Operations Research Perspectives 11 (2023) 100284

Contents lists available at ScienceDirect

Operations Research Perspectives

journal homepage: www.elsevier.com/locate/orp

# An unsupervised learning-based generalization of Data Envelopment Analysis

Raul Moragues [a,1], Juan Aparicio [a,b,*], Miriam Esteve [a]

[a] Center of Operations Research (CIO). Miguel Hernandez University of Elche (UMH), 03202 Elche (Alicante), Spain
[b] Valencian Graduate School and Research Network of Artificial Intelligence (valgrAI), Valencia, Spain

ARTICLE INFO

*Keywords:*
Data Envelopment Analysis
Unsupervised machine learning
Support Vector Machines
Frontier analysis
Technical efficiency

ABSTRACT

In this paper, we introduce an unsupervised machine learning method for production frontier estimation. This new approach satisfies fundamental properties of microeconomics, such as convexity and free disposability (shape constraints). The new method generalizes Data Envelopment Analysis (DEA) through the adaptation of One-Class Support Vector Machines with piecewise linear transformation mapping. The new technique aims to reduce the overfitting problem occurring in DEA. How to measure technical inefficiency through the directional distance function is also introduced. Finally, we evaluate the performance of the new technique via a computational experience, showing that the mean squared error in the estimation of the frontier is up to 83% better than the standard DEA in certain scenarios.

## 1. Introduction

The measurement of efficiency of companies and public institutions has been and is a topic of interest for economists, production engineers and in various other areas of knowledge in the scientific literature [1–3]. Given a production function, which represents the maximum producible output from a given mix of inputs, it is possible to determine the technical efficiency of a company or institution as the distance between the vector of inputs and outputs, which mathematically represents the unit to evaluate, and the production function. Production functions must satisfy a series of properties which are clearly identified in the microeconomic literature of production theory [4]. These properties characterize the shape of the production function. For example, concavity is one of the usually assumed properties in the literature. Concavity of the production function assures that the nonnegative region of points below the production function in the space of inputs and outputs; that is, the so-called technology, is a convex set. Another traditional property is monotonicity of the production function; which indicates that the produced amount of outputs can never decrease when the amount of resources used in the production process increases. Since direct observation of a production function or technology is not feasible in practice, the above features of production functions have been considered as shape constraints in the literature when the estimation of the production functions from the observation of a sample of units is the aim. For example, in the deterministic case (without random noise), the definition of a production function as the maximum producible output forces the observed units to be always located below the estimated function in the space of inputs and outputs.

As can be expected when a problem attracts the attention of a considerable number of researchers, there have been many and very different techniques introduced in the literature with the purpose of estimating a production function from the observation of a data sample. It is usual in practice to subdivide these techniques into two large categories: parametric and nonparametric methodologies. Parametric techniques were the first to be introduced in the scientific literature, see for example the case of the Cobb–Douglas production function [5]. Under this approach, the production function is identified mathematically through an expression dependent on a set of coefficients to be estimated throughout some method associated with the minimization of an error function or maximization of a likelihood function. The application of statistical inference tools on the coefficients of interest and the measurement of technical efficiency of the units to be assessed is usually one of the key goals in this type of methodologies. The inferential procedures used require, in this case, the assumption of some distribution of the error term and the technical inefficiency term [6]. It is also often an approach where the treatment of multiple outputs is neither natural nor easy. In contrast, we have the nonparametric methodologies. These do not need an a priori identification of the mathematical expression of the production function to estimate nor do they require the assumption of any type of associated probability distribution function related to the data generation process. Among

[*] Corresponding author at: Center of Operations Research (CIO). Miguel Hernandez University of Elche (UMH), 03202 Elche (Alicante), Spain.
*E-mail addresses:* rmoragues@umh.es (R. Moragues), j.aparicio@umh.es (J. Aparicio), miriam.estevec@umh.es (M. Esteve).
[1] Ph.D. Program in Economics (DEcIDE).

https://doi.org/10.1016/j.orp.2023.100284
Received 19 January 2023; Received in revised form 27 April 2023; Accepted 13 June 2023
Available online 19 June 2023

these nonparametric techniques, Data Envelopment Analysis (DEA) [7,8] stands out as one of the most used tools in both applications as in methodological contributions in the last decades [9].

In particular, DEA relies on the construction of a technology in the space of inputs and outputs that satisfies certain classical properties of production theory (e.g., free disposability and convexity). It is a data driven approach with a lot of advantages from a benchmarking point of view and in which the treatment of the multi-output framework is relatively easy. However, Data Envelopment Analysis has been criticized for its deterministic and non-statistical nature, even being labeled as a pure descriptive tool of the data sample at a frontier level with little inferential power (its inferential power is exclusively based on the property of consistency and the increase of the sample size instead of on the fundamentals of the method) [10]. In fact, DEA suffers from an overfitting problem as a consequence of the application of the minimal extrapolation principle, which places the estimator of the production function as close to the dataset of observed points as possible [11]. In line with this, various authors have attempted to correct these deficiencies within the nonparametric approach, introducing complementary and alternative methodologies to DEA. For example, Simar and Wilson adapted the methodology known as bootstrapping to the determination of confidence intervals for the efficiency score obtained via DEA in [12,13].

Additionally, we can find in the literature a few recent contributions that try to relate in some way the field of machine learning to the efficiency measurement. In this regard, Kuosmanen and Johnson paid attention to piecewise linear estimators and introduced the Corrected Concave Nonparametric Least Squares (CCNLS) method [14]. Parmeter and Racine introduced nonparametric kernel frontier estimators [15]. Daouia et al. resorted to quadratic and cubic splines with shape constraints to estimate suitable production functions [16]. Esteve et al. tailored the Classification and Regression Trees (CART) method to determine production frontiers [10]. Tsionas introduced smooth monotone concave probabilistic regression trees for the estimation of efficiency and showed how to deal with panel data [17]. Valero-Carreras et al. adapted Support Vector Machines to determine technical efficiency [11,18]. Guillen et al. tailored Boosting to estimate production functions [19]. Guerrero et al. adapted the Structural Risk Minimization principle to determine production frontiers [20]. Finally, Olesen and Ruggiero proposed the use of hinging hyperplanes as a flexible nonparametric representation of a production function [21,22].

These are some of the attempts made in the last few decades to relate a nonparametric technique of a descriptive nature such as DEA with more advanced and machine learning methods assuming shape constraints. However, there is still a scarcity of contributions relating machine learning techniques and methods of estimation of production functions and technical efficiency, despite the nonparametric, data-driven nature of these techniques, and the tendency of many scientific areas towards the use of this type of tools for data analysis [23,24].

Every technique mentioned above, except those based on DEA (such as standard DEA itself or bootstrapped DEA), are regression methods which require the prior identification of a response variable and of one or more predictor variables. In a production context, the response variable is identified in every case with the production output, whereas the predictor variables are those that refer to the inputs used in the production process. As is usual with statistical regression methods, the extension of those models to the multi-response case is not easy and, in fact, given a certain technique, there are generally various alternatives found in the literature that are accepted as possible extensions for the treatment of multiple response variables [25]. In production theory, one possibility is to aggregate all outputs into a single economic type measure, such as the company revenue, if information about market prices of the various outputs are available. In that case, the economic type variable would be the response variable of the problem. Another possibility, within the parametric context, is to define a transformation function or a distance function from an expression which depends on every input and output simultaneously [26]. In machine learning, the a priori identification of one or various response variables is a necessary common practice within the subarea of supervised analysis.

Under supervised learning, the data are pairs of the response variable and the predictors, and the aim is to determine the functional relationship that relates the response variable to the predictors. A learning problem with a binary response variable is referred to as a classification problem, whereas for a real-valued response variable, the problem becomes known as a regression problem. In contrast, under unsupervised learning, there are no response variables and the objective is to gain some understanding of the Data Generating Process (DGP) that yielded the data (density estimation, clustering, etc.). According to this classification, DEA, as a data driven technique, resembles more an unsupervised methodology than a supervised approach, unlike most alternative techniques that exist in the literature. From this point of view, it could be more natural to grant DEA some inferential power through its assimilation via some unsupervised technique within the machine learning area. This corresponds to a methodological development which, as far as we are aware, is yet to be treated in the literature.

In recent production theory literature, Daraio and Simar describe the DGP which lies behind every productive process in [27]. It is assumed that we observe a (learning) sample of an identically and independently distributed random input–output vector with an unknown joint distribution with a certain support. In the production framework, the technology, i.e., everything that is feasible to be observed, coincides with the support of the DGP. And we are going to exploit this relationship in this paper.

As several authors have recently pointed out, unsupervised learning within the machine learning field includes the estimation of the support of a distribution [28]. It is often easier and more manageable to determine the support of the underlying probability density, that is, a function where (almost) all of the data lives in the region where this function takes nonzero values, than directly identifying the density function [28]. A related point of view is to see unsupervised learning as a classification problem where only examples from one of the two considered classes are available [29,30]. This point of view gave rise to the OneClassSVM algorithm [28], which we in particular adapt in this paper to address the problem of the estimation of production technologies through the identification of the support of the underlying DGP.

In particular, in this paper, we define the so-called unsupervised Data Envelopment Analysis (uDEA) model, which is a OneClassSVM-inspired model adapted to the setting of production frontier estimation through a piecewise linear transformation mapping. We prove that certain properties of the mapping are sufficient to guarantee convexity and free disposability of the estimated technology, and characterize its weak and strong frontiers. Then, we describe a DEA-based methodology to obtain the hyperplanes involved in the feature mapping. We then determine the dual of the quadratic optimization program associated with the uDEA approach. Additionally, we identify the hyperparameter that controls the proportion of support vectors and permitted outliers. Moreover, we adapt the directional distance function measure of technical inefficiency to our context, also focusing on Farrell's output-oriented measure. We then present some computational experiments where we compare uDEA with DEA.

The rest of the paper is organized as follows. Section 2 sets up some notation and briefly introduces the DEA approach and the OneClassSVM algorithm. In Section 3, we extend the OneClassSVM algorithm to the problem of estimating production frontiers, and describe each step of the new unsupervised DEA technique (uDEA), and prove that it satisfies some desired properties. We investigate the performance of uDEA in Section 4. Finally, we conclude and discuss possible extensions of the work in Section 5.

## 2. Background

In this section, we introduce the notation that we use throughout the paper, the main notions related to Data Envelopment Analysis, the One-Class Support Vector Machine algorithm, and the piecewise linear feature mapping that we will use.

### 2.1. Notation

Throughout this paper, we use the following notation. We denote variable names in lowercase boldface letters when they are vectors, and in Roman letters when they are scalars. For any integer $d \geq 1$, we denote by $\mathbb{R}^d$ the $d$-dimensional Euclidean space, and write $\mathbb{R}^d_+$ ($\mathbb{R}^d_-$) for its nonnegative (nonpositive) orthants. In general, given a vector $\mathbf{a}$, we denote its $j$'th component by $a_j$. However, it is standard notation that the points of a dataset are denoted by $\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_n \in \mathbb{R}^{m+s}$ where $m$ is the number of inputs and $s$ is the number of outputs. Hence, we indicate the component of these vectors using brackets, that is $\mathbf{z} = (z(1), z(2), \dots, z(m+s))$ and $\mathbf{z}_i = (z_i(1), z_i(2), \dots, z_i(m+s))$. Similarly, the coefficients of the hyperplanes that appear in our program are denoted by $\mathbf{p}_{m+s+1}, \dots, \mathbf{p}_{m+s+h}$, where $\mathbf{p}_k = (p_k(1), p_k(2), \dots, p_k(m+s))$. Bold notation $\mathbf{0}$ and $\mathbf{1}$ is used to denote the vectors of zeros and ones respectively, of the adequate dimension for the context in which they appear.

Given two vectors $\mathbf{a} = (a_1, \dots, a_d), \mathbf{b} = (b_1, \dots, b_d) \in \mathbb{R}^d$, we denote by $\langle \mathbf{a} \cdot \mathbf{b} \rangle$ their inner product, defined by $\langle \mathbf{a} \cdot \mathbf{b} \rangle = \sum_{i=1}^d a_i b_i$. The vector inequality $\mathbf{a} \geq \mathbf{b}$ ($\mathbf{a} > \mathbf{b}$) means that the specified inequality holds for every component, i.e., $a_i \geq b_i$ ($a_i > b_i$) for all $i = 1, \dots, d$. Note that $\mathbf{a} \geq \mathbf{0}$ and $\mathbf{a} \neq \mathbf{0}$, that is, the condition that at least one entry of $\mathbf{a}$ is nonzero, is not the same as $\mathbf{a} > \mathbf{0}$, which means that every component of $\mathbf{a}$ is strictly positive.

### 2.2. Data envelopment analysis

We begin with some general definitions about the estimation of production frontiers. The scenario in which we are interested is the following. We consider $n$ decision making units (DMUs), with $m \geq 1$ inputs and $s \geq 1$ outputs, $(\mathbf{x}_1, \mathbf{y}_1), \dots, (\mathbf{x}_n, \mathbf{y}_n) \in \mathbb{R}^{m+s}_+$ whose efficiency is to be evaluated. Each $\mathbf{x}_i$ represents the inputs used by DMU $i$ to obtain $\mathbf{y}_i$ outputs. In this paper, for convenience, we use the *netput* notation for each DMU, say DMU $i$ is $\mathbf{z}_i = (-\mathbf{x}_i, \mathbf{y}_i) \in \mathbb{R}^m_- \times \mathbb{R}^s_+$, see [31–33]. In this notation, the input coordinates take nonpositive values while the outputs take nonnegative values. We denote the netput dataset by $\mathcal{Z}$.

With this setup, the optimization objective is to maximize each of the components of $\mathbf{z}$ while staying in the region of feasible points, the so-called technology. This allows for a uniform treatment of every coordinate, regardless of whether it is an input or an output. In some sense, this uniform treatment and lack of a specialized or target variable is what gives DEA its unsupervised machine learning nature. Next, we introduce some key notions in production theory.

**Definition 2.1.** The *technology* or *production possibility set* is

$T := \{\mathbf{z} \in \mathbb{R}^m_- \times \mathbb{R}^s_+ : \mathbf{z} \text{ is feasible}\}.$

Let $T$ be a technology. Then the *Weak Efficient Frontier* of $T$ is

$\partial^W(T) := \{\mathbf{z} \in T : \hat{\mathbf{z}} > \mathbf{z} \Rightarrow \hat{\mathbf{z}} \notin T\}.$

The *Strong Efficient Frontier* of $T$ is

$\partial^S(T) := \{\mathbf{z} \in T : \hat{\mathbf{z}} \geq \mathbf{z},\ \hat{\mathbf{z}} \neq \mathbf{z} \Rightarrow \hat{\mathbf{z}} \notin T\}.$

Both types of frontiers represent reference sets that are usually utilized for measuring technical inefficiency, as the distance between a netput $\mathbf{z}$ and the corresponding frontier.

The strong efficient frontier is generally a subset of the weak efficient frontier, but they may coincide. At any point along the strong efficient frontier, any increase in any variable will result in leaving the technology, whereas along those points on the weak efficient frontier which are not on the strong efficient frontier, it is possible to increase some coordinate while the rest remain fixed and stay within the technology.

We refer to the theoretical technology as $T$, but as we will often work with estimates of the technology, we will denote the estimated technology as $\hat{T}$. There are two main families of approaches to estimating the frontier of a technology in the literature: parametric and nonparametric methods. In particular, the Data Envelopment Analysis (DEA) approach is a long-standing nonparametric family of techniques for estimating production frontiers, and was initiated by Farrell in [34] for the single output multi-input case, and later taken up and extended by [7,8]. We take some notation and concepts from [35]. DEA aims to determine the efficient frontier, and then calculate the efficiency of each DMU via some measure of distance to the frontier. In particular, we take the DEA estimated technology $\hat{T}_{\text{DEA}}$ that satisfies the following properties as introduced by [8].

1. Deterministicness: for all $\mathbf{z} \in \mathcal{Z}$, $\mathbf{z} \in \hat{T}_{\text{DEA}}$.
2. Convexity: if $\mathbf{z}_1, \dots, \mathbf{z}_k \in \hat{T}_{\text{DEA}}$ and $\lambda_j \geq 0$ for $j = 1, \dots, k$ with $\sum_{j=1}^k \lambda_j = 1$, then $\sum_{j=1}^k \lambda_j \mathbf{z}_j \in \hat{T}_{\text{DEA}}$.
3. Free disposability: for every $\mathbf{z} \in \hat{T}_{\text{DEA}}$, if $\mathbf{z}' \leq \mathbf{z}$, then $\mathbf{z}' \in \hat{T}_{\text{DEA}}$.[2]
4. Minimal extrapolation principle: $\hat{T}_{\text{DEA}}$ is the intersection set of all sets $\hat{T}$ satisfying Properties 1, 2 and 3.

By estimating the smallest set possible, DEA makes a cautious or conservative estimate of the technology and therefore also a prudent estimator of the loss due to technical inefficiency [36]. However, for the same reason, the obtained estimator suffers from overfitting and may not generalize very well. Our approach attempts to reduce this. In particular, the DEA estimate of the technology is the convex closure of the dataset extended to satisfy free disposability within the appropriate quadrant of signs for the inputs and outputs. We remark that this estimation gives rise to a piecewise linear boundary for the technology, which is the main motivation behind our choice of transformation function in the model.

Once the DEA has obtained an estimate of the technology and its frontier, it is possible to calculate an estimate of the technical efficiency of each DMU. There are many measures of efficiency available in the literature, and the most relevant ones for this paper are the radial functions [7,8,34], and the directional distance function [31,37,38].

**Definition 2.2.** Let $\mathbf{z} = (-\mathbf{x}, \mathbf{y}) \in \mathbb{R}^m_- \times \mathbb{R}^s_+$ be a DMU. The *technical efficiency* of $\mathbf{z}$ is the distance from $\mathbf{z}$ to $\partial^W(T)$. Some possibilities for measuring this distance are:

The *output-oriented radial measure* or (output-oriented) Farrell measure: $\lambda(\mathbf{z}) = \max\{\lambda : (-\mathbf{x}, \lambda\mathbf{y}) \in T\}$.

The *directional distance function* with respect to $\mathbf{g} \in \mathbb{R}^{m+s}_+$, with $\mathbf{g} \neq \mathbf{0}$: $\delta(\mathbf{z}, \mathbf{g}) = \max\{\delta : (\mathbf{z} + \delta\mathbf{g}) \in T\}$.

We remark that the Farrell measure is a particular case of the directional distance function where, for each $\mathbf{z}$, we take $\mathbf{g}(\mathbf{z}) = (0, \dots, 0, z(m+1), \dots, z(m+s))$.

### 2.3. OneClassSVM

Support Vector Machines (SVMs) are one of the most recognized and commonly used machine learning techniques. They are versatile, and have been adapted to almost every type of problem, from classification to regression (SVR), to unsupervised versions like One-Class Support Vector Machine (OneClassSVM). SVM was first introduced by Vapnik

[2] This is one of the advantages of using netputs. We do not need to identify two types of inequalities, one for inputs and one for outputs, in the description of free disposability.

in [39,40], and is theoretically grounded on solid statistical learning theory.

The idea behind SVM is to map the predictor space into a high-dimensional feature space and then construct an optimal hyperplane which, in the classification setting, separates the different classes. Instead of exclusively minimizing the empirical error, it aims to minimize the upper bound on the generalization error, which results in good prediction capability.

Furthermore, it is suitable for dealing with a limited number of samples, regardless of the number of feature variables, since a key point is that they are mapped to a high-dimensional space. As such, it is widely used in a variety of machine learning settings, and adaptations of the general method exist for regression, support estimation, pattern recognition, data mining, etc. The techniques based on SVMs also extend to the unsupervised setting, such as for outlier detection and support estimation. An important extension is the OneClassSVM, first introduced in [28], which we adapt in this paper for the context of production theory.

Standard OneClassSVM is a natural extension of the Support Vector Machine algorithm to the case of unlabeled data, following the point of view that an unlabeled dataset can be seen as a dataset where only examples from one of the two considered classes are available. In this sense, estimating the area where the class of available examples lives corresponds to estimating the support of the data generating process.

Given a dataset $\mathcal{Z} = \{\mathbf{z}_1, \dots, \mathbf{z}_n \in \mathbb{R}^N\}$, we let $\boldsymbol{\phi} : \mathbb{R}^N \to F$ be a feature map, that is, a transformation of the data into an inner product space $F$ such that the inner product in the image of $\boldsymbol{\phi}$ can be computed by evaluating some simple kernel $K$ given by an inner product via $K(\mathbf{z}, \mathbf{z}') = \langle \boldsymbol{\phi}(\mathbf{z}) \cdot \boldsymbol{\phi}(\mathbf{z}') \rangle$.

The decision function of the OneClassSVM algorithm is binary and returns 1 in the support of the dataset, and $-1$ everywhere else. It is of the form

$$\hat{f}(\mathbf{z}) = \operatorname{sgn}(\langle \mathbf{w} \cdot \boldsymbol{\phi}(\mathbf{z}) \rangle - \rho), \tag{1}$$

where the parameters $\mathbf{w} \in F$ and $\rho \in \mathbb{R}$ of the decision function are obtained by solving the quadratic program (2), which serves to separate the dataset from the origin in the transformed space $F$. The feature mapping $\boldsymbol{\phi}$ chosen will determine the shape of the efficient frontier, which is the boundary of the region where $\hat{f}(\mathbf{z})$ is positive.

$$\begin{array}{lll} \displaystyle\min_{\mathbf{w} \in F, \xi \in \mathbb{R}^n, \rho \in \mathbb{R}} & \displaystyle\frac{1}{2}\|\mathbf{w}\|^2 + \frac{1}{\nu n}\sum_{i=1}^{n} \xi_i - \rho & \\ \text{subject to} & \langle \mathbf{w} \cdot \boldsymbol{\phi}(\mathbf{z}_i) \rangle \geq \rho - \xi_i & \forall i \in \{1, \dots, n\} \\ & \xi_i \geq 0 & \forall i \in \{1, \dots, n\} \end{array} \tag{2}$$

In the quadratic program (2), $\nu \in (0, 1]$ is a hyperparameter which, by [28, Proposition 3], is an upper bound on the proportion of outliers and a lower bound on the number of support vectors in the solution. This is because $\nu$ controls how much we penalize the regularization term $\xi$, which will ensure the decision function will be positive on most of the points in the training set.

The hyperparameters appearing in our model, such as $\nu$, will be tuned via a 70:30 train-test split, where we train a model for each combination on $\mathcal{Z}_{train}$, which consists of 70% of the data, and we evaluate its performance $\mathcal{Z}_{test}$, which is the remaining 30% of the data and is used as a test set, choosing the combination of hyperparameters that works best on this unseen subset. We will use the standard mean squared error to measure the performance of our estimation.

In the literature, the bulk of the work is usually performed on the dual problem, which involves the kernel function $K$. In this paper, the dual will allow us to prove some results about the role of $\nu$ in the algorithm, but we will directly solve the primal problem.

### 2.3.1. Piecewise linear feature mapping

We now proceed to describe the feature map $\boldsymbol{\phi}$ which we will use. Since the boundaries for the technology estimated by DEA are piecewise linear, we focus on a piecewise linear feature mapping, which will result in a piecewise linear boundary. This formulation is named hinging hyperplanes in [41], and was considered in detail in [42], where a variety of such feature maps are described. The particular feature mapping that we choose to adapt is [42, (12)] and has the following form, where $m + s$ is the dimension of the data, and $h$ is the number of hyperplanes in the mapping, which is a hyperparameter of the algorithm:

$$\boldsymbol{\phi}(\mathbf{z}) = \begin{cases} z(k) & \text{for } k \in \{1, \dots, m+s\} \\ \max\{0, \ \langle \mathbf{p}_k \cdot \mathbf{z} \rangle + q_k\} & \text{for } k \in \{m+s+1, \dots, m+s+h\} \end{cases} \tag{3}$$

The idea behind this transformation is to split the coordinate space via the hyperplanes where $\langle \mathbf{p}_k \cdot \mathbf{z} \rangle + q_k = 0$, so that on the appropriate side of each hyperplane that component of the transformation is activated. As such, the boundary will consist of straight segments between the hyperplanes, where due to the appearance (or disappearance) of various components, the boundary will have a turning point, thus resulting in a piecewise linear boundary.

We remark that the value of 0 that we compare the hyperplane to within the maximum in the transformation is arbitrary, and changing it to a different value $\mu$ will correspond to replacing the turning points of the boundary from the hyperplane $\langle \mathbf{p}_k \cdot \mathbf{z} \rangle + q_k = 0$ to the hyperplane $\langle \mathbf{p}_k \cdot \mathbf{z} \rangle + q_k = \mu$, i.e., shifting the cutoff hyperplane by $\mu$. This will be useful for technical reasons with regard to the planes that we will choose in our transformation.

The mapping in question has $h(m + s + 1)$ parameters to define, as each component involves some $\mathbf{p}_k \in \mathbb{R}^{m+s}$ and $q_k \in \mathbb{R}$. Out of the various feature mappings defined in [42], we choose their formulation (12) as our kernel. This is because (11), though simpler and requiring only $h$ parameters, gives less flexibility to the boundary. The more complex transformation (13) involves $h(m + s)^2 + h$ parameters, which are computationally more expensive to tune. The following is the transformation (13) with $m + s$ hyperplanes at each coordinate

$$\boldsymbol{\phi}(\mathbf{z}) = \begin{cases} z(k) \\ \qquad \text{for } k \in \{1, \dots, m+s\} \\ \max\{0, \ \langle \mathbf{p}_{k1} \cdot \mathbf{z} \rangle + q_{k1}, \dots, \langle \mathbf{p}_{k(m+s)} \cdot \mathbf{z} \rangle + q_{k(m+s)}\} \\ \qquad \text{for } k \in \{m+s+1, \dots, m+s+h\} \end{cases} \tag{4}$$

We remark that [42, Theorem 2] proves that any piecewise linear set can be represented as the solution of a piecewise linear equation with the mapping defined by (13) in [42] when the number of hyperplanes in each component of $\boldsymbol{\phi}$ coincides with the dimension of $\mathbf{z}$.

## 3. Unsupervised data envelopment analysis model

In this section, we introduce the so-called unsupervised Data Envelopment Analysis (uDEA) model for estimating production frontiers, and prove some properties related to production theory. The base of the model which we adapt is the OneClassSVM algorithm with a piecewise linear kernel as described in Section 2.3, with some modifications to satisfy convexity and other required properties linked to efficiency measurement. The adapted OneClassSVM, i.e., the uDEA model, is defined as follows:

$$\min_{\mathbf{w} \in \mathbb{R}^{m+s+h}, \xi \in \mathbb{R}^n, \rho \in \mathbb{R}} \frac{1}{2}\|\mathbf{w}\|^2 + \frac{1}{\nu n}\sum_{i=1}^{n} \xi_i - \rho \tag{5}$$

$$\text{subject to } \langle \mathbf{w} \cdot \boldsymbol{\phi}(\mathbf{z}_i) \rangle \geq \rho - \xi_i \qquad \forall i \in \{1, \dots, n\} \tag{5a}$$

$$\xi_i \geq 0 \qquad \forall i \in \{1,\dots,n\} \tag{5b}$$

$$w_j \geq 0 \qquad \forall j \in \{1,\dots,m+s+h\} \tag{5c}$$

$$\langle \mathbf{w} \cdot \boldsymbol{\phi}(\mathbf{0}) \rangle = \rho \tag{5d}$$

Model (5) is a quadratic program where the objective function and restrictions (5a) and (5b) are identical to those of (2), whereas restriction (5c) will ensure convexity, and restriction (5d) will guarantee that the efficient frontier will pass through the origin, as we will prove later in this section. For the choice of feature mapping $\boldsymbol{\phi}$, we use the following adaptation of (3):

$$\boldsymbol{\phi}(\mathbf{z}) = \begin{cases} -z(k) & \text{for } k \in \{1,\dots,m+s\} \\ -\max\{\mu,\ \langle \mathbf{p}_k \cdot \mathbf{z}\rangle + q_k\} & \text{for } k \in \{m+s+1,\dots,m+s+h\} \end{cases} \tag{6}$$

The modifications to $\boldsymbol{\phi}$ are as follows. The 0 inside the maximum has been replaced by an offset hyperparameter $\mu$ that we will tune, and we also add a negative sign to the mapping. The sign change transforms the original convex, nondecreasing function into a concave, nonincreasing mapping, which are the necessary conditions to obtain a convex set satisfying free disposability, as we will show. Additionally, $\mathbf{p}_k$ and $q_k$ are hyperparameters of the model to be determined and are linked to the coefficients and offset, respectively, of different hyperplanes. Section 3.3 introduces a way of setting these hyperparameters.

We now prove some key properties of the chosen feature mapping, which will be useful for establishing certain features of the technology induced by Model (5).

**Lemma 3.1.** *For all* $k \in \{1,\dots,m+s+h\}$*,* $\phi_k(\mathbf{z})$ *is a concave function. Furthermore, if* $\mathbf{p}_k \geq \mathbf{0}$*, then whenever* $\mathbf{z}' \geq \mathbf{z}$ *we have* $\phi_k(\mathbf{z}') \leq \phi_k(\mathbf{z})$*.*

**Proof.** For $k \in \{1,\dots,m+s\}$, $\phi_k(\mathbf{z}) = -z(k)$ which is a nonincreasing linear function, hence concave.

For $k \in \{m+s+1,\dots,m+s+h\}$, $\phi_k(\mathbf{z}) = -\max\{\mu, \langle \mathbf{p}_k \cdot \mathbf{z}\rangle + q_k\}$. Both $\mu$ and $\langle \mathbf{p}_k \cdot \mathbf{z}\rangle + q_k$ are linear functions, hence convex, which means that the area above their curves is convex. Now the area above the curve of the max of two functions is the intersection of those of the two functions, so as the intersection of convex sets is convex, $\max\{\mu, \langle \mathbf{p}_k \cdot \mathbf{z}\rangle + q_k\}$ is convex, hence $\phi_k(\mathbf{z})$ is concave.

Furthermore, if $\mathbf{p}_k \geq \mathbf{0}$ then $\langle \mathbf{p}_k \cdot \mathbf{z}\rangle + q_k$ is nondecreasing. As $\mu$ is constant, we see that $\phi_k(\mathbf{z})$ is nonincreasing. Thus, $\phi_k(\mathbf{z})$ is nonincreasing for all $k \in \{1,\dots,m+s+h\}$, and so, whenever $\mathbf{z}' \geq \mathbf{z}$, we have $\phi_k(\mathbf{z}') \leq \phi_k(\mathbf{z})$. □

From now on, we will denote by $(\mathbf{w}^*, \xi^*, \rho^*)$ an optimal solution of Model (5). As with OneClassSVM, the above optimization problem defines a *decision function*

$$\hat{f}_{\text{uDEA}}(\mathbf{z}) = \operatorname{sgn}(\langle \mathbf{w}^* \cdot \boldsymbol{\phi}(\mathbf{z})\rangle - \rho^*)$$

which will be positive on most examples $\mathbf{z}_i$, while potentially leaving some outliers taking negative values. This will be regulated by the hyperparameter $\nu$ involved in Program (5), which in practice will be small. We will make this relationship precise in Lemma 3.9. We now define various basic notions related to Model (5).

**Definition 3.2.** The estimated *technology* defined by the optimization problem (5) is

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

Additionally, we define the frontier of $\hat{T}_{\text{uDEA}}$ as $F(\hat{T}_{\text{uDEA}}) = \{\mathbf{z} \in \mathbb{R}^m_- \times \mathbb{R}^s_+ : \langle \mathbf{w}^* \cdot \boldsymbol{\phi}(\mathbf{z})\rangle = \rho^*\}$, and its *weak frontier* as $F^W(\hat{T}_{\text{uDEA}}) = F(\hat{T}_{\text{uDEA}}) \cup \{\mathbf{z} \in \hat{T}_{\text{uDEA}} : z(k) = 0 \text{ for some } k \in \{1,\dots,m\}\}$.

Notice that $\mathbf{0} \in F(\hat{T}_{\text{uDEA}})$ by definition of the frontier and restriction (5d).

The weak frontier is useful in the case that the estimated technology does not live in the appropriate quadrant for the netput setting, $\mathbb{R}^m_- \times \mathbb{R}^s_+$, and includes a section along which some input coordinate is 0. Furthermore, if $F(\hat{T}_{\text{uDEA}}) \subseteq \mathbb{R}^m_- \times \mathbb{R}^s_+$, then $F^W(\hat{T}_{\text{uDEA}}) = F(\hat{T}_{\text{uDEA}})$. The name of weak frontier comes from the fact, which we will prove, that these extra added sections will be on the weak efficient frontier, $\partial^W(\hat{T}_{\text{uDEA}}) = \{\mathbf{z} \in \hat{T}_{\text{uDEA}} : \hat{\mathbf{z}} > \mathbf{z} \Rightarrow \hat{\mathbf{z}} \notin \hat{T}_{\text{uDEA}}\}$, but not on the strong efficient frontier $\partial^S(\hat{T}_{\text{uDEA}}) = \{\mathbf{z} \in \hat{T}_{\text{uDEA}} : \hat{\mathbf{z}} \geq \mathbf{z},\ \hat{\mathbf{z}} \neq \mathbf{z} \Rightarrow \hat{\mathbf{z}} \notin \hat{T}_{\text{uDEA}}\}$ of uDEA.

### 3.1. Properties

We now proceed to prove convexity and free disposability of $\hat{T}_{\text{uDEA}}$. In particular, convexity of $\hat{T}_{\text{uDEA}}$ will hold for any concave $\boldsymbol{\phi}$, while any nonincreasing $\boldsymbol{\phi}$ will give rise to an estimated technology satisfying free disposability.

**Proposition 3.3.** $\hat{T}_{\text{uDEA}}$ *is convex.*

**Proof.** Let $\mathbf{z}', \mathbf{z}^\dagger \in \hat{T}_{\text{uDEA}}$. We need to prove that, for any $\lambda \in [0,1]$, we have $\mathbf{z} = \lambda\mathbf{z}' + (1-\lambda)\mathbf{z}^\dagger \in \hat{T}_{\text{uDEA}}$. Since $\mathbf{z}', \mathbf{z}^\dagger \in \hat{T}_{\text{uDEA}}$, we have $\langle \mathbf{w}^* \cdot \boldsymbol{\phi}(\mathbf{z}')\rangle \geq \rho^*$ and $\langle \mathbf{w}^* \cdot \boldsymbol{\phi}(\mathbf{z}^\dagger)\rangle \geq \rho^*$. Let $\lambda \in [0,1]$. Then, since $\lambda \geq 0$ and $1-\lambda \geq 0$, we observe that $\mathbf{z} = \lambda\mathbf{z}' + (1-\lambda)\mathbf{z}^\dagger \in \mathbb{R}^m_- \times \mathbb{R}^s_+$.

By Lemma 3.1, $\phi_k(\mathbf{z})$ is a concave function for all $k \in \{1,\dots,m+s+h\}$, hence so is $\boldsymbol{\phi}(\mathbf{z})$. Thus, we have $\boldsymbol{\phi}(\lambda\mathbf{z}'+(1-\lambda)\mathbf{z}^\dagger) \geq \lambda\boldsymbol{\phi}(\mathbf{z}')+(1-\lambda)\boldsymbol{\phi}(\mathbf{z}^\dagger)$. Furthermore, as $\mathbf{w}^* \geq \mathbf{0}$ by constraint (5c), we have

$$\langle \mathbf{w}^* \cdot \boldsymbol{\phi}(\mathbf{z})\rangle = \langle \mathbf{w}^* \cdot \boldsymbol{\phi}(\lambda\mathbf{z}'+(1-\lambda)\mathbf{z}^\dagger)\rangle \geq \lambda\langle \mathbf{w}^* \cdot \boldsymbol{\phi}(\mathbf{z}')\rangle + (1-\lambda)\langle \mathbf{w}^* \cdot \boldsymbol{\phi}(\mathbf{z}^\dagger)\rangle \geq \rho^*.$$

In other words, $\mathbf{z} \in \hat{T}_{\text{uDEA}}$. Therefore, $\hat{T}_{\text{uDEA}}$ is convex. □

We remark that convexity of $\hat{T}_{\text{uDEA}}$ relies only on concavity of $\boldsymbol{\phi}$ and $\mathbf{w}^* \geq \mathbf{0}$. For free disposability, the additional assumption that $\mathbf{p}_k \geq \mathbf{0}$ is required.

**Proposition 3.4.** *Suppose that* $\mathbf{p}_k \geq \mathbf{0}$ *for all* $k \in \{m+s+1,\dots,m+s+h\}$*. Then* $\hat{T}_{\text{uDEA}}$ *satisfies free disposability. In other words, if* $\mathbf{z} \in \hat{T}_{\text{uDEA}}$ *and* $\mathbf{z}' \leq \mathbf{z}$ *with* $\mathbf{z}' \in \mathbb{R}^m_- \times \mathbb{R}^s_+$ *then* $\mathbf{z}' \in \hat{T}_{\text{uDEA}}$*.*

**Proof.** Assume $\mathbf{z} \in \hat{T}_{\text{uDEA}}$ and let $\mathbf{z}' \leq \mathbf{z}$. Then, we have $\langle \mathbf{w}^* \cdot \boldsymbol{\phi}(\mathbf{z})\rangle \geq \rho^*$. As $\mathbf{p}_k \geq \mathbf{0}$, Lemma 3.1 implies that $\phi_k(\mathbf{z}') \geq \phi_k(\mathbf{z})$ for all $k \in \{1,\dots,m+s+h\}$. Furthermore, since $\mathbf{w}^* \geq \mathbf{0}$, we have $w^*_k\phi_k(\mathbf{z}') \geq w^*_k\phi_k(\mathbf{z})$ for all $k \in \{1,\dots,m+s+h\}$. Therefore, $\langle \mathbf{w}^* \cdot \boldsymbol{\phi}(\mathbf{z}')\rangle \geq \langle \mathbf{w}^* \cdot \boldsymbol{\phi}(\mathbf{z})\rangle \geq \rho^*$ and, as $\mathbf{z}' \in \mathbb{R}^m_- \times \mathbb{R}^s_+$, we have proved that $\mathbf{z}' \in \hat{T}_{\text{uDEA}}$ as claimed. Thus, $\hat{T}_{\text{uDEA}}$ satisfies free disposability. □

### 3.2. The production frontier

We now compare the sets $F(\hat{T}_{\text{uDEA}})$ and $F^W(\hat{T}_{\text{uDEA}})$ with $\partial^W(\hat{T}_{\text{uDEA}})$ and $\partial^S(\hat{T}_{\text{uDEA}})$. The weak and strong frontiers were introduced in Definition 2.1. We remark that since $\partial^W(\hat{T}_{\text{uDEA}}) \subseteq \hat{T}_{\text{uDEA}}$, the coordinates of $\mathbf{z} \in \partial^W(\hat{T}_{\text{uDEA}})$ already have the appropriate signs for netputs due to the definition of $\hat{T}_{\text{uDEA}}$, and similarly for $\partial^S(\hat{T}_{\text{uDEA}})$. The two frontiers, however, behave differently along points where some input is 0, where it may be the case that some such areas are in $\partial^W(\hat{T}_{\text{uDEA}}) \setminus \partial^S(\hat{T}_{\text{uDEA}})$.

In order to state the next result, we define the following function which selects the indices where the maximum term in the feature mapping attains its cutoff value of $-\mu$. Let $\mathbf{z}, \hat{\mathbf{z}} \in \mathbb{R}^{m+s}$. Then

$$I_1(\mathbf{z}, \hat{\mathbf{z}}) = \{k \in \{m+s+1,\dots,m+s+h\} : \phi_k(\mathbf{z}) = \phi_k(\hat{\mathbf{z}}) = -\mu\}.$$

Then, the relationship between the weak frontier $F^W(\hat{T}_{\text{uDEA}})$ and the weak efficient frontier $\partial^W(\hat{T}_{\text{uDEA}})$ is as follows:

**Proposition 3.5.** *Assume that $\mathbf{p}_k \geq \mathbf{0}$ with $\mathbf{p}_k \neq \mathbf{0}$ for all $k \in \{m+s+1,\dots,m+s+h\}$. Then either $\partial^W(\hat{T}_{\mathrm{uDEA}}) = F^W(\hat{T}_{\mathrm{uDEA}})$ or $F(\hat{T}_{\mathrm{uDEA}}) = \hat{T}_{\mathrm{uDEA}}$. The latter case can only happen when there is $\mathbf{z} \in F^W(\hat{T}_{\mathrm{uDEA}})$ and $\hat{\mathbf{z}} > \mathbf{z}$ such that $\rho^* = -\mu \sum_{k\in I_1(\mathbf{z},\hat{\mathbf{z}})} w^*_k$ and $w^*_k = 0$ for all $k \notin I_1(\mathbf{z},\hat{\mathbf{z}})$.*

**Proof.** We begin by proving that $\partial^W(\hat{T}_{\mathrm{uDEA}}) \subseteq F^W(\hat{T}_{\mathrm{uDEA}})$. Let $\mathbf{z} \in \hat{T}_{\mathrm{uDEA}} \setminus F^W(\hat{T}_{\mathrm{uDEA}})$. Then, $\langle \mathbf{w}^* \cdot \boldsymbol{\phi}(\mathbf{z})\rangle = \rho^* + \varepsilon > \rho^*$ for some $\varepsilon > 0$, and $z(k) < 0$ for all $k \in \{1,\dots,m\}$. Note that $\langle \mathbf{w}^* \cdot \boldsymbol{\phi}(\mathbf{z})\rangle$ is a continuous function. Then, by definition of continuity, there exists $r > 0$ such that the ball $B_r(\mathbf{z})$ of radius $r > 0$ centered at $\mathbf{z}$ satisfies that for every $\mathbf{z}^* \in B_r(\mathbf{z})$ we have $\langle \mathbf{w}^* \cdot \boldsymbol{\phi}(\mathbf{z}^*)\rangle > \rho^*$. Since $z(k) < 0$ for all $k \in \{1,\dots,m\}$, we have $B_r(\mathbf{z}) \cap \{\mathbf{t} \in \mathbb{R}^m_- \times \mathbb{R}^s_+ : \mathbf{t} > \mathbf{z}\} \neq \emptyset$.

Let $\mathbf{z}^* \in B_r(\mathbf{z}) \cap \{\mathbf{t} \in \mathbb{R}^m_- \times \mathbb{R}^s_+ : \mathbf{t} > \mathbf{z}\}$, then $\mathbf{z}^*$ satisfies $\mathbf{z}^* > \mathbf{z}$ with $\langle \mathbf{w}^* \cdot \boldsymbol{\phi}(\mathbf{z}^*)\rangle > \rho^*$, that is $\mathbf{z}^* \in \hat{T}_{\mathrm{uDEA}}$, which proves that $\mathbf{z} \notin \partial^W(\hat{T}_{\mathrm{uDEA}})$. Therefore, $\partial^W(\hat{T}_{\mathrm{uDEA}}) \subseteq F^W(\hat{T}_{\mathrm{uDEA}})$. Note that this holds regardless of the value of $\rho^*$ and $\mathbf{w}^*$.

Now, we consider the inclusion $F^W(\hat{T}_{\mathrm{uDEA}}) \subseteq \partial^W(\hat{T}_{\mathrm{uDEA}})$. Let $\mathbf{z} \in F^W(\hat{T}_{\mathrm{uDEA}})$, then either $\langle \mathbf{w}^* \cdot \boldsymbol{\phi}(\mathbf{z})\rangle = \rho^*$ or $\mathbf{z} \in \hat{T}_{\mathrm{uDEA}}$ with $z(k) = 0$ for some $k = \{1,\dots,m\}$. In the second case, $\hat{\mathbf{z}} > \mathbf{z}$ implies that $\hat{z}(k) > 0$ for some $k = \{1,\dots,m\}$, so $\hat{\mathbf{z}} \notin \hat{T}_{\mathrm{uDEA}}$ as required.

If $\langle \mathbf{w}^* \cdot \boldsymbol{\phi}(\mathbf{z})\rangle = \rho^*$, we suppose that $\hat{\mathbf{z}}$ satisfies $\hat{\mathbf{z}} > \mathbf{z}$. Then, we need to prove that $\langle \mathbf{w}^* \cdot \boldsymbol{\phi}(\hat{\mathbf{z}})\rangle < \rho^*$, that is $\hat{\mathbf{z}} \notin \hat{T}_{\mathrm{uDEA}}$. By Lemma 3.1, we have $\phi_k(\hat{\mathbf{z}}) \leq \phi_k(\mathbf{z})$ for all $k \in \{1,\dots,m+s+h\}$ so, as $\mathbf{w}^* \geq \mathbf{0}$, we have $w^*_k\phi_k(\hat{\mathbf{z}}) \leq w^*_k\phi_k(\mathbf{z})$. Then, $\langle \mathbf{w}^* \cdot \boldsymbol{\phi}(\hat{\mathbf{z}})\rangle \leq \langle \mathbf{w}^* \cdot \boldsymbol{\phi}(\mathbf{z})\rangle$. Assume that $\langle \mathbf{w}^* \cdot \boldsymbol{\phi}(\hat{\mathbf{z}})\rangle = \langle \mathbf{w}^* \cdot \boldsymbol{\phi}(\mathbf{z})\rangle$. Then, as $w^*_k\phi_k(\hat{\mathbf{z}}) \leq w^*_k\phi_k(\mathbf{z})$ for all $k$, we must have $w^*_k\phi_k(\hat{\mathbf{z}}) = w^*_k\phi_k(\mathbf{z})$ for all $k \in \{1,\dots,m+s+h\}$.

For $k \in \{1,\dots,m+s\}$, we have $\phi_k(\mathbf{z}) = -z(k)$ so, as $\hat{\mathbf{z}} > \mathbf{z}$, we have $\phi_k(\hat{\mathbf{z}}) < \phi_k(\mathbf{z})$. Thus, in order to have $w^*_k\phi_k(\hat{\mathbf{z}}) = w^*_k\phi_k(\mathbf{z})$, we must have $w^*_k = 0$.

For $k \in \{m+s+1,\dots,m+s+h\}$, $\phi_k(\mathbf{z}) = -\max\{\mu, \langle \mathbf{p}_k \cdot \mathbf{z}\rangle + q_k\} \leq -\mu$, and we have different cases.

- If $\phi_k(\hat{\mathbf{z}}) = -\mu$ then, as $\phi_k(\hat{\mathbf{z}}) \leq \phi_k(\mathbf{z}) \leq -\mu$, we have $\phi_k(\mathbf{z}) = -\mu$. Then $k \in I_1(\mathbf{z},\hat{\mathbf{z}})$ and equality is possible.
- If $\phi_k(\hat{\mathbf{z}}) \neq -\mu$ but $\phi_k(\mathbf{z}) = -\mu$ then we must have $w^*_k = 0$.
- Finally, if $\phi_k(\hat{\mathbf{z}})$ and $\phi_k(\mathbf{z})$ are both strictly smaller than $-\mu$ then $\phi_k(\hat{\mathbf{z}}) = -\langle \mathbf{p}_k \cdot \hat{\mathbf{z}}\rangle - q_k \leq -\langle \mathbf{p}_k \cdot \mathbf{z}\rangle - q_k = \phi_k(\mathbf{z})$. Then, as $\hat{\mathbf{z}} > \mathbf{z}$, in order to have $w^*_k\phi_k(\hat{\mathbf{z}}) = w^*_k\phi_k(\mathbf{z})$ we need that either $\mathbf{p}_k = \mathbf{0}$ or $w^*_k = 0$. As we assume that the first does not happen, we must have $w^*_k = 0$.

Then, as $\langle \mathbf{w}^* \cdot \boldsymbol{\phi}(\hat{\mathbf{z}})\rangle = \langle \mathbf{w}^* \cdot \boldsymbol{\phi}(\mathbf{z})\rangle = \rho^*$, we must have

$$\begin{aligned}\rho^* = \langle \mathbf{w}^* \cdot \boldsymbol{\phi}(\mathbf{z})\rangle &= -\sum_{k=1}^{m+s} 0 \cdot z(k) - \sum_{k\in I_1(\mathbf{z},\hat{\mathbf{z}})} w^*_k \cdot \max\{\mu,\mu\} \\ &\quad - \sum_{k\notin I_1(\mathbf{z},\hat{\mathbf{z}})} 0 \cdot \max\{\mu, \langle \mathbf{p}_k \cdot \mathbf{z}\rangle + q_k\} \\ &= -\sum_{k\in I_1(\mathbf{z},\hat{\mathbf{z}})} w^*_k \mu.\end{aligned} \tag{7}$$

Thus, whenever $\rho^* \neq -\mu\sum_{k\in I_1(\mathbf{z},\hat{\mathbf{z}})} w^*_k$ or $w^*_k \neq 0$ for some $k \notin I_1(\mathbf{z},\hat{\mathbf{z}})$, we have $\langle \mathbf{w}^* \cdot \boldsymbol{\phi}(\hat{\mathbf{z}})\rangle < \langle \mathbf{w}^* \cdot \boldsymbol{\phi}(\mathbf{z})\rangle = \rho^*$, that is $\hat{\mathbf{z}} \notin \hat{T}_{\mathrm{uDEA}}$, hence proving that $F^W(\hat{T}_{\mathrm{uDEA}}) \subseteq \partial^W(\hat{T}_{\mathrm{uDEA}})$.

Now, assume that $\rho^* = -\mu\sum_{k\in I_1(\mathbf{z},\hat{\mathbf{z}})} w^*_k$ and $w^*_k = 0$ for all $k \notin I_1(\mathbf{z},\hat{\mathbf{z}})$. Then, for all $\mathbf{z}' \in \mathbb{R}^{m+s}$, we have $\langle \mathbf{w}^* \cdot \boldsymbol{\phi}(\mathbf{z}')\rangle = -\sum_{k\in I_1(\mathbf{z},\hat{\mathbf{z}})} w^*_k \cdot \max\{\mu, \langle \mathbf{p}_k \cdot \mathbf{z}'\rangle + q_k\} \leq \rho^*$, so $\hat{T}_{\mathrm{uDEA}} = F(\hat{T}_{\mathrm{uDEA}})$. □

The relationship with the strong efficient frontier of $\hat{T}_{\mathrm{uDEA}}$ is the following.

**Proposition 3.6.** *Assume that $\mathbf{p}_k > \mathbf{0}$. If $\partial^W(\hat{T}_{\mathrm{uDEA}}) = F^W(\hat{T}_{\mathrm{uDEA}})$ then $\partial^S(\hat{T}_{\mathrm{uDEA}}) = F(\hat{T}_{\mathrm{uDEA}})$.*

**Proof.** We will prove that $\partial^W(\hat{T}_{\mathrm{uDEA}}) \setminus \partial^S(\hat{T}_{\mathrm{uDEA}}) = \{\mathbf{z} \in \hat{T}_{\mathrm{uDEA}} : z(k) = 0 \text{ for some } k \in \{1,\dots,m\}\}$. The lemma follows from this and Proposition 3.5. Note that, by definition, we have $\partial^S(\hat{T}_{\mathrm{uDEA}}) \subseteq \partial^W(\hat{T}_{\mathrm{uDEA}})$.

Let $\mathbf{z} \in \partial^W(\hat{T}_{\mathrm{uDEA}}) \setminus \partial^S(\hat{T}_{\mathrm{uDEA}})$. Then, there exists some $\hat{\mathbf{z}} \in \hat{T}_{\mathrm{uDEA}}$ such that $\hat{\mathbf{z}} \geq \mathbf{z}$ with $\hat{\mathbf{z}} \neq \mathbf{z}$ but $\hat{\mathbf{z}} \not> \mathbf{z}$. In other words, there exist $i,j \in \{1,\dots m+s\}$ such that $\hat{z}(i) = z(i)$ but $\hat{z}(j) > z(j)$. Pick such a $\hat{\mathbf{z}}$.

Now, since $\partial^W(\hat{T}_{\mathrm{uDEA}}) = F^W(\hat{T}_{\mathrm{uDEA}})$, we have either $\langle \mathbf{w}^* \cdot \boldsymbol{\phi}(\mathbf{z})\rangle = \rho^*$, or $z(k') = 0$ for some $k' \in \{1,\dots,m\}$. In the former case, by Lemma 3.1 we have $\phi_k(\hat{\mathbf{z}}) \leq \phi_k(\mathbf{z})$ for all $k \in \{1,\dots,m+s+h\}$, and so, since $w^*_k \geq 0$, it follows that $w^*_k\phi_k(\hat{\mathbf{z}}) \leq w^*_k\phi_k(\mathbf{z})$ for all $k \in \{1,\dots,m+s+h\}$, hence $\boldsymbol{\phi}(\hat{\mathbf{z}}) \leq \boldsymbol{\phi}(\mathbf{z})$, with equality only if every component is equal or the corresponding $w^*_k = 0$.

In order to have equality, we must have $-\max\{\mu, \langle \mathbf{p}_k \cdot \hat{\mathbf{z}}\rangle + q_k\} = -\max\{\mu, \langle \mathbf{p}_k \cdot \mathbf{z}\rangle + q_k\}$ for all $k$, so that whenever $\langle \mathbf{p}_k \cdot \mathbf{z}\rangle + q_k$ is greater than $\mu$ we must have $\sum_{j=1}^{m+s} p_k(j)\hat{z}(j) = \langle \mathbf{p}_k \cdot \hat{\mathbf{z}}\rangle = \langle \mathbf{p}_k \cdot \mathbf{z}\rangle = \sum_{j=1}^{m+s} p_k(j)z(j)$. By assumption, $p_k(j) > 0$ for all $k,j$, so we must have $z(j) = \hat{z}(j)$ for all $j$, contradicting our assumption that $\mathbf{z} \in \partial^W(\hat{T}_{\mathrm{uDEA}}) \setminus \partial^S(\hat{T}_{\mathrm{uDEA}})$.

Thus, if $\mathbf{z} \in \partial^W(\hat{T}_{\mathrm{uDEA}}) \setminus \partial^S(\hat{T}_{\mathrm{uDEA}})$ then $z(k') = 0$ for some $k' \in \{1,\dots,m\}$, and $\partial^S(\hat{T}_{\mathrm{uDEA}}) = F(\hat{T}_{\mathrm{uDEA}})$. □

Regarding the hypothesis in Proposition 3.6, we remark that it is not necessary that $p_k(j) > 0$ for all $k$ and $j$, it is enough to obtain inequality along some component along which we remain in the estimated technology.

### *3.3. Choosing the parameters for the piecewise linear transformation.*

We now describe how we choose the parameters $\mathbf{p}_k$ and $q_k$ of the transformation mapping $\boldsymbol{\phi}$ which define the hyperplanes. We will also, along the way, determine an appropriate value of $h$, and introduce $\mu$. As discussed in Section 2.3.1, each of the components $\phi_k(\mathbf{z})$ involving a hyperplane will get activated whenever $\mathbf{z}$ is on the appropriate side of the hyperplane, and attain the value $-\mu$ elsewhere.

As such, the parameters in piecewise linear feature mappings determine the hyperplanes along which the boundary of the estimated technology changes its direction, and the SVM optimization technique then finds the best parameters to minimize the distance from the dataset to the boundary. As a consequence, the results obtained by uDEA will heavily depend on which planes are involved in the feature mapping. Since the hyperplanes determine where the boundary is allowed turning points, we observe that these hyperplanes should be located in the region between the observed DMUs and the theoretical boundary, which will typically be unknown, but will determine a slightly larger region than the convex hull of the dataset $\mathcal{Z}$, which is the region determined by DEA.

Since DEA satisfies the minimal extrapolation principle, it will yield an estimated technology which is contained in the theoretical technology, and the hyperplanes which define the DEA boundary will be appropriate turning points for our estimator of the technology.

DEA determines the convex hull of the points, and as such assigns to each DMU $\mathbf{z}_i$ a supporting hyperplane $(\mathbf{p}_k, q_k)$, and these will be used as a basis for our parameters.

We use the directional distance function program in its multiplier form, which can be found in [43, Program 3]. We solve the following program $n$ times, one for each $\mathbf{z}_i \in \mathcal{Z}$. We choose the directional function $\mathbf{g} = \mathbf{1}$, which corresponds to the Chebyshev norm or $l_\infty$ norm as introduced in [44].

When adapted to the netput setting, the linear program to solve for DMU $\mathbf{z}_i$ is:

$$\begin{array}{lll} \min\limits_{\mathbf{p}_k,q_k} & -\langle \mathbf{p}_k \cdot \mathbf{z}_i\rangle - q_k = -\mu_i & \\ \text{subject to} & \langle \mathbf{p}_k \cdot \mathbf{z}_r\rangle + q_k \leq 0 & \forall r \in \{1,\dots,n\} \\ & \langle \mathbf{p}_k \cdot \mathbf{1}\rangle = 1 & \\ & \mathbf{p}_k \geq \mathbf{0} & \end{array} \tag{8}$$

The solution to this problem gives us the parameters $\mathbf{p}_k$ and $q_k$ of a supporting hyperplane for the technology estimated by DEA, which is the convex hull of $\mathcal{Z}$ plus the region generated by free disposability. Due to our notation, $k = m+s+i$, so that the planes associated with each

**Table 1**
Correspondence between dual variables and primal restrictions.

| Dual variables | Primal restrictions |
|---|---|
| $\alpha_i$ | $\langle \mathbf{w} \cdot \boldsymbol{\phi}(\mathbf{z}_i)\rangle \geq \rho - \xi_i$ |
| $\beta_i = 1/\nu n - \alpha_i$ | $\xi_i \geq 0$ |
| $\gamma_j$ | $w_j \geq 0$ |
| $\alpha_0$ | $\langle \mathbf{w} \cdot \boldsymbol{\phi}(\mathbf{0})\rangle = \rho$ |

DMU appear in the same order in the transformation $\boldsymbol{\phi}$. The hyperplane thus defined is the region of the convex closure $\hat{T}_{\text{DEA}}$ to which the DMU $\mathbf{z}_i$ is projected along the directional vector $\mathbf{g}$. As such, each hyperplane will contain every DMU in the region where $\langle \mathbf{p}_k \cdot \mathbf{z}_i\rangle + q_k \leq 0$ and pass through at least one DMU. Furthermore, since the values of $\mathbf{p}_k$ are nonnegative and add up to one, the coefficients of the hyperplanes are normalized.

With this setup, we obtain $n$ hyperplanes, so that $h = n$, that is, the number of hyperplanes in $\boldsymbol{\phi}$ is the number of DMUs in the dataset. The hyperplanes may be repeated, since more than one DMU can project to the same region of the convex closure. Furthermore, for each DMU $\mathbf{z}_i$, we obtain a measure $\mu_i = \langle \mathbf{p}_k \cdot \mathbf{z}_i\rangle + q_k \leq 0$ of the inefficiency of $\mathbf{z}_i$, which is the term $\phi_k(\mathbf{z}_i)$ in the corresponding restriction (5a). As such, we obtain that the value of $\phi_k(\mathbf{z}_i) = -\max\{\mu, \langle \mathbf{p}_k \cdot \mathbf{z}_i\rangle + q_k\} = -\max\{\mu, \mu_i\}$ depends on the value of $\mu$. Therefore, we observe that $\mu$ acts as a cutoff parameter which forces $\phi_k(\mathbf{z}_i) = -\mu$ whenever $\mu_i = \langle \mathbf{p}_k \cdot \mathbf{z}_i\rangle + q_k \leq \mu$ and, as such, gives a constant value to all components that arise from DMUs that are further from the DEA estimated efficient frontier than the cutoff distance $\mu$. We define $\mu_{\min} := \min_i\{\mu_i\}$, then an appropriate interval of possible values for the hyperparameter $\mu$ is $[\mu_{\min}, 0)$.

In particular, the closer that $\mu$ is to 0, the fewer terms that will appear in $\boldsymbol{\phi}$ with values different to $-\mu$, and when $\mu = \min_i\{\mu_i\}$ then every single point will give rise to a plane which sometimes takes values larger than $-\mu$. Furthermore, as $\mu$ changes values, the cutoff region where each hyperplane is activated will be translated by that amount, slightly changing the region where the boundary changes direction.

### *3.4. Lagrangian, dualization, and outlier control*

In order to prove some results about when DMUs are on the efficient frontier, outliers, or support vectors, as well as the role of the hyperparameter $\nu$, we will use the dual problem of (5) as calculated in the Appendix, which is the following:

$$\begin{aligned} \min_{\boldsymbol{\alpha},\boldsymbol{\gamma},\alpha_0} \quad & \frac{1}{2}\,\|\mathbf{w}\|^2 \\ \text{subject to} \quad & 0 \leq \alpha_i \leq 1/\nu n \qquad \text{for } i \in \{1,\dots,n\}, \\ & \sum_{i=1}^{n} \alpha_i + \alpha_0 = 1, \\ & \boldsymbol{\gamma} \geq \mathbf{0}, \end{aligned} \tag{9}$$

where $\mathbf{w} = \sum_{i=1}^{n} \alpha_i \boldsymbol{\phi}(\mathbf{z}_i) + \boldsymbol{\gamma} + \alpha_0 \boldsymbol{\phi}(\mathbf{0})$ (see Appendix, in particular (A.1), for the details).

The correspondence between variables of the dual and restrictions of the primal problem appears in Table 1.

We now consider what information about solutions of the primal problem is given by the dual. Since we are working with a standard quadratic program, the Karush–Kuhn–Tucker (KKT) conditions are both necessary and sufficient conditions to check that a solution is optimal for the problem. In particular, complementary slackness holds. This allows us to prove the following, where we assume that $(\mathbf{w}^*, \boldsymbol{\xi}^*, \rho^*, \boldsymbol{\alpha}^*, \boldsymbol{\beta}^*, \boldsymbol{\gamma}^*, \alpha_0^*)$ is a solution of the KKT conditions.

**Proposition 3.7.** *The following hold for each* $i \in \{1,\dots,n\}$*:*

*(1) If* $0 < \alpha_i^* < 1/\nu n$ *then* $\mathbf{z}_i \in F(\hat{T}_{\text{uDEA}})$.

*(2) If* $\mathbf{z}_i \in \hat{T}_{\text{uDEA}} \setminus F(\hat{T}_{\text{uDEA}})$ *then* $\alpha_i^* = 0$.

*(3)* $\mathbf{z}_i \in \hat{T}_{\text{uDEA}}$ *if and only if* $\xi_i^* = 0$.

**Proof.** Let $(\mathbf{w}^*, \boldsymbol{\xi}^*, \rho^*, \boldsymbol{\alpha}^*, \boldsymbol{\beta}^*, \boldsymbol{\gamma}^*, \alpha_0^*)$ be a solution of the KKT conditions. Then, the following hold for all $i \in \{1,\dots,n\}$ and all $j \in \{1,\dots,m+s+h\}$:

**(KKT1)** Either $\alpha_i^* = 0$ or $\langle \mathbf{w}^* \cdot \boldsymbol{\phi}(\mathbf{z}_i)\rangle = \rho^* - \xi_i^*$

**(KKT2)** Either $\beta_i^* = 1/\nu n - \alpha_i^* = 0$ or $\xi_i^* = 0$

**(KKT3)** Either $\gamma_j^* = 0$ or $w_j^* = 0$.

If $0 < \alpha_i^* < 1/\nu n$ then (KKT2) implies that $\xi_i^* = 0$ and thus (KKT1) implies that $\langle \mathbf{w}^* \cdot \boldsymbol{\phi}(\mathbf{z}_i)\rangle = \rho^*$. As $\mathbf{z}_i \in \mathbb{R}^m_- \times \mathbb{R}^s_+$, this means that $\mathbf{z}_i \in F(\hat{T}_{\text{uDEA}})$, and so part (1) holds.

Now, assume that $\mathbf{z}_i \in \hat{T}_{\text{uDEA}} \setminus F(\hat{T}_{\text{uDEA}})$, then the second equality of (KKT1) does not hold, so we must have $\alpha_i^* = 0$, and (2) holds.

For (3), we first assume that $\xi_i^* \neq 0$. Then, $\beta_i^* = 0$ by (KKT2) and so, by (KKT1), we must have $\langle \mathbf{w}^* \cdot \boldsymbol{\phi}(\mathbf{z}_i)\rangle = \rho^* - \xi_i^* < \rho^*$, that is $\mathbf{z}_i \notin \hat{T}_{\text{uDEA}}$. Conversely, if $\mathbf{z}_i \notin \hat{T}_{\text{uDEA}}$, we have that $\langle \mathbf{w}^* \cdot \boldsymbol{\phi}(\mathbf{z}_i)\rangle < \rho^*$. But from the statement of the problem, we must have $\langle \mathbf{w}^* \cdot \boldsymbol{\phi}(\mathbf{z}_i)\rangle \geq \rho^* - \xi_i^*$. Thus, $\xi_i^* \neq 0$. □

The converses to (1) and (2) are not necessarily true, as the KKT conditions force at least one of the two components involved to be 0, but both could be 0 at the same time.

We now study the link between the hyperparameter $\nu$ and the number of support vectors and outliers obtained. We begin by defining these concepts.

**Definition 3.8.** The DMUs $\mathbf{z}_i$ with $\alpha_i^* > 0$ are called *support vectors*. Let $n_{SV}$ be the number of support vectors, $n_{SV} = |\{i : \alpha_i^* > 0\}|$. The DMUs $\mathbf{z}_i$ with $\xi_i^* > 0$ are called *outliers*. Let $n_{OL}$ be the number of outliers, that is $n_{OL} = |\{i : \xi_i^* > 0\}|$.

We will prove that $\nu$ controls the proportion of points that are allowed to be outliers and the proportion of points that are forced to be support vectors. Recall that $(\mathbf{w}^*, \boldsymbol{\xi}^*, \rho^*, \boldsymbol{\alpha}^*, \boldsymbol{\beta}^*, \boldsymbol{\gamma}^*, \alpha_0^*)$ is a solution of the KKT conditions.

**Lemma 3.9.** *The number of outliers is at most* $\nu n(1 - \alpha_0^*) \leq \nu n$ *and the number of support vectors is at least* $\nu n(1 - \alpha_0^*)$*. In other words,*

$$n_{OL} \leq \nu n(1 - \alpha_0^*) \leq n_{SV}.$$

**Proof.** The values $\alpha_i^*$ have the restrictions $0 \leq \alpha_i^* \leq 1/n\nu$ for $i \in \{1,\dots,n\}$ and $\sum_{i=1}^{n} \alpha_i^* + \alpha_0^* = 1$. Thus, $0 \leq \alpha_0^* = 1 - \sum_{i=1}^{n} \alpha_i^* \leq 1$. A point $\mathbf{z}_i$ is an outlier if and only if $\xi_i^* > 0$ so, by Proposition 3.7 (2), $\beta_i^* = 0$ and so $\alpha_i^* = 1/\nu n$. Thus, we have $1 = \sum_{i=1}^{n} \alpha_i^* + \alpha_0^* \geq n_{OL}/\nu n + \alpha_0^*$, that is $n_{OL} \leq \nu n(1 - \alpha_0^*)$.

Furthermore, the only elements that contribute to the sum are the support vectors $\mathbf{z}_i$, which have $0 < \alpha_i^* \leq 1/\nu n$. Thus, we have $1 = \sum_{i=1}^{n} \alpha_i^* + \alpha_0^* \leq n_{SV}/\nu n + \alpha_0^*$, and so $\nu n(1 - \alpha_0^*) \leq n_{SV}$. □

In particular, if $\nu < 1/n$ then $n_{OL} \leq \nu n(1 - \alpha_0) < 1$, so $n_{OL} = 0$, and no outliers would be allowed in this case. Therefore, an appropriate interval of values for $\nu$, which should be small enough to allow for few if any outliers, is $[1/n, 0.1]$ whenever $n \geq 10$, and $[0.1, 0.3]$ whenever $n \leq 10$. Thus, unless we have very few DMUs, we allow for a maximum of 10% of DMUs to be outliers, a minimum of 0, and we choose the value of $\nu$ in this interval that yields the estimator that attains the smallest mean squared error on the test set. In terms of computing time, we note that by [28, Table 1], larger values of $\nu$ yield longer training times, whereas choosing a small $\nu$, as we do, does not make the algorithm much slower than the algorithm without this regularization term.

### 3.5. Technical inefficiency

We now describe how to calculate the technical inefficiency of each DMU with respect to $\hat{T}_{\text{uDEA}}$. For this purpose, we choose the directional distance function (DDF), as described in Definition 2.2 with $\mathbf{g} \in \mathbb{R}^{m+s}_{+}$, and $\mathbf{g} \neq \mathbf{0}$. To adapt it to the uDEA setting, we let $(\mathbf{w}^*, \boldsymbol{\xi}^*, \rho^*)$ be an optimal solution of (5) and consider $\hat{T}_{\text{uDEA}}$. Then, in order to obtain the inefficiency $\delta$ of a DMU $\mathbf{z}$, we solve the following optimization problem:

$$\begin{aligned}\delta_{\text{uDEA}}(\mathbf{z}, \mathbf{g}) &= \max\{\delta \in \mathbb{R} : (\mathbf{z} + \delta\mathbf{g}) \in \hat{T}_{\text{uDEA}}\} \\ &= \max\{\delta : \langle \mathbf{w}^* \cdot \boldsymbol{\phi}(\mathbf{z} + \delta\mathbf{g})\rangle \geq \rho^*\}. \end{aligned} \tag{10}$$

This program, however, is not linear due to the appearance of the maximum function in the transformation mapping $\boldsymbol{\phi}$. Hence, we linearize it by adding a new variable $\boldsymbol{\sigma} \in \mathbb{R}^h$. We then obtain the following linear program:

$$\begin{aligned} \max_{\delta\in\mathbb{R},\boldsymbol{\sigma}\in\mathbb{R}^h} \quad & M\delta - \sum_{j=m+s+1}^{m+s+h} \sigma_j \\ \text{subject to} \quad & -\sum_{j=1}^{m+s} w_j^*(z(j) + \delta g(j)) - \sum_{j=m+s+1}^{m+s+h} w_j^*\sigma_j \geq \rho^* \\ & \sigma_j \geq \mu \qquad\qquad \forall j \in \{m+s+1, \dots, m+s+h\} \\ & \sigma_j \geq \sum_{k=1}^{m+s} \left[p_j(k)\,(z(k) + \delta g(k))\right] + q_j \qquad \forall j \in \{m+s+1, \dots, m+s+h\} \end{aligned} \tag{11}$$

Note that, in order to force $\sigma_j$ to take the value $\sigma_j = \max\{\mu, \langle \mathbf{p}_j \cdot (\mathbf{z} + \delta\mathbf{g})\rangle + q_j\}$, we penalize it in the objective function, and $M$ is a constant large enough so that small changes in $\delta$ affect the objective function more than corresponding changes in $\boldsymbol{\sigma}$.

We now prove that an optimal solution $(\delta^*, \boldsymbol{\sigma}^*)$ of (11) gives us an optimal solution of (10), so that we can solve the linear problem to obtain the directional distance function inefficiency of a DMU when uDEA is applied. We first prove the following auxiliary result.

**Lemma 3.10.** *Let $(\delta^*, \boldsymbol{\sigma}^*)$ be an optimal solution of (11). Then, $\sigma_j^* = \max\{\mu, \langle \mathbf{p}_j \cdot (\mathbf{z} + \delta^*\mathbf{g})\rangle + q_j\}$ for all $j \in \{m+s+1, \dots, m+s+h\}$.*

**Proof.** Suppose $(\delta^*, \boldsymbol{\sigma}^*)$ is an optimal solution to (11) with $\sigma_j^* > \max\{\mu, \langle \mathbf{p}_j \cdot (\mathbf{z} + \delta^*\mathbf{g})\rangle + q_j\}$ for some $j$. Then, we consider the potential solution $(\delta^*, \boldsymbol{\sigma}')$ where $\sigma_i' = \sigma_i^*$ for $i \neq j$ and $\sigma_j' = \max\{\mu, \langle \mathbf{p}_j \cdot (\mathbf{z} + \delta^*\mathbf{g})\rangle + q_j\} < \sigma_j^*$. This is still a feasible point of (11) as the first restriction becomes greater, and the last two restrictions do not change, so it is still a solution to the optimization problem with a larger objective, contradicting the assumption that $(\delta^*, \boldsymbol{\sigma}^*)$ was optimal. □

We can now prove the main result.

**Proposition 3.11.** *The optimal value of (10) is $\delta^*$ if and only if $(\delta^*, \boldsymbol{\sigma}^*)$ is an optimal solution of (11), with $\sigma_j^* = \max\{\mu, \langle \mathbf{p}_j \cdot (\mathbf{z} + \delta^*\mathbf{g})\rangle + q_j\}$ for all $j \in \{m+s+1, \dots, m+s+h\}$.*

**Proof.** As $\mathbf{g} \geq \mathbf{0}$, whenever $\delta_1 \geq \delta_2$ we have $\mathbf{z} + \delta_1\mathbf{g} \geq \mathbf{z} + \delta_2\mathbf{g}$ so by Lemma 3.1 applied to each component of $\boldsymbol{\phi}$, and since $\mathbf{w}^* \geq \mathbf{0}$, we have $\langle \mathbf{w}^* \cdot \boldsymbol{\phi}(\mathbf{z} + \delta_1\mathbf{g})\rangle \leq \langle \mathbf{w}^* \cdot \boldsymbol{\phi}(\mathbf{z} + \delta_2\mathbf{g})\rangle$ and thus, whenever $\delta_1$ is an optimal solution of (10), we have $\langle \mathbf{w}^* \cdot \boldsymbol{\phi}(\mathbf{z} + \delta_1\mathbf{g})\rangle = \rho^*$.

Let $(\delta^*, \boldsymbol{\sigma}^*)$ be a solution of (11) and suppose that $\delta^*$ is not the optimal value of (10). Then there exists $\delta' > \delta^*$ such that $\delta'$ is an optimal solution of (10). Then, $\langle \mathbf{w}^* \cdot \boldsymbol{\phi}(\mathbf{z} + \delta'\mathbf{g})\rangle = \rho^*$.

Define $\boldsymbol{\sigma}'$ by $\sigma_j' = \max\{\mu, \langle \mathbf{p}_j \cdot (\mathbf{z} + \delta'\mathbf{g})\rangle + q_j\}$ for each $j \in \{m+s+1, \dots, m+s+h\}$. Then $(\delta', \boldsymbol{\sigma}')$ is a feasible solution of (11), as $-\sum_{j=1}^{m+s} w_j^* z(j) - \sum_{j=m+s+1}^{m+s+h} w_j^*\sigma_j' = \rho^*$ and by definition of $\boldsymbol{\sigma}'$.

Since $\delta' > \delta^*$ and $M$ is large enough (to offset the effect of the change in $\sigma_j'$), this is a solution of (11) with $M\delta' - \sum_{j=m+s+1}^{m+s+h} \sigma_j' > M\delta^* - \sum_{j=m+s+1}^{m+s+h} \sigma_j^*$, hence $(\delta^*, \boldsymbol{\sigma}^*)$ is not an optimal solution of (11), contradicting our assumption. Thus, whenever $(\delta^*, \boldsymbol{\sigma}^*)$ is an optimal solution of (11), $\delta^*$ is the optimal value of (10).

**Table 2**
Hyperparameters and variables of uDEA.

| Hyperparameters | | Primal variables | Dual variables |
|---|---|---|---|
| $h \in \mathbb{N}$ | | $\mathbf{w} \in \mathbb{R}^{m+s+h}_{+}$ | $\boldsymbol{\alpha}, \boldsymbol{\beta} \in \mathbb{R}^n_{+}$ |
| $\nu \in \mathbb{R}$ | | $\boldsymbol{\xi} \in \mathbb{R}^n_{+}$ | $\boldsymbol{\gamma} \in \mathbb{R}^{m+s+h}_{+}$ |
| $\mu \in \mathbb{R}_{-}$ | | $\rho \in \mathbb{R}$ | $\alpha_0 \in \mathbb{R}$ |
| $\mathbf{p}_k \in \mathbb{R}^{m+s}_{+}$ | for $k \in \{m+s+1, \dots, m+s+h\}$ | | |
| $q_k \in \mathbb{R}$ | | | |

Conversely, assume that $\delta^{(\dagger)}$ is the optimal value of (10). Define $\sigma_j^{(\dagger)} := \max\{\mu, \langle \mathbf{p}_j \cdot (\mathbf{z} + \delta^{(\dagger)}\mathbf{g})\rangle + q_j\}$. We want to prove that $(\delta^{(\dagger)}, \boldsymbol{\sigma}^{(\dagger)})$ is an optimal solution of (11). Since $\delta^{(\dagger)}$ is a solution of (10) and by definition of $\boldsymbol{\sigma}^{(\dagger)}$, $(\delta^{(\dagger)}, \boldsymbol{\sigma}^{(\dagger)})$ is a feasible solution of (11). Now, assume that $(\delta^{(\dagger)}, \boldsymbol{\sigma}^{(\dagger)})$ is not optimal, then there exists an optimal solution $(\delta', \boldsymbol{\sigma}')$ of (11) which satisfies $M\delta' - \sum_{j=m+s+1}^{m+s+h} \sigma_j' > M\delta^{(\dagger)} - \sum_{j=m+s+1}^{m+s+h} \sigma_j^{(\dagger)}$.

Now, as $M$ is large enough, we have $\delta' \geq \delta^{(\dagger)}$ and, as $\delta^{(\dagger)}$ is the optimal value of (10), $\delta' \leq \delta^{(\dagger)}$. Thus, we have $\delta' = \delta^{(\dagger)}$. But then Lemma 3.10 implies that for all $j$ we must have $\sigma_j^{(\dagger)} = \sigma_j'$, contradicting our setup. Hence, we conclude that $(\delta^{(\dagger)}, \boldsymbol{\sigma}^{(\dagger)})$ is an optimal solution of (11). □

### 3.6. Steps of the algorithm

In this section, we gather the various pieces previously introduced and describe step by step the uDEA algorithm. We summarize the roles of the various variables involved in the uDEA algorithm in Table 2. We recall that $\mathbf{z}_i \in \mathbb{R}^m_{-} \times \mathbb{R}^s_{+}$ are the given netputs, so that the values of $n, m, s$ are given by the dataset $\mathcal{Z}$.

The total number of potential hyperparameters is $3 + h(m + s + 1)$, which we observe is quadratic in $m$ and $s$, obtained from the dataset, as well as $h$, which is a hyperparameter.

The steps of the algorithm are as follows[3] (see also Fig. 1).

1. We begin by letting $h = n$, that is, we choose to have a hyperplane corresponding to each DMU.
2. We solve (8) 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 the values for $\mu_i$, so we calculate $\mu_{min} = \min_i\{\mu_i\}$.
3. We plug these values of $\mathbf{p}_k$ and $q_k$ into $\boldsymbol{\phi}$. In particular, for each $\mathbf{z}_i \in \mathcal{Z}$, $\boldsymbol{\phi}(\mathbf{z}_i)$ is a fixed vector, as well as $\boldsymbol{\phi}(\mathbf{0})$.
4. We 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%.
5. At this point, the quadratic Program (5) (on $\mathcal{Z}_{train}$) is ready to be solved, with variables $\mathbf{w}, \boldsymbol{\xi}, \rho$, and the hyperparameters left to tune are $\nu$ and $\mu$. Everything else in the program is fixed.
6. Then, the hyperparameters $\nu$ and $\mu$ remain to be tuned. We suggest choosing 5 possible values for each from their respective appropriate intervals. One possible choice for the intervals is $\mu \in \left[\mu_{min}, 0\right)$, and $\nu \in [1/n, 0.1]$ (whenever $n \geq 10$).
7. For each pair $(\mu, \nu)$, we solve Model (5) using the training set, obtaining the optimal solution $(\mathbf{w}^*, \boldsymbol{\xi}^*, \rho^*)$.
8. For each $\mathbf{z}_i$ in the test set, we solve Program (11) to calculate $\delta_{\text{uDEA}}(\mathbf{z}_i, \mathbf{g}_i)$ using Farrell's output-oriented measure of efficiency, that is $\mathbf{g}_i = (0, \dots, 0, z_i(m+1), \dots, z_i(m+s))$.
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. Then, 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 $\text{MSE}(\mu, \nu) = \sum_{\mathbf{z}_i \in \mathcal{Z}} \sum_k (\hat{z}_i(k) -$

[3] See https://github.com/JuanAparicioUMH/uDEA, where each step is shown in detail and, additionally, an example is provided and solved by uDEA.

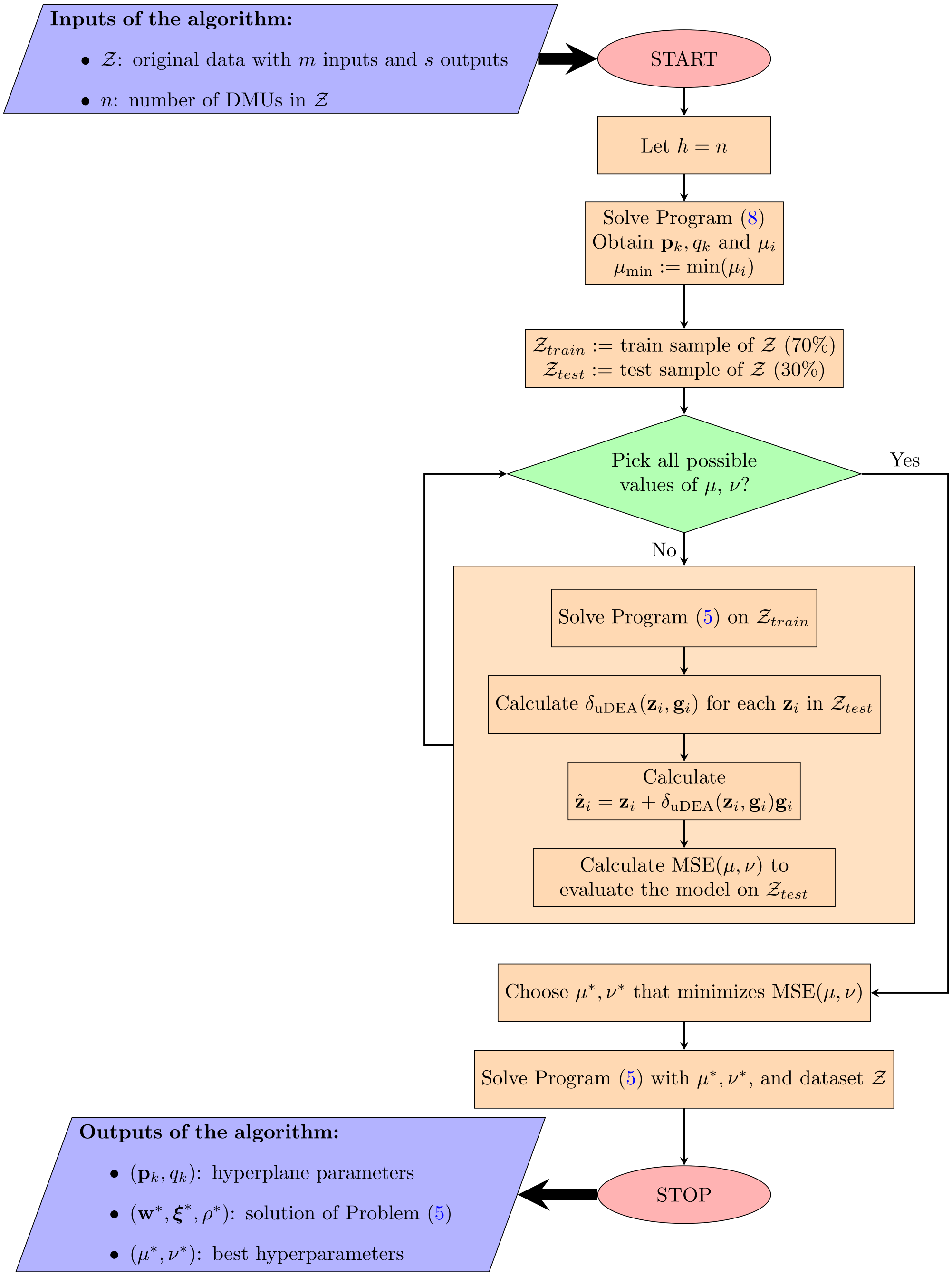
Inputs of the algorithm:
• $\mathcal{Z}$: original data with $m$ inputs and $s$ outputs
• $n$: number of DMUs in $\mathcal{Z}$
START
Let $h = n$
Solve Program (8)
Obtain $\mathbf{p}_k, q_k$ and $\mu_i$
$\mu_{\min} := \min(\mu_i)$
$\mathcal{Z}_{train}$ := train sample of $\mathcal{Z}$ (70%)
$\mathcal{Z}_{test}$ := test sample of $\mathcal{Z}$ (30%)
Pick all possible values of $\mu$, $\nu$?
Yes
No
Solve Program (5) on $\mathcal{Z}_{train}$
Calculate $\delta_{\text{uDEA}}(\mathbf{z}_i, \mathbf{g}_i)$ for each $\mathbf{z}_i$ in $\mathcal{Z}_{test}$
Calculate
$\hat{\mathbf{z}}_i = \mathbf{z}_i + \delta_{\text{uDEA}}(\mathbf{z}_i, \mathbf{g}_i)\mathbf{g}_i$
Calculate MSE$(\mu, \nu)$ to evaluate the model on $\mathcal{Z}_{test}$
Choose $\mu^*, \nu^*$ that minimizes MSE$(\mu, \nu)$
Solve Program (5) with $\mu^*, \nu^*$, and dataset $\mathcal{Z}$
STOP
Outputs of the algorithm:
• $(\mathbf{p}_k, q_k)$: hyperplane parameters
• $(\mathbf{w}^*, \boldsymbol{\xi}^*, \rho^*)$: solution of Problem (5)
• $(\mu^*, \nu^*)$: best hyperparameters

**Fig. 1.** Flowchart of the uDEA algorithm.

$z_i(k))^2$ and choose the hyperparameter pair $(\mu^*, \nu^*)$ which minimizes this value.

11. Once we have selected the best $\mu^*$ and $\nu^*$, we solve Program (5) with these values of $\mu^*$ and $\nu^*$ on the whole dataset $\mathcal{Z}$, obtaining the final values for $(\mathbf{w}^*, \boldsymbol{\xi}^*, \rho^*)$.

## 4. Computational experience

This section shows the results obtained from a computational experience with the aim of comparing the DEA and uDEA methods. Hence, we resort to data simulation for a systematic assessment of these frontier methods. The descriptions of the frontiers that we simulate appear in Table 3.

For the theoretical production frontiers, we used several typical Cobb–Douglas functions from the literature, where the exponents of the considered variables add up to 0.5. The input data were randomly sampled from $Uni[0,1]$ and the inefficiency term from $u \sim \exp(1/3)$. We tested with data sizes of 30, 50, 70, 100 and 200. We ran 100 trials for each combination of number of inputs and sample size. The performance of each method is evaluated by two typical measures: the mean squared error (MSE) and bias.

In our simulation setting, we resort to a Cobb–Douglas production function [5], which is a classical and usual specification of a production function in microeconomics [45]. In a Cobb–Douglas production function, the exponent of each input represents the share of an increase in the output attributable to that input. Additionally, the sum of the

**Table 3**
Considered theoretical production frontiers.

| # inputs | $f(\mathbf{x})$ |
|---|---|
| 1 | $f(\mathbf{x}) = x_1^{0.5}$ |
| 2 | $f(\mathbf{x}) = x_1^{0.4} \cdot x_2^{0.1}$ |
| 3 | $f(\mathbf{x}) = x_1^{0.3} \cdot x_2^{0.1} \cdot x_3^{0.1}$ |
| 4 | $f(\mathbf{x}) = x_1^{0.3} \cdot x_2^{0.1} \cdot x_3^{0.08} \cdot x_4^{0.02}$ |
| 5 | $f(\mathbf{x}) = x_1^{0.3} \cdot x_2^{0.1} \cdot x_3^{0.08} \cdot x_4^{0.01} \cdot x_5^{0.01}$ |
| 6 | $f(\mathbf{x}) = x_1^{0.3} \cdot x_2^{0.1} \cdot x_3^{0.08} \cdot x_4^{0.01} \cdot x_5^{0.006} \cdot x_6^{0.004}$ |
| 9 | $f(\mathbf{x}) = x_1^{0.3} \cdot x_2^{0.1} \cdot x_3^{0.08} \cdot x_4^{0.005} \cdot x_5^{0.004} \cdot x_6^{0.001} \cdot x_7^{0.005} \cdot x_8^{0.004} \cdot x_9^{0.001}$ |
| 12 | $f(\mathbf{x}) = x_1^{0.2} \cdot x_2^{0.075} \cdot x_3^{0.025} \cdot x_4^{0.05} \cdot x_5^{0.05} \cdot x_6^{0.08} \cdot x_7^{0.005} \cdot x_8^{0.004} \cdot x_9^{0.001} \cdot x_{10}^{0.005} \cdot x_{11}^{0.004} \cdot x_{12}^{0.001}$ |
| 15 | $f(\mathbf{x}) = x_1^{0.15} \cdot x_2^{0.025} \cdot x_3^{0.025} \cdot x_4^{0.05} \cdot x_5^{0.025} \cdot x_6^{0.025} \cdot x_7^{0.05} \cdot x_8^{0.05} \cdot x_9^{0.08} \cdot x_{10}^{0.005} \cdot x_{11}^{0.004} \cdot x_{12}^{0.001} \cdot x_{13}^{0.005} \cdot x_{14}^{0.004} \cdot x_{15}^{0.001}$ |

exponents is associated with the returns to scale of the production process. In particular, a value less than one in the sum of the exponents is related to non-increasing returns to scale, while a value equal to one signals constant returns to scale, and a value greater than one identifies non-decreasing returns to scale. In our simulation scenarios, we arbitrarily set this sum to be always equal to 0.5 with the objective of maintaining this assumption constant for all the analyzed settings (a similar assumption was made for the corresponding simulation scenarios in [19], and [20], where Cobb–Douglas production functions were simulated). Other returns to scales and data configurations could be considered, but this extension is beyond the scope of this paper, and it is a line for future research.

Table 4 describes the MSE and bias statistics for the two methods evaluated (DEA and uDEA). The first two columns indicate the sample size and the number of inputs. The next two columns show the MSE associated with DEA and uDEA, respectively. The subsequent two columns indicate the bias of these techniques. In addition, we report in brackets the relative difference between DEA and uDEA with respect to MSE and bias for ease of comparison. These percentages are the reduction in MSE and bias when uDEA is applied instead of DEA.

We observe that in the single-input case the results of both methods are almost identical, and that as the number of inputs increases, both the MSE and bias of the uDEA technique grows much more slowly than in the case of DEA, which translates into increasing percentages of improvement. When the number of DMUs is small, at a certain point, increasing the number of inputs results in smaller relative improvements for uDEA when compared to DEA, whereas with larger numbers of inputs this relative improvement keeps increasing. In some cases, we obtain up to a 83.34% improvement in MSE and up to a 59.85% improvement in bias, both of which are attained with 200 DMUs and 15 inputs.

Fig. 2 shows a graphical example of the result of one of our simulations. This example consists of 1 input, 1 output and 30 DMUs, and illustrates how, after the fine tuning process, the frontier turns at appropriate points near the theoretical frontier. In this case, uDEA has 0.0008 MSE and 0.0288 bias whereas DEA yields an MSE of 0.0019 and bias of 0.0437, a large improvement.

From a computational point of view, it is worth mentioning the computing time spent in the new approach in comparison with the DEA technique. The simulations were executed on a PC with a 1.8 GHz dual-core Intel Core i7 processor, 8 Gigabyte of RAM and a Microsoft Windows 10 Enterprise operating system. The algorithm was implemented in Python code. So, for an experiment composed by 50 DMUs and three inputs, the uDEA technique used 7.61s for calculating all the estimations, while the DEA technique utilized 0.79s (approximately ten times less than uDEA).

**Table 4**
Result of DEA and uDEA estimation methods based on the MSE and bias criteria.

| Num. obs. | Num. inp. | Mean Squared Error | | BIAS | |
|---|---|---|---|---|---|
| | | DEA | uDEA | DEA | uDEA |
| 30 | 1 | 0.0022 | 0.0022 (0.00%) | 0.0437 | 0.0435 (0.40%) |
| | 2 | 0.0064 | 0.0058 (9.05%) | 0.0776 | 0.0721 (7.14%) |
| | 3 | 0.0121 | 0.0056 (53.92%) | 0.1082 | 0.0708 (34.54%) |
| | 4 | 0.0170 | 0.0058 (65.73%) | 0.1283 | 0.0730 (43.13%) |
| | 5 | 0.0191 | 0.0048 (74.71%) | 0.1361 | 0.0663 (51.29%) |
| | 6 | 0.0264 | 0.0080 (69.60%) | 0.1605 | 0.0852 (46.91%) |
| | 9 | 0.0391 | 0.0163 (58.24%) | 0.1961 | 0.1191 (39.24%) |
| | 12 | 0.0376 | 0.0244 (35.18%) | 0.1920 | 0.1472 (23.34%) |
| | 15 | 0.0396 | 0.0367 (7.42%) | 0.1973 | 0.1830 (7.29%) |
| 50 | 1 | 0.0011 | 0.0011 (0.00%) | 0.0308 | 0.0308 (0.00%) |
| | 2 | 0.0038 | 0.0037 (2.46%) | 0.0604 | 0.0595 (1.52%) |
| | 3 | 0.0086 | 0.0055 (35.52%) | 0.0917 | 0.0718 (21.72%) |
| | 4 | 0.0141 | 0.0060 (57.23%) | 0.1172 | 0.0753 (35.76%) |
| | 5 | 0.0139 | 0.0044 (68.34%) | 0.1170 | 0.0641 (45.23%) |
| | 6 | 0.0233 | 0.0083 (64.31%) | 0.1513 | 0.0883 (41.64%) |
| | 9 | 0.0327 | 0.0093 (71.63%) | 0.1796 | 0.0918 (48.87%) |
| | 12 | 0.0357 | 0.0099 (72.31%) | 0.1878 | 0.0944 (49.75%) |
| | 15 | 0.0370 | 0.0240 (35.20%) | 0.1911 | 0.1448 (24.21%) |
| 70 | 1 | 0.0006 | 0.0006 (0.00%) | 0.0231 | 0.0231 (0.00%) |
| | 2 | 0.0030 | 0.0029 (2.07%) | 0.0536 | 0.0522 (2.69%) |
| | 3 | 0.0069 | 0.0046 (33.28%) | 0.0822 | 0.0664 (19.28%) |
| | 4 | 0.0109 | 0.0055 (49.66%) | 0.1035 | 0.0726 (29.88%) |
| | 5 | 0.0120 | 0.0044 (63.56%) | 0.1089 | 0.0643 (41.01%) |
| | 6 | 0.0194 | 0.0090 (53.51%) | 0.1385 | 0.0920 (33.55%) |
| | 9 | 0.0298 | 0.0092 (69.28%) | 0.1715 | 0.0918 (46.46%) |
| | 12 | 0.0340 | 0.0083 (75.57%) | 0.1834 | 0.0879 (52.06%) |
| | 15 | 0.0365 | 0.0124 (65.99%) | 0.1902 | 0.1010 (46.89%) |
| 100 | 1 | 0.0004 | 0.0004 (0.00%) | 0.0181 | 0.0180 (0.50%) |
| | 2 | 0.0023 | 0.0022 (3.98%) | 0.0472 | 0.0455 (3.71%) |
| | 3 | 0.0052 | 0.0040 (23.82%) | 0.0716 | 0.0619 (13.53%) |
| | 4 | 0.0095 | 0.0044 (53.62%) | 0.0968 | 0.0654 (32.49%) |
| | 5 | 0.0107 | 0.0048 (55.11%) | 0.1031 | 0.0681 (33.94%) |
| | 6 | 0.0166 | 0.0079 (52.16%) | 0.1282 | 0.0876 (31.65%) |
| | 9 | 0.0279 | 0.0107 (61.51%) | 0.1664 | 0.1006 (39.52%) |
| | 12 | 0.0336 | 0.0071 (78.83%) | 0.1827 | 0.0816 (55.36%) |
| | 15 | 0.0356 | 0.0080 (77.68%) | 0.1883 | 0.0841 (55.35%) |
| 200 | 1 | 0.0002 | 0.0002 (0.00%) | 0.0122 | 0.0121 (0.33%) |
| | 2 | 0.0011 | 0.0011 (0.00%) | 0.0331 | 0.0331 (0.00%) |
| | 3 | 0.0030 | 0.0029 (3.39%) | 0.0545 | 0.0545 (0.00%) |
| | 4 | 0.0062 | 0.0038 (38.89%) | 0.0781 | 0.0608 (22.14%) |
| | 5 | 0.0074 | 0.0038 (49.35%) | 0.0859 | 0.0606 (29.52%) |
| | 6 | 0.0125 | 0.0058 (53.77%) | 0.1114 | 0.0756 (32.16%) |
| | 9 | 0.0235 | 0.0095 (59.75%) | 0.1530 | 0.0962 (37.12%) |
| | 12 | 0.0305 | 0.0073 (76.18%) | 0.1743 | 0.0838 (51.89%) |
| | 15 | 0.0345 | 0.0057 (83.34%) | 0.1855 | 0.0745 (59.85%) |

We also compared uDEA with other recent approaches in the same line of research. In particular, we compared the new method with EATBoosting by [19] and Data Envelopment Analysis Machines (DEAM) by [20]. In the first case, the boosting algorithm is used, while, in the second case, the Structural Risk Minimization principle in machine learning is applied. In both cases, the provided estimate of the technology satisfies the following set of microeconomic properties: convexity, free disposability and envelopment. The same happens in the case of uDEA. These facts make the comparison among these techniques fair. Another recent approach based on machine learning to estimate efficiency scores is [46]. However, the comparison with this last technique is not direct because this method was not designed to guarantee the three previously mentioned properties on the technology.

Table 5 shows the results obtained by the EATBoosting and DEAM approaches[4]. Combining this information with the results obtained using the new method in Table 4, we observe that uDEA is competitive

[4] In both cases, we executed the corresponding codes during approximately 10 days, i.e., the same time spent by uDEA to get the results for all the simulated scenarios. In 10 days, the EATBoosting method by [19] was able to execute all the scenarios, although we had to resort to a heuristic version of the model suggested by these authors. In the case of the DEAM approach

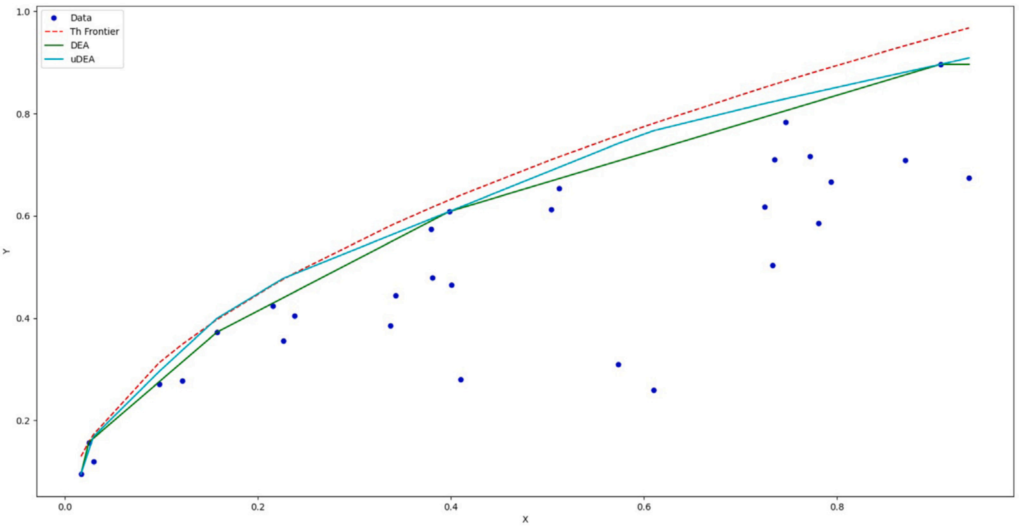

**Fig. 2.** The uDEA frontier vs the DEA and theoretical frontiers.

with respect to the other approaches regarding both bias and mean squared error. Our results seem to indicate that the uDEA estimator could be considered as one of the valid alternatives that currently exist in the literature on the measurement of technical efficiency. Additionally, and from a computational point of view, the execution of the EATBoosting method was only possible by resorting to the heuristic version of the model proposed by their authors (see [19]). As for DEAM, it is worth mentioning that, in the period of 10 days used for all the approaches to solve all the simulated scenarios, this technique was only able to solve the settings involving up to 70 DMUs. In contrast, the new approach, based on the adaptation of OneClassSVM, determined the solution for all the simulated scenarios and without resorting to heuristic methods.

## 5. Conclusions and future works

In this paper, we have developed an unsupervised machine learning method, which we call uDEA, to study production frontiers. Up until quite recently, these areas had grown separately without much intertwining despite their shared characteristics. After all, the study of production frontiers can be seen as a learning problem consisting of determining the boundary of the region of feasible points or technology by using the observed DMUs as learning data. Recently, a few bridges have been built using various supervised machine learning methods for the purpose of frontier estimation, such as Boosting and the Structural Risk Minimization principle in [19,20], respectively. These have a distinguished output variable and thus face challenges when generalizing to higher output dimensions. This unsupervised, homogeneous treatment of the variables is, in a way, already present in DEA, when the netput notation is used, since the only difference between inputs and outputs comes from the property of free disposability, which disappears in netput notation.

The main novel contribution of our method of unsupervised Data Envelopment Analysis (uDEA) is the unsupervised treatment of the data, by not distinguishing between input and output variables, which allows it to generalize without any changes to the multi-output setting. We do this by adapting the OneClassSVM algorithm to deal with production functions in the netput setting, with the use of a piecewise linear transformation function in order to obtain a piecewise linear frontier similar to the frontier obtained by DEA. We prove that the uDEA estimated technology satisfies convexity and free disposability, while it does not fulfill minimal extrapolation. Then, we describe a DEA-based method for obtaining the hyperplanes involved in this transformation function, and introduce two hyperparameters: $\nu$, which controls the proportion of outliers allowed, and $\mu$, which acts as an offset parameter thus allowing flexibility in the frontier obtained.

We have evaluated the performance of the uDEA algorithm against the standard DEA with simulated data using Monte Carlo experiments where we observe that uDEA outperforms DEA with respect to multiple traditional error measures such as mean squared error (MSE) and bias, with larger improvements as the number of variables and DMUs increases. Comparing the MSE values, we observe that with only one input and one output, the performance is very similar, and as the number of input variables increases, the percentage of improvement of the uDEA score tends to grow when compared to the corresponding DEA one, reaching improvement values of up to 83%. Regarding the bias, we observe a similar increasing tendency, reaching up to 60% improvement in some cases. We also observe that the hyperparameter $\nu$ controlling the proportion of outliers allowed takes small values, which slightly increase as more DMUs and dimensions are added.

Another advantage of uDEA is that, whereas DEA suffers, via the assumption of the minimal extrapolation property, from a problem of overfitting in the machine learning sense, uDEA avoids this problem via the use of a SVM-style regularizer, and yields an estimated frontier which is closer to the theoretical frontier than the convex hull of the data. Furthermore, whereas DEA determines the degree of ‘relative’ technical efficiency for each DMU, our approach identifies ‘absolute’ efficiency, that is, efficiency measured with respect to the (estimated) production frontier associated with the data generating process from which the data is drawn, instead of the efficiency measured in comparative terms with the performance of exactly the $n$ observed units in our data sample.

Finally, we mention some possible adaptations of the uDEA algorithm for further research. The choice of transformation function and the position of the hyperplanes will greatly affect the frontiers obtained and may admit different shapes to explore. The hyperplane parameters could also be tuned as hyperparameters, although due to their potentially large number, this will probably be computationally expensive. Among the regularization methods that exist in the literature, in this paper, we resorted to the same type used by [28], since we have based our model in the adaptation of the technique introduced by these same authors. Nevertheless, other regularizations could be used (see,

by [20], their algorithm only provided the results for the scenarios up to 70 observations in this period.

**Table 5**
Result of EATBoosting and DEAM estimation methods based on the MSE and bias criteria.

| Num. obs. | Num. inp. | Mean Squared Error | | BIAS | |
|---|---|---|---|---|---|
| | | EATBoosting | DEAM | EATBoosting | DEAM |
| 30 | 1 | 0.0017 | 0.0018 | 0.0302 | 0.0326 |
| | 2 | 0.0041 | 0.0053 | 0.0481 | 0.0574 |
| | 3 | 0.0072 | 0.0083 | 0.0671 | 0.0733 |
| | 4 | 0.0102 | 0.0103 | 0.0778 | 0.0805 |
| | 5 | 0.0126 | 0.0127 | 0.0845 | 0.0869 |
| | 6 | 0.0147 | 0.0147 | 0.0912 | 0.0928 |
| | 9 | 0.0189 | 0.0190 | 0.1027 | 0.1052 |
| | 12 | 0.0222 | 0.0214 | 0.1126 | 0.1125 |
| | 15 | 0.0231 | 0.0278 | 0.1147 | 0.1243 |
| 50 | 1 | 0.0011 | 0.0009 | 0.0225 | 0.0236 |
| | 2 | 0.0024 | 0.0034 | 0.0339 | 0.0465 |
| | 3 | 0.0043 | 0.0060 | 0.0474 | 0.0631 |
| | 4 | 0.0060 | 0.0078 | 0.0565 | 0.0687 |
| | 5 | 0.0085 | 0.0112 | 0.0679 | 0.0810 |
| | 6 | 0.0108 | 0.0117 | 0.0769 | 0.0830 |
| | 9 | 0.0144 | 0.0166 | 0.0890 | 0.0981 |
| | 12 | 0.0147 | 0.0189 | 0.0906 | 0.1054 |
| | 15 | 0.0181 | 0.0201 | 0.1009 | 0.1027 |
| 70 | 1 | 0.0012 | 0.0006 | 0.0236 | 0.0195 |
| | 2 | 0.0021 | 0.0025 | 0.0323 | 0.0401 |
| | 3 | 0.0034 | 0.0047 | 0.0411 | 0.0554 |
| | 4 | 0.0047 | 0.0067 | 0.0479 | 0.0645 |
| | 5 | 0.0059 | 0.0091 | 0.0552 | 0.0738 |
| | 6 | 0.0081 | 0.0116 | 0.0638 | 0.0823 |
| | 9 | 0.0111 | 0.0163 | 0.0756 | 0.0982 |
| | 12 | 0.0121 | 0.0170 | 0.0818 | 0.1000 |
| | 15 | 0.0136 | 0.0170 | 0.0862 | 0.1004 |
| 100 | 1 | 0.0013 | – | 0.0266 | – |
| | 2 | 0.0023 | – | 0.0363 | – |
| | 3 | 0.0030 | – | 0.0396 | – |
| | 4 | 0.0033 | – | 0.0404 | – |
| | 5 | 0.0044 | – | 0.0464 | – |
| | 6 | 0.0053 | – | 0.0511 | – |
| | 9 | 0.0081 | – | 0.0646 | – |
| | 12 | 0.0093 | – | 0.0712 | – |
| | 15 | 0.0117 | – | 0.4848 | – |
| 200 | 1 | 0.0019 | – | 0.0349 | – |
| | 2 | 0.0039 | – | 0.0512 | – |
| | 3 | 0.0045 | – | 0.0539 | – |
| | 4 | 0.0035 | – | 0.0446 | – |
| | 5 | 0.0034 | – | 0.0429 | – |
| | 6 | 0.0040 | – | 0.0450 | – |
| | 9 | 0.0051 | – | 0.0502 | – |
| | 12 | 0.0059 | – | 0.0556 | – |
| | 15 | 0.0070 | – | 0.0515 | – |

for example, [47]); a topic that deserves further exploration. Other research lines could be the application of the new technique to real databases in various empirical contexts to further check the validity of the technique in practice, in particular multi-output databases.

Furthermore, when efficiency measurement is the concern, a limitation of the new method, in comparison with the standard DEA technique, is that efficiency scores are obtained in a second stage after the technology has been estimated. The uDEA model includes one-sided error terms ($\xi_i \geq 0,\ i = 1, \ldots, n$). However, the direct interpretation of these quantities as technical inefficiency of the assessed units is not trivial. In OneClassSVM, which is the technique that we have adapted to the production context, the value of each decision variable $\xi_i$ is equal to zero for every observation (DMU in our framework) located inside the technology and only attains strictly positive values for observations which are deemed as (slight) outliers for the new technique (weighed by the hyperparameter $\nu$, which can be tuned). Proposing solutions for this weakness of the new approach could be seen as an interesting line for future research. In addition, the possibility of using uDEA to measure productivity change over time and decompose this measure into its usual drivers, i.e., efficiency change, scale efficiency change and technical change, is a topic that deserves future explorations.

## CRediT authorship contribution statement

**Raul Moragues:** Conceptualization, Methodology, Writing – original draft, Writing – review & editing. **Juan Aparicio:** Conceptualization, Methodology. **Miriam Esteve:** Software.

## Declaration of competing interest

All authors declare that they have no conflicts of interest.

## Data availability

Data will be made available on request

## Acknowledgments

We thank three anonymous reviewers for providing constructive comments and help in improving the contents and presentation of this paper. The authors thank the grant PID2019-105952GB-I00 funded by Ministerio de Ciencia e Innovación/ Agencia Estatal de Investigación, Spain /10.13039/501100011033. R. Moragues wishes to thank the Cátedra Santander en Eficiencia *y* Productividad, Miguel Hernandez University (UMH), for the funding provided. Furthermore, M. Esteve was also supported by the Spanish Ministry of Science, Innovation and Universities under Grant FPU17/05365. Additionally, J. Aparicio thanks the grant PROMETEO/2021/063 funded by the Valencian Community (Spain), which supported this work.

## Appendix. Calculation of the dual problem

The Lagrangian corresponding to (5), with KKT multipliers $\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}, \alpha_0$, is:

$$\begin{aligned} L(\mathbf{w}, \boldsymbol{\xi}, \rho, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}, \alpha_0) = & \frac{1}{2}\, \|\mathbf{w}\|^2 + \frac{1}{\nu n} \sum_{i=1}^{n} \xi_i - \rho \\ & + \sum_{i=1}^{n} \alpha_i [-\langle \mathbf{w} \cdot \boldsymbol{\phi}(\mathbf{z}_i) \rangle + \rho - \xi_i] \\ & + \sum_{i=1}^{n} \beta_i(-\xi_i) \\ & + \sum_{i=1}^{m+s+h} \gamma_i(-w_i) \\ & + \alpha_0(\rho - \langle \mathbf{w} \cdot \boldsymbol{\phi}(\mathbf{0}) \rangle) \end{aligned}$$

We observe that $\boldsymbol{\alpha}, \boldsymbol{\beta} \in \mathbb{R}^n$ while $\boldsymbol{\gamma} \in \mathbb{R}^{m+s+h}$ since $\mathbf{w} \in \mathbb{R}^{m+s+h}$, and $\alpha_0 \in \mathbb{R}$. Differentiating with respect to the primal variables and equating to 0 we obtain

$$\frac{dL}{d\mathbf{w}} = \mathbf{w} - \sum_{i=1}^{n} \alpha_i \boldsymbol{\phi}(\mathbf{z}_i) - \boldsymbol{\gamma} - \alpha_0 \boldsymbol{\phi}(\mathbf{0}) \text{ yielding } \mathbf{w} = \sum_{i=1}^{n} \alpha_i \boldsymbol{\phi}(\mathbf{z}_i) + \boldsymbol{\gamma} + \alpha_0 \boldsymbol{\phi}(\mathbf{0}), \tag{A.1}$$

$$\frac{dL}{d\xi_i} = \frac{1}{\nu n} - \alpha_i - \beta_i \text{ yielding } 0 \leq \alpha_i \leq \frac{1}{\nu n} \tag{A.2}$$

and

$$\frac{dL}{d\rho} = -1 + \sum_{i=1}^{n} \alpha_i + \alpha_0, \text{ yielding } \sum_{i=1}^{n} \alpha_i + \alpha_0 = 1. \tag{A.3}$$

The above expressions hold at any stationary point of $L$, so we substitute them into $L$.

$$\begin{aligned}L(\mathbf{w},\boldsymbol{\xi},\rho,\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\gamma},\alpha_0) =&\frac{1}{2}\left\|\mathbf{w}\right\|^2 + \frac{1}{\nu n}\sum_{i=1}^{n}\xi_i - \rho + \sum_{i=1}^{n}\alpha_i[-\langle\mathbf{w}\cdot\boldsymbol{\phi}(\mathbf{z}_i)\rangle + \rho - \xi_i]\\ &+\sum_{i=1}^{n}\beta_i(-\xi_i) + \sum_{i=1}^{m+s+h}\gamma_i(-w_i) + \alpha_0(\rho - \langle\mathbf{w}\cdot\boldsymbol{\phi}(\mathbf{0})\rangle)\end{aligned} \tag{A.4}$$

First, we expand each of the components separately, obtaining

$$\begin{aligned}&\frac{1}{2}\left\|\mathbf{w}\right\|^2 = \frac{1}{2}\langle\mathbf{w}\cdot\mathbf{w}\rangle = \frac{1}{2}\left\|\sum_{i=1}^{n}\alpha_i\boldsymbol{\phi}(\mathbf{z}_i) + \boldsymbol{\gamma} + \alpha_0\boldsymbol{\phi}(\mathbf{0})\right\|^2 =\\ &\frac{1}{2}\Bigg(\sum_{i,j=1}^{n}\alpha_i\alpha_j\langle\boldsymbol{\phi}(\mathbf{z}_i)\cdot\boldsymbol{\phi}(\mathbf{z}_j)\rangle + 2(\sum_{i=1}^{n}\alpha_i\langle\boldsymbol{\phi}(\mathbf{z}_i)\cdot\boldsymbol{\gamma}\rangle) + \langle\boldsymbol{\gamma}\cdot\boldsymbol{\gamma}\rangle\\ &+\alpha_0^2\langle\boldsymbol{\phi}(\mathbf{0})\cdot\boldsymbol{\phi}(\mathbf{0})\rangle + 2\sum_{i=1}^{n}\alpha_i\alpha_0\langle\boldsymbol{\phi}(\mathbf{z}_i)\cdot\boldsymbol{\phi}(\mathbf{0})\rangle + 2\alpha_0\langle\boldsymbol{\gamma}\cdot\boldsymbol{\phi}(\mathbf{0})\rangle\Bigg),\end{aligned} \tag{A.5}$$

$$\begin{aligned}\sum_{i=1}^{n}\alpha_i[-\langle\mathbf{w}\cdot\boldsymbol{\phi}(\mathbf{z}_i)\rangle + \rho - \xi_i] =& -\sum_{i=1}^{n}\alpha_i\left\langle(\sum_{j=1}^{n}\alpha_j\boldsymbol{\phi}(\mathbf{z}_j) + \boldsymbol{\gamma} + \alpha_0\boldsymbol{\phi}(\mathbf{0}))\cdot\boldsymbol{\phi}(\mathbf{z}_i)\right\rangle\\ &+\sum_{i=1}^{n}\alpha_i\rho - \sum_{i=1}^{n}\alpha_i\xi_i =\\ &-\sum_{i=1}^{n}\alpha_i\sum_{i=j}^{n}\alpha_j\langle\boldsymbol{\phi}(\mathbf{z}_j)\cdot\boldsymbol{\phi}(\mathbf{z}_i)\rangle - \sum_{i=1}^{n}\alpha_i\langle\boldsymbol{\gamma}\cdot\boldsymbol{\phi}(\mathbf{z}_i)\rangle\\ &-\sum_{i=1}^{n}\alpha_i\alpha_0\,\langle\boldsymbol{\phi}(\mathbf{0})\cdot\boldsymbol{\phi}(\mathbf{z}_i)\rangle\\ &+\sum_{i=1}^{n}\alpha_i\rho - \sum_{i=1}^{n}\alpha_i\xi_i,\end{aligned} \tag{A.6}$$

and finally

$$\begin{aligned}-\sum_{i=1}^{m+s+h}\gamma_i w_i = -\langle\boldsymbol{\gamma}\cdot\mathbf{w}\rangle =& -\left\langle\boldsymbol{\gamma}\cdot\left(\sum_{i=1}^{n}\alpha_i\boldsymbol{\phi}(\mathbf{z}_i) + \boldsymbol{\gamma} + \alpha_0\boldsymbol{\phi}(\mathbf{0})\right)\right\rangle\\ =& -\sum_{i=1}^{m+s+h}\alpha_i\langle\boldsymbol{\gamma}\cdot\boldsymbol{\phi}(\mathbf{z}_i)\rangle - \langle\boldsymbol{\gamma}\cdot\boldsymbol{\gamma}\rangle - \alpha_0\langle\boldsymbol{\gamma}\cdot\boldsymbol{\phi}(\mathbf{0})\rangle.\end{aligned} \tag{A.7}$$

With respect to $\boldsymbol{\xi}$, using (A.2) we obtain $(n\frac{1}{\nu n} - \sum_{i=1}^{n}\alpha_i - \sum_{i=1}^{n}\beta_i)\boldsymbol{\xi} = \sum_{i=1}^{n}(\frac{1}{\nu n} - \alpha_i - \beta_i)\boldsymbol{\xi} = 0\boldsymbol{\xi}$. With $\rho$, using (A.3) we get $(-1 + \sum_{i=1}^{n}\alpha_i + \alpha_0)\rho = 0\rho$. Substituting the above expressions into $L$, we obtain

$$\begin{aligned}L(\mathbf{w},\boldsymbol{\xi},\rho,\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\gamma},\alpha_0) =& (\frac{1}{2} - 1)\sum_{i,j=1}^{n}\alpha_i\alpha_j\langle\boldsymbol{\phi}(\mathbf{z}_i)\cdot\boldsymbol{\phi}(\mathbf{z}_j)\rangle + (1 - 1 - 1)\sum_{i=1}^{n}\alpha_i\langle\boldsymbol{\phi}(\mathbf{z}_i)\cdot\boldsymbol{\gamma}\rangle\\ &+(\frac{1}{2} - 1)\langle\boldsymbol{\gamma}\cdot\boldsymbol{\gamma}\rangle + (\frac{1}{2} - 1)\alpha_0^2\langle\boldsymbol{\phi}(\mathbf{0})\cdot\boldsymbol{\phi}(\mathbf{0})\rangle + (1 - 1 - 1)\sum_{i=1}^{n}\alpha_i\alpha_0\,\langle\boldsymbol{\phi}(\mathbf{0})\cdot\boldsymbol{\phi}(\mathbf{z}_i)\rangle\\ &+(1 - 1 - 1)\alpha_0\,\langle\boldsymbol{\gamma}\cdot\boldsymbol{\phi}(\mathbf{0})\rangle = -\frac{1}{2}\left\|\mathbf{w}\right\|^2\end{aligned} \tag{A.8}$$

Hence, the **dual problem** to (5) becomes

$$\begin{array}{lll}\min\limits_{\boldsymbol{\alpha},\boldsymbol{\gamma},\alpha_0} & \frac{1}{2}\left\|\boldsymbol{w}\right\|^2 & \\ \text{subject to} & 0 \le \alpha_i \le 1/\nu n & \text{for } i \in \{1,\dots,n\},\\ & \sum_{i=1}^{n}\alpha_i + \alpha_0 = 1, & \\ & \boldsymbol{\gamma} \ge \mathbf{0}, & \end{array} \tag{A.9}$$

where $\mathbf{w} = \sum_{i=1}^{n}\alpha_i\boldsymbol{\phi}(\mathbf{z}_i) + \boldsymbol{\gamma} + \alpha_0\boldsymbol{\phi}(\mathbf{0})$ is as in (A.1).

## References

[1] Aparicio J, Pastor JT, Vidal F, Zofío JL. Evaluating productive performance: A new approach based on the product-mix problem consistent with data envelopment analysis. Omega 2017;67:134–44.

[2] Arnaboldi M, Azzone G, Giorgino M. Performance measurement and management for engineers. Academic Press; 2014.

[3] O'Donnell CJ, et al. Productivity and efficiency analysis. Singapore: Springer; 2018.

[4] Färe R, Primont D. Multi-output production and duality: theory and applications. Netherlands: Kluwer Academic Publishers; 1995, http://dx.doi.org/10.1007/978-94-011-0651-1.

[5] Cobb CW, Douglas PH. A theory of production. Am Econ Rev 1928;18(1):139–65.

[6] Aigner D, Lovell CAK, Schmidt P. Formulation and estimation of stochastic frontier production function models. J Econometrics 1977;6(1):21–37. http://dx.doi.org/10.1016/0304-4076(77)90052-5.

[7] Charnes A, Cooper WW, Rhodes E. Measuring the efficiency of decision making units. European J Oper Res 1978;2(6):429–44.

[8] Banker RD, Charnes A, Cooper WW. Some models for estimating technical and scale inefficiencies in data envelopment analysis. Manage Sci 1984;30(9):1078–92.

[9] Emrouznejad A, Yang G-l. A survey and analysis of the first 40 years of scholarly literature in DEA: 1978–2016. Soc-Econ Plan Sci 2018;61:4–8.

[10] Esteve M, Aparicio J, Rabasa A, Rodriguez-Sala JJ. Efficiency analysis trees: A new methodology for estimating production frontiers through decision trees. Expert Syst Appl 2020;162:113783.

[11] Valero-Carreras D, Aparicio J, Guerrero NM. Support vector frontiers: A new approach for estimating production functions through support vector machines. Omega 2021;104:102490. http://dx.doi.org/10.1016/j.omega.2021.102490.

[12] Simar L, Wilson PW. Sensitivity analysis of efficiency scores: How to bootstrap in nonparametric frontier models. Manage Sci 1998;44(1):49–61.

[13] Simar L, Wilson PW. A general methodology for bootstrapping in non-parametric frontier models. J Appl Stat 2000;27(6):779–802.

[14] Kuosmanen T, Johnson AL. Data envelopment analysis as nonparametric least-squares regression. Oper Res 2010;58(1):149–60.

[15] Parmeter CF, Racine JS. Smooth constrained frontier analysis. In: Chen X, Swanson NR, editors. Recent Advances and future directions in causality, prediction, and specification analysis: essays in honor of Halbert L. White Jr. New York, NY: Springer New York; 2013, p. 463–88. http://dx.doi.org/10.1007/978-1-4614-1653-1_18.

[16] Daouia A, Noh H, Park BU. Data envelope fitting with constrained polynomial splines. J R Stat Soc Ser B Stat Methodol 2016;78(1):3–30.

[17] Tsionas MG. Efficiency estimation using probabilistic regression trees with an application to Chilean manufacturing industries. Int J Prod Econ 2022;108492.

[18] Valero-Carreras D, Aparicio J, Guerrero NM. Multi-output support vector frontiers. Comput Oper Res 2022;143:105765.

[19] Guillen MD, Aparicio J, Esteve M. Gradient tree boosting and the estimation of production frontiers. Expert Syst Appl 2023;214:119134.

[20] Guerrero NM, Aparicio J, Valero-Carreras D. Combining data envelopment analysis and machine learning. Mathematics 2022;10(6):909.

[21] Olesen OB, Ruggiero J. An improved Afriat–Diewert–Parkan nonparametric production function estimator. European J Oper Res 2018;264(3):1172–88.

[22] Olesen O, Ruggiero J. The hinging hyperplanes: An alternative nonparametric representation of a production function. European J Oper Res 2022;296(1):254–66. http://dx.doi.org/10.1016/j.ejor.2021.03.054.

[23] Zhu Q, Wu J, Song M. Efficiency evaluation based on data envelopment analysis in the big data context. Comput Oper Res 2018;98:291–300.

[24] Zhu J. DEA under big data: Data enabled analytics and network data envelopment analysis. Ann Oper Res 2020;1–23.

[25] Borchani H, Varando G, Bielza C, Larranaga P. A survey on multi-output regression. Wiley Interdiscip Rev: Data Min Knowl Discov 2015;5(5):216–33.

[26] Parmeter CF, Sun K, Henderson DJ, Kumbhakar SC. Estimation and inference under economic restrictions. J Product Anal 2014;41(1):111–29.

[27] Daraio C, Simar L. Advanced robust and nonparametric methods in efficiency analysis: methodology and applications. Studies in productivity and efficiency, US: Springer; 2007, URL https://books.google.es/books?id=QAtGqmOwyIwC.

[28] Schölkopf B, Platt JC, Shawe-Taylor J, Smola AJ, Williamson RC. Estimating the support of a high-dimensional distribution. Neural Comput 2001;13(7):1443–71.

[29] Steinwart I, Hush D, Scovel C. A classification framework for anomaly detection. J Mach Learn Res 2005;6(2).

[30] De Vito E, Rosasco L, Toigo A. Learning sets with separating kernels. Appl Comput Harmon Anal 2014;37(2):185–217.

[31] Luenberger DG. New optimality principles for economic efficiency and equilibrium. J Optim Theory Appl 1992;75(2):221–64. http://dx.doi.org/10.1007/BF00941466.

[32] Cherchye L, De Rock B, Walheer B. Multi-output profit efficiency and directional distance functions. Omega 2016;61:100–9.

[33] Chambers RG, Färe R. Distance functions in production economics. In: Handbook of production economics. Springer; 2020, p. 1–35.

[34] Farrell MJ. The measurement of productive efficiency. J R Stat Soc A (General) 1957;120(3):253–90, URL http://www.jstor.org/stable/2343100.

[35] Cooper WW, Seiford LM, Tone K. Introduction to data envelopment analysis and its uses: with DEA-solver software and references. Springer Science & Business Media; 2006.

[36] Bogetoft P, Otto L. Data envelopment analysis DEA. In: Benchmarking with DEA, SFA, and R. Springer; 2011, p. 81–113.

[37] Luenberger DG. Benefit functions and duality. J Math Econ 1992;21(5):461–81. http://dx.doi.org/10.1016/0304-4068(92)90035-6.

[38] Chambers RG, Chung Y, Färe R. Profit, directional distance functions, and Nerlovian efficiency. J Optim Theory Appl 1998;98(2):351–64.

[39] Vapnik V. Statistical learning theory. A wiley-interscience publication, Wiley; 1998, URL https://books.google.es/books?id=GowoAQAAMAAJ.

[40] Vapnik V. The nature of statistical learning theory. Information science and statistics, Springer New York; 2013, URL https://books.google.es/books?id=EqgACAAAQBAJ.

[41] Breiman L. Hinging hyperplanes for regression, classification, and function approximation. IEEE Trans Inform Theory 1993;39(3):999–1013.

[42] Huang X, Mehrkanoon S, Suykens J. Support vector machines with piecewise linear feature mapping. Neurocomputing 2013;117:118–27. http://dx.doi.org/10.1016/j.neucom.2013.01.023.

[43] Pastor J, Lovell C, Aparicio J. Families of linear efficiency programs based on Debreu's loss function. J Product Anal 2012;38(2):109–20. http://dx.doi.org/10.1007/s11123-011-0216-4.

[44] Briec W. Hölder distance function and measurement of technical efficiency. J Product Anal 1999;11(2):111–31. http://dx.doi.org/10.1023/A:1007764912174.

[45] Coelli TJ, Rao DSP, O'Donnell CJ, Battese GE. An introduction to efficiency and productivity analysis. springer science & business media; 2005.

[46] Zhu N, Zhu C, Emrouznejad A. A combined machine learning algorithms and DEA method for measuring and predicting the efficiency of Chinese manufacturing listed companies. J Manag Sci Eng 2021;6(4):435–48.

[47] Hastie T, Tibshirani R, Friedman JH, Friedman JH. The elements of statistical learning: data mining, inference, and prediction. Vol. 2, Springer; 2009.