} [p_j(k)\,(z(k)+\delta g(k))] + q_j & \forall j \in \{m+s+1, \ldots, m+s+h\} \end{array} \tag{4}$$

9. We then calculate the uDEA-predicted outputs for $\mathbf{z}_i$ using $\hat{\mathbf{z}}_i = \mathbf{z}_i + \delta_{\text{uDEA}}(\mathbf{z}_i, \mathbf{g}_i)\mathbf{g}_i$.

10. Next, we evaluate the performance of each model by considering the mean squared error in prediction in each of the values of the dataset. As such, we calculate

$$\text{MSE}(\mu, \nu) = \sum_{\mathbf{z}_i \in \mathcal{Z}} \sum_k (\hat{z}_i(k) - z_i(k))^2$$

   and choose the hyperparameter pair (μ^*, ν^*) which minimizes this value.

   With the combination of hyperparameters above, we obtain the following MSE values, where we report only when a combination improves on the previous smallest error.

   ### Best err = 0.090269921408768 when $\nu = 0.001$, $\mu = -0.298312$.

   ### Best err = 0.08641468734755063 when $\nu = 0.001$, $\mu = -0.096951$.

   ### Best err = 0.07909658263683102 when $\nu = 0.001$ $\mu = -0.029831$.

   In this case, the chosen hyperparameters which minimise the MSE are $\mu^* = -0.0298312$ and $\nu^* = 0.001$.

11. Once we have selected the best μ^* and ν^*, we solve Program (3) with these values of μ^* and ν^* on the whole dataset $\mathcal{Z}$, obtaining the final values for $(\mathbf{w}^*, \xi^*, \rho^*)$.

This yields the following values for $(\mathbf{w}^*, \xi^*, \rho^*)$:
$\mathbf{w}^* = [0, 0, 0.005597, 0.005299, 0.012563, 0.005597, 0.012563, 0.020726, 0.020726, 0.005597, 0.027982, 0.006909, 0.005299, 0.012563, 0.006909, 0.012563, 0.005299, 0.005299, 0.038012, 0.005597, 0.020726, 0.005597]$,

$\boldsymbol{\xi}^* = [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]$,
and $\rho^* = 0.004512$.

In particular, we observe that every component of ξ is 0 and the technology envelops the dataset. These values will only become distinct from 0 in case there are some slight outliers which cannot be incorporated into the estimated technology.

The technology, defined by $\hat{T}_{\text{uDEA}} = \{\mathbf{z} \in \mathbb{R}^m_- \times \mathbb{R}^s_+ : \langle \mathbf{w}^* \cdot \boldsymbol{\phi}(\mathbf{z})\rangle \geq \rho^*\}$, is represented in Figure 6, together with the locations of the hyperplanes at the chosen offset value μ^*. It can be observed that the uDEA-estimated technology consists of a polyhedral set. The turning points of its boundary correspond to the regions where it intersects each offset hyperplane, with the corresponding $\mathbf{w}^*$ being a measure of how much the boundary turns at each such intersection.

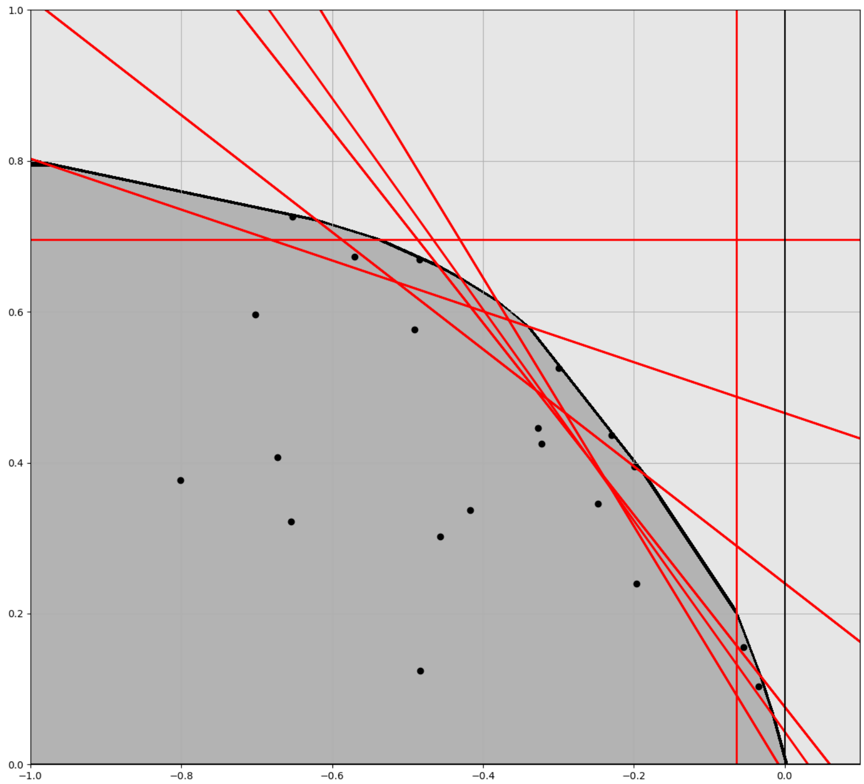

Figure 6: Technology estimated by uDEA (dark grey) with the corresponding hyperplanes.

We can also compare the estimated technology with the DEA-estimated technology in Figure 7. Furthermore, the estimated frontier values at each DMU, and the corresponding efficiencies, are given in Table 3. The Directional Distance Function of a DMU $\mathbf{z}_i$, $\delta_{\text{uDEA}}(\mathbf{z}_i)$ is calculated using Model (4), with $\mathbf{g_i} = (0, ..., 0, z_i(m+1), ..., z_i(m+s))$, and the Farrell output distance can be calculated via the relationship $\delta(\mathbf{z}_i) + 1$ or as the ratio between the efficient value $\hat{y}$ and the real value y of the DMU.

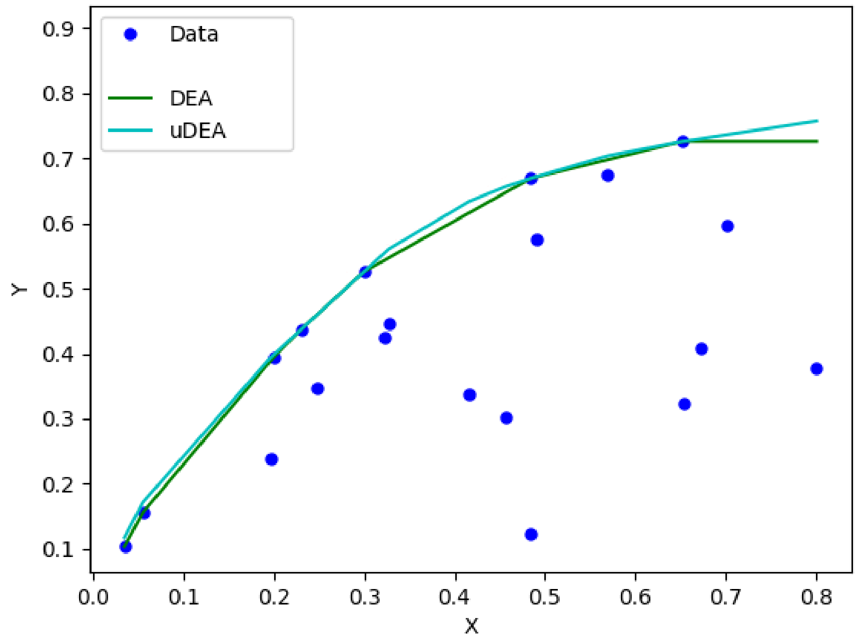

Figure 7: Comparison of uDEA and DEA efficient frontiers.

DMU	x	y	ŷ		Farrell output		DDF	
			DEA	uDEA	DEA	uDEA	DEA	uDEA
1	0.484505	0.669027	0.669027	0.669033	1.00	1.00	0.00	0.00
2	0.326993	0.445562	0.546832	0.560344	1.23	1.26	0.23	0.26
3	0.054679	0.155675	0.155675	0.171419	1.00	1.10	0.00	0.10
4	0.654055	0.322275	0.725845	0.726063	2.25	2.25	1.25	1.25
5	0.196588	0.239305	0.389355	0.394377	1.63	1.65	0.63	0.65
6	0.701487	0.59617	0.725845	0.736046	1.22	1.23	0.22	0.23
7	0.801368	0.376969	0.725845	0.757068	1.93	2.01	0.93	1.01
8	0.672302	0.407271	0.725845	0.729904	1.78	1.79	0.78	0.79
9	0.653045	0.725845	0.725845	0.72585	1.00	1.00	0.00	0.00
10	0.483559	0.123976	0.668293	0.668608	5.39	5.39	4.39	4.39
11	0.456264	0.302394	0.647118	0.656376	2.14	2.17	1.14	1.17
12	0.247688	0.345865	0.459407	0.459413	1.33	1.33	0.33	0.33
13	0.229938	0.436817	0.436817	0.436823	1.00	1.00	0.00	0.00
14	0.199769	0.394594	0.394594	0.398427	1.00	1.01	0.00	0.01
15	0.321987	0.425492	0.542949	0.553974	1.28	1.30	0.28	0.30
16	0.416869	0.337296	0.616556	0.634004	1.83	1.88	0.83	0.88
17	0.03427	0.103275	0.103275	0.117227	1.00	1.14	0.00	0.14
18	0.299813	0.525747	0.525747	0.525753	1.00	1.00	0.00	0.00
19	0.569931	0.673545	0.697826	0.703812	1.04	1.04	0.04	0.04
20	0.490719	0.576058	0.671122	0.671817	1.17	1.17	0.17	0.17

Table 3: DMU values, predicted outputs, and estimated efficiency with DEA and uDEA.