



Measuring technical efficiency for multi-input multi-output production processes through OneClass Support Vector Machines: a finite-sample study

Raul Moragues^{1,2} · Juan Aparicio^{1,3}  · Miriam Esteve¹

Received: 8 November 2022 / Revised: 4 May 2023 / Accepted: 14 June 2023 /

Published online: 6 July 2023

© The Author(s) 2023

Abstract

We introduce a new method for the estimation of production technologies in a multi-input multi-output context, based on OneClass Support Vector Machines with piecewise linear transformation mapping. We compare via a finite-sample simulation study the new technique with Data Envelopment Analysis (DEA) to estimate technical efficiency. The criteria adopted for measuring the performance of the estimators are bias and mean squared error. The simulations reveal that the approach based on machine learning seems to provide better results than DEA in our finite-sample scenarios. We also show how to adapt several well-known technical efficiency measures to the introduced estimator. Finally, we compare the new technique with respect to DEA via its application to an empirical database of USA schools from the Programme for International Student Assessment, where we obtain statistically significant differences in the efficiency scores determined through the Slacks-Based Measure.

Keywords Data envelopment analysis · OneClassSVM · Technical efficiency · Overfitting · Efficiency measures

✉ Juan Aparicio
j.aparicio@umh.es

Raul Moragues
rmoragues@umh.es

Miriam Esteve
miriam.estevec@umh.es

¹ Center of Operations Research (CIO), Miguel Hernandez University of Elche (UMH), 03202 Elche, Alicante, Spain

² PhD Program in Economics (DEcIDE), Elche, Spain

³ Valencian Graduate School and Research Network of Artificial Intelligence (valgrAI), Valencia, Spain

1 Introduction

Measuring the technical efficiency of firms and organizations in the context of production processes with multiple inputs and outputs is a problem that has received considerable attention in the specialized literature over the last decades [see, for example, the recent contributions by Aparicio et al. (2020), Ebrahimi et al. (2021), Liao et al. (2022)]. Since the seminal work by Farrell (1957), who dealt with single-output production frameworks, many methods have been proposed for this task, but only a few have received enough attention to develop into their own sub-areas of research. These methods include both non-parametric perspectives, such as Data Envelopment Analysis (DEA) (Charnes et al. 1978; Banker et al. 1984), as well as parametric approaches such as Stochastic Frontier Analysis (SFA) (Aigner et al. 1977; Meeusen and van Den Broeck 1977). Between these two approaches, the non-parametric perspective can be highlighted as being one of the most appealing techniques, due to its flexibility, the set of mild conditions required, and the natural way with which it deals with multi-input multi-output production contexts. This paper focuses on this type of sub-area of research.

In this context, the usual goal is to estimate the technical efficiency of a set of observed production units, usually called Decision Making Units (DMUs). This is often done by evaluating each unit against a set of feasible input–output bundles, also known as the production possibility set or technology, and considering how much each DMU can be modified before leaving this technology following certain improvement paths. In this framework, a certain part of the border of the technology, the so-called efficient frontier, plays a major role. In particular, a DMU is efficient when any permitted modification of the DMU would result in the DMU being projected outside the technology. In the case of inefficient DMUs, there are many possible directions in which a DMU can be projected towards the efficient frontier. Some of the approaches modify either only the inputs or only the outputs, leaving the rest unchanged. Amongst these, the first to be introduced were the Farrell radial input (output) measure, see Banker et al. (1984), where every input (output) is scaled by the same factor, thus keeping the mix of outputs (inputs) constant. Later, more general measures were introduced, such as the Russell measures (Färe and Lovell 1978), which allow the use of different scaling factors along each input (output). Furthermore, by considering the ratio between input and output Russell efficiencies, Pastor et al. (1999) and Tone (2001) introduced the Slacks-Based Measure (SBM), which satisfies additional desirable properties. Another approach considers directional distance functions (DDF), which chooses a directional vector and then projects each DMU in this direction (Luenberger 1992a, b; Chambers et al. 1998). Other alternative measures have been introduced over the last decades. For example, the Additive Model introduced in Charnes et al. (1985) is an alternative formulation of the radial and directional models which measures inefficiency via slacks in each input–output dimension. Through the weighting of each of the components, a Weighted Additive Model (WAM) was introduced in Lovell and Pastor (1995), which, through different choices of the weights, results in different technical inefficiency measures (see, e.g., Cooper et al. 1999).

Another attractive area when the focus is quantitatively analysing a data sample, and which had practically been growing separately from efficiency estimation, is that of Machine Learning (ML). In this area, certain algorithms are applied to determine a target function from available data. In our opinion, somehow, DEA can be seen as a ML technique where the target is the underlying technology that is responsible for generating the set of observations. Under the statistical approach in production theory postulated by (Daraio and Simar (2007), Chapter 3), the so-called technology coincides with the support of the joint probability distribution of inputs and outputs. In this way, determining this support leads to the identification of the technology and vice versa.

Amongst machine learning algorithms, a family which is widely used and has solid theoretical foundations is that of Support Vector Machines (SVM), introduced in Vapnik (1998), Vapnik (2013), which are based on the minimization of two types of errors, namely the empirical and the generalization errors, with a hyperparameter-led weighting between the two possibilities. Originally introduced for classification between two classes of labelled data, many adaptations and extensions to other contexts have been proposed. In particular, an adaptation of the classical SVM classifier to the context of unsupervised learning is the OneClass Support Vector Machine algorithm (OneClassSVM), introduced in Schölkopf et al. (2001), which takes the approach that an unlabelled dataset can be seen as a binary classification dataset where only examples from one of the two classes are available, and so translates the separation of the classes into a problem of estimating the support of the available data, and separating it from the rest of the coordinate space. Thus, this algorithm estimates the support of a joint probability distribution from which the available data are random samples. However, as we are aware, so far, this technique has not been adapted to identify technologies in production theory, i.e., input–output sets satisfying certain microeconomic axioms (such as convexity or free disposability), or to determine the usual technical efficiency measures (radial models, Russell measures, directional distance functions and so on). This will be the main objective of this paper, thus narrowing the existing gap between machine learning techniques and technical efficiency measurement from a non-parametric perspective.

In this paper, for the first time, we show how to determine a list of well-known technical efficiency measures as a result of tailoring the so-called OneClassSVM technique. As a way of gathering evidence towards the validity of this adaptation, we will compare our results with those obtained by the standard DEA methodology. We will do that through two different strategies. First of all, we will evaluate the goodness of our technology estimator by means of a simulation study based on finite samples, along the lines of papers such as Gong and Sickles (1992). Subsequently, we will illustrate how the new methodology performs through an empirical example and the Slacks-Based Measure. One of the tools which allows the standard SVM techniques to be powerful estimators is the introduction of a transformation function and associated kernel on the data, which embeds the data into a higher-dimensional vector space where the separation can be performed better. For this purpose, and taking the piecewise linear technology estimated by DEA as inspiration, we choose a piecewise linear (PWL) transformation function introduced in Huang et al. (2013), which we adapt to make the estimated technology satisfy the usual microeconomic

axioms in Production Theory (such as convexity and free disposability in inputs and outputs).

Overall, in this paper, we establish a new link between ML techniques and the measurement of technical efficiency, in the same line previously followed by authors such as Tsionas (2022), Esteve et al. (2020, 2022), Valero-Carreras et al. (2021, 2022), Olesen and Ruggiero (2018, 2022), Daouia et al. (2016), Parmeter and Racine (2013), and Liao et al. (2022). In particular, Valero-Carreras et al. (2021) and Valero-Carreras et al. (2022) introduced an adaptation of SVM for regression, that is, they tailored a supervised machine learning technique to estimate production frontiers. In contrast, in this paper, we focus our attention on the adaptation of One-ClassSVM, which is an unsupervised methodology. Other recent and related papers on ML and efficiency are Tsolas et al. (2020) and Thaker et al. (2022).

This paper follows the distinction between absolute and relative technical efficiency recently introduced by Aparicio and Esteve (2022). DEA identifies relative technical efficiency, that is, the degree of efficiency measured in comparative terms with respect to the performance of exactly the N observed units in the data sample. In contrast, absolute technical efficiency corresponds to the efficiency measured regarding the unknown Data Generating Process from which the data were drawn. In particular, the new method that we introduce attempts to measure absolute technical efficiency.

The remainder of the paper is structured as follows. Section 2 introduces the usual DEA context, with the netput notation that we will use throughout the paper, the list of most well-known efficiency measures as well as the fundamentals of the standard OneClassSVM algorithm. Section 3 develops the model that we propose, as well as two strategies to obtain the hyperparameters involved. Then, in Sect. 4, we adapt multiple measures of efficiency to the context of our algorithm and introduce the linear problems that will be solved to calculate the efficiency scores. Then, Sect. 5 contains some computational experiments that we use to evaluate the performance of our approach and compare it with traditional DEA under a finite-sample analysis. In Sect. 6, we apply the new approach to an empirical database from the literature, consisting of schools from the USA involved in the 2015 Programme for International Student Assessment (PISA) report, to illustrate and compare the new approach and DEA using, in particular, the Slacks-Based Measure. Finally, Sect. 7 contains the conclusions of this research and outlines several possible future research lines.

2 Background

2.1 Notation

In this section, we describe the notation that we use throughout this paper. We denote scalar variables in Roman letters, and in lowercase boldface letters when they are vectors. We denote the d -dimensional Euclidean space by \mathbb{R}^d , its non-negative (non-positive) orthant by \mathbb{R}_+^d (\mathbb{R}_-^d). A dataset \mathcal{Z} of DMUs contains N DMUs, where I is the number of inputs and O is the number of outputs. Usually, given a vector \mathbf{a} , we denote its j 'th component by a_j . However, in the

case of the DMUs, which we denote by $\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_N \in \mathbb{R}^{I+O}$, we indicate the components of these vectors using brackets, that is $\mathbf{z} = (z(1), \dots, z(I + O))$ and $\mathbf{z}_i = (z_i(1), \dots, z_i(I + O))$. We similarly denote other vectors indexed in two different ways, such as the coefficients of the hyperplanes in our programs, which are denoted by $\mathbf{p}_{I+O+1}, \dots, \mathbf{p}_{I+O+H}$, as well as the slacks vectors. In particular, the j 'th component of the k 'th hyperplane is $p_k(j)$. Bold numbers such as $\mathbf{0}, \mathbf{1}$ denote constant vectors with this number in every component. The dimension of these vectors is clear from the context in which they appear.

For operations between vectors, given $\mathbf{a} = (a_1, \dots, a_d), \mathbf{b} = (b_1, \dots, b_d) \in \mathbb{R}^d$, we use $\langle \mathbf{a} \cdot \mathbf{b} \rangle$ to denote their inner product, that is $\langle \mathbf{a} \cdot \mathbf{b} \rangle = \sum_{i=1}^d a_i b_i$. We denote the componentwise vector product, also called Hadamard product, by $\mathbf{a} \odot \mathbf{b} = (a_1 b_1, a_2 b_2, \dots, a_d b_d)$. Vector inequalities indicate that the specified inequality holds componentwise, e.g. $\mathbf{a} \geq \mathbf{b}$ indicates that $a_i \geq b_i$ for all $i = 1, \dots, d$. We remark that $\mathbf{a} > \mathbf{0}$ indicates that every component of \mathbf{a} is strictly positive, whereas $\mathbf{a} \geq \mathbf{0}$ and $\mathbf{a} \neq \mathbf{0}$ indicates that at least one component of \mathbf{a} is nonzero, with every component non-negative, but some components may be 0.

2.2 Data envelopment analysis

Given a set \mathcal{Z} of N Decision Making Units (DMUs) to be assessed, we denote the DMU k in *netput notation* by $\mathbf{z}_k = (z_k(1), \dots, z_k(I + O)) = (-\mathbf{x}_k, \mathbf{y}_k) \in \mathbb{R}_-^I \times \mathbb{R}_+^O$. This DMU uses $\mathbf{x}_k = (x_k(1), \dots, x_k(I)) > \mathbf{0}$ amounts of inputs and produces $\mathbf{y}_k = (y_k(1), \dots, y_k(O)) > \mathbf{0}$ amounts of outputs.

Given this dataset \mathcal{Z} , we have an underlying production process which we want to understand. In this context, the first goal is to estimate the technology or production possibility set, denoted by T , which consists of those pairs of input and output vectors that can be produced by the production process, i.e. the combinations of inputs and outputs which are feasible. In the netput notation,

$$T := \{\mathbf{z} \in \mathbb{R}_-^I \times \mathbb{R}_+^O : \mathbf{z} \text{ is feasible}\}. \tag{1}$$

In this context, DEA estimates a technology which coincides with the convex closure of the dataset, extended in the directions that are appropriate to satisfy the free disposability of inputs and outputs.

We denote this estimate technology by \hat{T}_{DEA} . This technology \hat{T}_{DEA} is the unique set satisfying the following properties, see Banker et al. (1984):

1. Envelopment: for all $\mathbf{z} \in \mathcal{Z}, \mathbf{z} \in \hat{T}_{\text{DEA}}$.
2. Convexity: for any $\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_k \in \hat{T}_{\text{DEA}}$, and any $\lambda \geq \mathbf{0}$, with $\sum_{i=1}^k \lambda_i = 1$, we have $\sum_{i=1}^k \lambda_i \mathbf{z}_i \in \hat{T}_{\text{DEA}}$.
3. Free disposability of inputs and outputs: for every $\mathbf{z} \in \hat{T}_{\text{DEA}}$ and $\mathbf{z}' \leq \mathbf{z}$, we have $\mathbf{z}' \in \hat{T}_{\text{DEA}}$.

4. Minimal extrapolation principle: \hat{T}_{DEA} is the smallest set satisfying Axioms (1), (2) and (3), that is, it is the intersection of all sets which satisfy Axioms (1), (2) and (3).

As such, the technology estimated by DEA is conservative, in the sense that it will fit perfectly to the data. From a machine learning point of view, this results in overfitting, and perhaps its associated relatively weak power of generalization to unseen data. This is one of the issues that we wish to address by considering DEA as a machine learning problem, that is, we want to estimate technologies that are closer to the theoretical, underlying technology, rather than to the particular dataset that we have available.

This way of thinking about the underlying Data Generating Process (DGP) was formalized by Daraio and Simar in (Daraio and Simar 2007, Chapter 3), and allows us to consider the estimation of the technology as a problem of estimating the support of the underlying DGP.

2.3 Measures of efficiency in data envelopment analysis

In the context of the measurement of technical efficiency, in particular when working with DEA and related techniques, technical efficiency measures how much a DMU can be modified while staying within the technology. This direction must be such that it reduces the inputs, increases the outputs, or a combination of both. In terms of netputs, it must either keep constant or increase every component of the DMU. The DMUs can be modified in many different ways while satisfying this condition and staying with the technology T , and we now introduce some of the methods described in the literature to project the DMUs in appropriate directions. We can obtain estimates of the technical efficiency by replacing the theoretical T by an appropriate estimator, such as \hat{T}_{DEA} .

Some of the first methods for measuring technical efficiency in the DEA context are the radial models, both input and output-oriented (Farrell 1957; Charnes et al. 1978; Banker et al. 1984). These are usually called the Farrell input (output) distances, and project along the inputs (outputs) by multiplying all of them by the same constant, that is, in a radial direction, while leaving the outputs (inputs) constant.

The *output-oriented Farrell distance* is:

$$F_{\text{out}}(\mathbf{z}) = \lambda(\mathbf{z}) = \max\{\lambda : (-\mathbf{x}, \lambda\mathbf{y}) \in T\}. \quad (2)$$

The *input-oriented Farrell distance* is:

$$F_{\text{in}}(\mathbf{z}) = \theta(\mathbf{z}) = \min\{\theta : (\theta(-\mathbf{x}), \mathbf{y}) \in T\}. \quad (3)$$

These measures are radial and scale every output (input) by the same constant. As such, they leave the relative proportions (mix) of outputs (inputs) constant. The range of values that they can obtain is, whenever $\mathbf{z} \in T$, $F_{\text{out}}(\mathbf{z}) \in [1, +\infty)$, whereas

$F_{in}(\mathbf{z}) \in (0, 1]$ where, in both cases, a value of 1 indicates technical efficiency. However, with the Farrell measures, a DMU may still have some room for improvement (slack) along some component of \mathbf{z} , and not others, which led to the introduction of other measures.

The so-called Russell measures are a generalization of the Farrell measures which allow for the rescaling of each input (output) by a different constant, thus allowing for the presence of slacks along only some of the inputs (outputs). They are non-radial measures of efficiency, introduced in Färe and Lovell (1978), (Färe et al. 1985, p.149), and are also called the Färe-Lovell efficiency indices in the literature.

The *Russell measure of output efficiency*, in the netput context, is defined as:

$$R_{out}(\mathbf{z}) = \max \left\{ \frac{1}{O} \sum_{r=I+1}^{I+O} \delta_r : \delta \odot \mathbf{z} \in T, \delta_r = 1 \text{ for } r \in \{1, \dots, I\} \text{ and } \delta_r \geq 1 \text{ for } r \in \{I+1, \dots, I+O\} \right\}. \tag{4}$$

Similarly, the *Russell measure of input efficiency* is defined as:

$$R_{in}(\mathbf{z}) = \min \left\{ \frac{1}{I} \sum_{r=1}^I \delta_r : \delta \odot \mathbf{z} \in T, \delta_r \in [0, 1] \text{ for } r \in \{1, \dots, I\} \text{ and } \delta_r = 1 \text{ for } r \in \{I+1, \dots, I+O\} \right\}. \tag{5}$$

We observe that the difference between the two orientations of the Russell measure are which coordinates are allowed to be rescaled (outputs or inputs), while the others (inputs or outputs) are kept constant, and that, in order to have the appropriate orientation, the ranges for these scaling factors differ. As a result, as written, whenever $\mathbf{z} \in T$, we have $R_{in}(\mathbf{z}) \in (0, 1]$, whereas $R_{out}(\mathbf{z}) \in [1, +\infty)$. In either case, a DMU is considered efficient when it attains a value of 1, which indicates that no improvement is possible along any input (output).

The Farrell measures are special cases of the corresponding Russell measures with the additional restriction that $\delta_1 = \delta_2 = \dots = \delta_I$ for the input orientation and $\delta_{I+1} = \delta_{I+2} = \dots = \delta_{I+O}$ for the output orientation.

Another non-radial measure of efficiency is the *Directional Distance Function* (DDF), considered in Luenberger (1992b), Luenberger (1992a), Chambers et al. (1998), Färe and Grosskopf (2000). In this model, the netput notation allows for a homogeneous treatment of both inputs and outputs. It projects each DMU along some pre-specified direction $\mathbf{g} \in \mathbb{R}_+^{I+O}$ such that $\mathbf{g} \neq \mathbf{0}$, as follows:

$$\delta(\mathbf{z}, \mathbf{g}) = \max\{\delta : (\mathbf{z} + \delta\mathbf{g}) \in T\}. \tag{6}$$

This is a measure of the distance from \mathbf{z} to the boundary of the technology along direction \mathbf{g} , as such, it takes the value 0 for efficient DMUs (any change along the \mathbf{g} direction leaves the technology), and, for $\mathbf{z} \in T$, we have $\delta(\mathbf{z}, \mathbf{g}) \geq 0$. Therefore, $\delta \in [0, +\infty)$. We remark that the Farrell measures are particular cases of the directional distance function where, for each \mathbf{z} , we take $\mathbf{g}(\mathbf{z}) = (-z(1), \dots, -z(I), 0, \dots, 0)$ for the input orientation and $\mathbf{g}(\mathbf{z}) = (0, \dots, 0, z(I+1), \dots, z(I+O))$ for the output orientation.

Additionally, another alternative is the additive-type measure. It, like the DDF, allows for changes in both inputs and outputs simultaneously. Furthermore, by assigning an independent slack to each variable via a slacks vector $\mathbf{s} \in \mathbb{R}_+^{I+O}$ (here all slacks are non-negative due to the netput notation) and weighting the penalties through a vector $\mathbf{v} \in \mathbb{R}_+^{I+O}$ with $\mathbf{v} > \mathbf{0}$, they allow for the detection of further inefficiencies along some of the directions. The basic formulation is the *Weighted Additive Model* (WAM, Lovell and Pastor 1995), and different choices of weights \mathbf{v} then result in different measures. This model also treats inputs and outputs in a homogeneous way due to the netput notation. The WAM formulation is:

$$WAM(\mathbf{z}, \mathbf{v}(\mathbf{z})) = \max \left\{ \sum_{j=1}^{I+O} v_j s(j) : \mathbf{z} + \mathbf{s} \in T, \mathbf{s} \geq \mathbf{0} \right\}. \tag{7}$$

Amongst the choices for weights, we consider the following:

1. Measure of Inefficiency Proportions (MIP, Cooper et al. 1999): $v_j = \frac{1}{|z(j)|}$.
2. Range Adjusted Measure (RAM, Cooper et al. 1999): $v_j = \frac{1}{(I+O)R_j}$, where R_j is the range of variable j , that is, $R_j = \max\{z_r(j) : r \in \{1, \dots, N\}\} - \min\{z_r(j) : r \in \{1, \dots, N\}\}$.

With these additive measures, as with the DDF, a DMU is considered to be efficient whenever its efficiency value is 0, and their range is $[0, +\infty)$. However, due to the variation in normalizing factors, their magnitudes are different and are not directly comparable.

We now introduce a *Slacks-Based Measure*, see Pastor et al. (1999), Tone (2001), also called Enhanced Russell Graph measure. This measure combines the ratio between the Russell input and Russell output measures. The SBM is based on an ordinary linear fractional programming model that can be linearized using a standard approach in the literature (see Charnes and Cooper 1962). The formulation which we take as a starting point for adaptation is the following additive model, that is Model (7) in Tone (2001):

$$SBM(\mathbf{z}) = \min \left\{ \left(1 + \frac{1}{I} \sum_{i=1}^I \frac{s(i)}{z(i)} \right) / \left(1 + \frac{1}{O} \sum_{i=I+1}^{I+O} \frac{s(i)}{z(i)} \right) : \mathbf{z} + \mathbf{s} \in T, \mathbf{s} \geq \mathbf{0} \right\}. \tag{8}$$

In this paper, we will tailor all these measures to the context of our ML model.

2.4 OneClass Support Vector Machines

We now introduce the unsupervised machine learning algorithm, called OneClass Support Vector Machine, sometimes abbreviated as OneClassSVM, 1CSVM or 1SVM, that we adapt for the estimation of production technologies. It was introduced in Schölkopf et al. (2001) as an adaptation of classification Support Vector

Machines (SVM) to the setting of estimating the support of a high-dimensional distribution, bringing us to the context of estimating a technology as the support of the probability distribution of inputs and outputs discussed above. An observation at the core of this approach is that estimating the support of a sample can be seen as a particular case of binary classification where only examples of one of the classes are available. The OneClassSVM quadratic program is:

$$\begin{aligned}
 & \min_{\mathbf{w} \in \mathbb{R}^{I+O+H}, \xi \in \mathbb{R}^N, \rho \in \mathbb{R}} \frac{1}{2} \|\mathbf{w}\|^2 + \frac{1}{vN} \sum_{i=1}^N \xi_i - \rho \\
 & \text{subject to} \quad \langle \mathbf{w} \cdot \boldsymbol{\phi}(\mathbf{z}_i) \rangle \geq \rho - \xi_i, \quad \forall i \in \{1, \dots, N\} \\
 & \quad \quad \quad \xi_i \geq 0, \quad \quad \quad \forall i \in \{1, \dots, N\}
 \end{aligned} \tag{9}$$

This program will obtain a solution $(\mathbf{w}^*, \xi^*, \rho^*)$, which defines the estimated support of the dataset, of the form:

$$\{\mathbf{z} \in \mathbb{R}^{I+O} : \langle \mathbf{w}^* \cdot \boldsymbol{\phi}(\mathbf{z}) \rangle \geq \rho^*\}. \tag{10}$$

This program involves a hyperparameter v determining how much weight is given to each component of the objective function and it also involves a transformation function $\boldsymbol{\phi} : \mathbb{R}^{I+O} \rightarrow \mathbb{R}^{I+O+H}$, that is, a transformation from the space of netputs into a higher dimensional real vector space that is key to determine the properties of the estimated set. We choose a transformation function based on the following piecewise linear transformation function (PWL) for our algorithm, which will result in a polyhedral set.

$$\boldsymbol{\phi}(\mathbf{z}) = \begin{cases} z(k), & \text{for } k \in \{1, \dots, I+O\} \\ \max\{0, \langle \mathbf{p}_k \cdot \mathbf{z} \rangle + q_k\}, & \text{for } k \in \{I+O+1, \dots, I+O+H\} \end{cases} \tag{11}$$

This piecewise linear transformation function was introduced and studied in [Huang et al. 2013, Expression (12)]. We choose it for its parallelism with DEA, which estimates technologies with a piecewise linear boundary, that is, a polyhedral set.

This function $\boldsymbol{\phi}$ involves a number H of hyperplanes, each defined by its slope vector $\mathbf{p}_k \in \mathbb{R}^{I+O}$, and its intercept $q_k \in \mathbb{R}$. These can be either considered hyperparameters of the model, which involves large amounts of computation to calculate the $H(I+O+1)$ hyperparameters required, or they can be chosen by a reasoned heuristic which is the approach that we take in this paper.

The role of the hyperplanes in the transformation is that the boundary of the estimated set will consist of flat portions until it reaches each of the hyperplanes where $\langle \mathbf{p}_k \cdot \mathbf{z} \rangle + q_k = 0$, at which point at least one of the hyperplane components will change the function chosen in the maximum function, thus allowing the boundary to change direction. Therefore, the hyperplanes will determine where the edges of the estimated polyhedral set are located.

3 The new approach

The model for the estimation of the technology that we use is the following adaptation of the OneClassSVM model (9) to the world of efficiency estimation.

$$\min_{\mathbf{w} \in \mathbb{R}^{I+O+H}, \xi \in \mathbb{R}^N, \rho \in \mathbb{R}} \frac{1}{2} \|\mathbf{w}\|^2 + \frac{1}{vN} \sum_{i=1}^N \xi_i - \rho \tag{12}$$

$$\text{subject to } \langle \mathbf{w} \cdot \boldsymbol{\phi}(\mathbf{z}_i) \rangle \geq \rho - \xi_i, \quad \forall i \in \{1, \dots, N\} \tag{12a}$$

$$\xi_i \geq 0, \quad \forall i \in \{1, \dots, N\} \tag{12b}$$

$$w_j \geq 0, \quad \forall j \in \{1, \dots, I + O + H\} \tag{12c}$$

The objective function of (12) and restrictions (12a) and (12b) are identical to the original OneClassSVM, while restriction (12c) guarantees convexity due to our choice of piecewise linear mapping (see Huang et al. 2013), which we use in the following adapted version:

$$\boldsymbol{\phi}(\mathbf{z}) = \begin{cases} -z(k), & \text{for } k \in \{1, \dots, I + O\} \\ -\max\{\mu, \langle \mathbf{p}_k \cdot \mathbf{z} \rangle + q_k\}, & \text{for } k \in \{I + O + 1, \dots, I + O + H\} \end{cases} \tag{13}$$

This transformation is changed from the formulation in (11) in two ways. The first adaptation is that we take the negatives of each component, so that the estimated region coincides with the area where the data lies, and the second modification replaces the 0 in the hyperplane components by a hyperparameter μ which we will tune during the training process. The effect of this change is that, instead of allowing the edges of the technology to be at the hyperplanes satisfying $\langle \mathbf{p}_k \cdot \mathbf{z} \rangle + q_k = 0$, now these edges will be located at the regions where $\langle \mathbf{p}_k \cdot \mathbf{z} \rangle + q_k = \mu$. By considering different values of μ , we allow these edges to be in slightly different regions, and the hyperparameter tuning will then compare them and select the value of μ which yields the best estimator. This will, therefore, enable the estimator to select amongst various candidate sets, and help reduce overfitting. We will show later in the paper how our estimator works through a computational experience.

Problem (12) is solved with given values for μ and v . This way, $(\mathbf{w}^*, \boldsymbol{\xi}^*, \rho^*)$ denotes an optimal solution of (12). From it, the technology estimated by program (12) is defined by:

$$\hat{T}(\mu, v) := \{\mathbf{z} \in \mathbb{R}_-^I \times \mathbb{R}_+^O : \langle \mathbf{w}^* \cdot \boldsymbol{\phi}(\mathbf{z}) \rangle \geq \rho^*\}.$$

The estimated technology $\hat{T}(\mu, v)$ then satisfies the following microeconomic axioms:

Proposition 3.1 *The following hold:*

1. (CONVEXITY) $\hat{T}(\mu, \nu)$ is convex.
2. (FREE DISPOSABILITY) Suppose that $\mathbf{p}_k \geq \mathbf{0}$ for all $k \in \{I + O + 1, \dots, I + O + H\}$. Then $\hat{T}(\mu, \nu)$ satisfies free disposability. In other words, if $\mathbf{z} \in \hat{T}(\mu, \nu)$ and $\mathbf{z}' \leq \mathbf{z}$ with $\mathbf{z}' \in \mathbb{R}_-^I \times \mathbb{R}_+^O$ then $\mathbf{z}' \in \hat{T}(\mu, \nu)$.
3. (ENVELOPMENT) The number of outliers (n_{OL}) is at most νN and the number of support vectors (n_{SV}) is at least νN . In other words,

$$n_{OL} \leq \nu N \leq n_{SV}.$$

Convexity of $\hat{T}(\mu, \nu)$ follows as in (Huang et al. (2013), Section 5), given that the defined ϕ is concave and $\mathbf{w} \geq \mathbf{0}$. Free disposability of inputs and outputs is satisfied as in CNLS [see (Kuosmanen and Johnson 2010), Section 2.2] when imposing additionally that $\mathbf{p}_k \geq \mathbf{0}$, so we will determine these parameters in a manner consistent with this choice. Finally, the bound on the fraction of outliers holds as in (Schölkopf et al. (2001), Proposition 3). The following corollary is a consequence of the principle of minimal extrapolation in DEA and Proposition 3.1(3). It states that the DEA estimator of the technology is always a subset of the estimator built from the adaptation of OneClassSVM.

Corollary 3.2 *If $\nu < 1/N$ then $\hat{T}_{DEA} \subseteq \hat{T}(\mu, \nu)$.*

Problem (12) with transformation (13) involves hyperparameters ν and μ , which we tune via a train-test split in order to obtain the best ones for each situation. In particular, by Proposition 3.1(3), ν can be seen to be a lower bound for the fraction of Support Vectors allowed, and an upper bound for the fraction of outliers. We choose ν in the range $[1/(N + 1), 0.1]$, except when $N \leq 10$, where we choose $\nu \in [0.1, 0.3]$. This results in a minimum of 0 outliers, and a maximum of 10% of DMUs being outliers. The values of μ depend on the hyperplanes chosen for ϕ , and we describe its role in the next Section.

3.1 Hyperplane parameters

The hyperplanes involved in the piecewise linear feature mapping that we use have a large impact on the performance of the estimator. The parameters \mathbf{p}_k, q_k and μ define, as in Huang et al. (2013), the regions where the boundary of the technology has turning points. As such, we are interested in using hyperplanes which lay between the dataset and the edge of the theoretical technology. Some examples of such hyperplanes are given by the convex closure of the data, in other words, the hyperplanes obtained by the DEA estimator. In order to obtain them, we solve the directional distance function (DDF) DEA program in its multiplier form, with directional vector $\mathbf{g} = \mathbf{1}$, corresponding to the Chebyshev norm

l_∞ , see Briec (1999). We solve the program for each $\mathbf{z}_i \in \mathcal{Z}$ to obtain its corresponding parameters \mathbf{p}_k, q_k :

$$\begin{aligned} \min_{\mathbf{p}_k, q_k} \quad & - \langle \mathbf{p}_k \cdot \mathbf{z}_i \rangle - q_k = -\mu_i \\ \text{subject to} \quad & \langle \mathbf{p}_k \cdot \mathbf{z}_r \rangle + q_k \leq 0, \quad \forall r \in \{1, \dots, N\} \\ & \langle \mathbf{p}_k \cdot \mathbf{1} \rangle = 1, \\ & \mathbf{p}_k \geq \mathbf{0} \end{aligned} \tag{14}$$

This is a netput-adapted form of the DDF program introduced in Equation (6) using the DEA-estimated technology, as found in (Pastor et al. (2012), Program 3). By solving this program, we ensure that the sum of the coefficients of the slope of each hyperplane add up to one, thus obtaining components in the transformation which have comparable magnitudes. Furthermore, they are non-negative, thus ensuring free disposability of the obtained estimators [see Proposition 3.1(2)].

Solving this problem for each DMU, we obtain N hyperplanes in the desired region. Furthermore, we obtain a set of values μ_i which determine the distance along directional vector $\mathbf{g} = \mathbf{1}$ of each DMU with respect to the DEA-estimated frontier, in other words, the convex closure of the dataset (extended by free disposability). This yields a minimum reasonable value for the offset hyperparameter μ , given by $\mu_{min} := \min_i \{\mu_i\} \leq 0$. When μ takes the value μ_{min} , the hyperplanes get offset so that every DMU is above at least one hyperplane and, if $\mu \leq \mu_{min}$, the hyperplanes are located in the region between the dataset and the origin, not enabling the frontier to have edges in the appropriate regions. Therefore, we choose the interval $[\mu_{min}, 0)$ as a suitable range of values for μ .

We now discuss two possible ways to define larger sets of hyperplanes for the PWL mapping: (1) the duplication of the hyperplane slopes with slightly modified offsets, and (2) the calculation of linear combinations of the existing hyperplanes, in particular via the mean.¹

3.1.1 Duplicate hyperplanes strategy

Following the process above, we obtained N hyperplanes with slopes \mathbf{p}_k . A way to obtain a larger number of hyperplanes is to duplicate the hyperplanes with the same slopes and with a small offset both upwards and downwards in order to obtain more flexibility in the estimated technologies. After some testing, we choose the offset for the duplicated hyperplanes to be $0.05R$, where R is the range of values taken by the data. This yields a value of $H = 3N$ for the number of hyperplane components of the PWL mapping.

Thus, in this case, we work with a set of slopes \mathbf{p}_k obtained by solving Problem (14) once for each DMU and, for each slope \mathbf{p}_k , we take the corresponding intercept

¹ We also considered other methods of obtaining the hyperplane parameters. Treating them as hyperparameters results in a large number $H(I + O + 1)$ of hyperparameters to tune, and so involves large computational expense without significant improvements. A grid of “flat” hyperplanes with $\mathbf{p}_k = (0, \dots, 1, \dots, 0)$ performed worse than the DEA hyperplanes alone, and took much smaller weights when considered together. Various other pre-set values for the slopes such as $\mathbf{p}_k = \mathbf{1}/(I + O)$ posed the same problem.

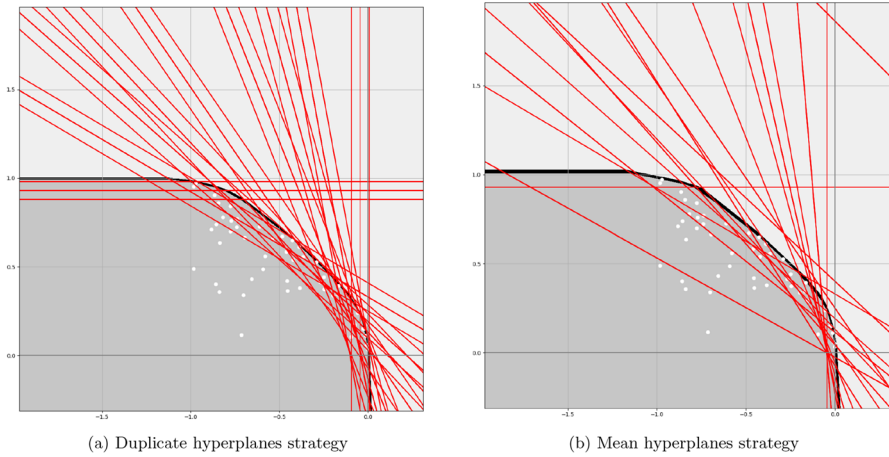


Fig. 1 Hyperplanes strategies

term q_k , and consider three hyperplanes defined by $(\mathbf{p}_k, q_k - 0.05R)$, (\mathbf{p}_k, q_k) , and $(\mathbf{p}_k, q_k + 0.05R)$. As such, the piecewise linear transformation has $I + O + H$ components, where $H = 3N$, which keeps a reasonable size. We provide an illustration of the types of hyperplanes thus obtained in Fig. 1a.

3.1.2 Mean hyperplanes strategy

Another approach we consider in order to obtain a higher number of hyperplanes in the transformation consists of, after solving Problem (14) for every DMU, the definition of new hyperplanes by taking linear combinations of the existing hyperplanes. We take the hyperplanes defined by the mean of two existing hyperplanes. In other words, given two hyperplanes (\mathbf{p}_k, q_k) , and (\mathbf{p}_l, q_l) , we define the mean hyperplane by $\mathbf{p}_{k,l} = (\mathbf{p}_k + \mathbf{p}_l)/2$, and $q_{k,l} = (q_k + q_l)/2$. This yields hyperplanes with varying slopes which still live in the appropriate region for the edges of the technology, and creates $N(N - 1)/2$ hyperplanes in the transformation. We illustrate this approach in Fig. 1b.

From our computational experience (Sect. 5), we conclude that the duplicate hyperplanes strategy is superior to the mean hyperplanes strategy and is less computationally expensive.

3.2 Technical inefficiency: the output-oriented directional distance function program

In order to evaluate and select the most appropriate technology estimator from the different choices of the hyperparameters μ and ν , we need a method to evaluate the different candidates and choose amongst them. We do so by calculating the Directional Distance Function (DDF) measure of inefficiency which, when we estimate a technology $\hat{T}(\mu, \nu)$, is defined by:

$$\begin{aligned} \hat{\delta}(\mathbf{z}, \mathbf{g}) &= \max\{\delta \in \mathbb{R} : (\mathbf{z} + \delta\mathbf{g}) \in \hat{T}(\mu, \nu)\} \\ &= \max\{\delta : \langle \mathbf{w}^* \cdot \boldsymbol{\phi}(\mathbf{z} + \delta\mathbf{g}) \rangle \geq \rho^*, \mathbf{z} + \delta\mathbf{g} \in \mathbb{R}_-^I \times \mathbb{R}_+^O\}. \end{aligned} \tag{15}$$

This requires the specification of a directional vector $\mathbf{g} \in \mathbb{R}_+^{I+O}$, with $\mathbf{g} \neq \mathbf{0}$. For this purpose, we choose the vector \mathbf{g}_i associated to the Farrell output distance for each DMU \mathbf{z}_i , that is: $\mathbf{g}_i = \mathbf{g}(\mathbf{z}_i) = (\mathbf{0}, \mathbf{y}_i) = (0, \dots, 0, z_i(I + 1), \dots, z_i(I + O))$. Furthermore, while Program (15) is not directly implementable by usual solvers, it can be rewritten as a standard linear program. We do this in Sect. 4.5, since the linearization process and associated proofs are analogous for the measures introduced in that Section. The linear program thus obtained is:

$$\begin{aligned} \max_{\delta \in \mathbb{R}, \sigma \in \mathbb{R}^H} \quad & M\delta - \sum_{j=I+O+1}^{I+O+H} \sigma_j \\ \text{subject to} \quad & - \sum_{j=1}^{I+O} w_j^*(z(j) + \delta g(j)) - \sum_{j=I+O+1}^{I+O+H} w_j^* \sigma_j \geq \rho^*, \\ & \sigma_j \geq \mu^*, \quad \forall j \in \{I + O + 1, \dots, I + O + H\} \cdot \\ & \sigma_j \geq \sum_{k=1}^{I+O} [p_j(k)(z(k) + \delta g(k))] + q_j, \quad \forall j \in \{I + O + 1, \dots, I + O + H\} \\ & z(j) + \delta g(j) \leq 0, \quad \forall j \in \{1, \dots, I\} \\ & z(j) + \delta g(j) \geq 0, \quad \forall j \in \{I + 1, \dots, I + O\} \end{aligned} \tag{16}$$

3.3 Description of the algorithm: tuning the hyperparameters

First, for each DMU $\mathbf{z}_i \in \mathcal{Z}$, we solve Program (14) to obtain N basic hyperplanes, with their appropriate slopes \mathbf{p}_k and intercepts q_k . We also define $\mu_{min} := \min_i\{\mu_i\}$. We then choose a strategy and use it to obtain a larger number H of hyperplanes and their corresponding values \mathbf{p}_k, q_k which we use to define the transformation function $\boldsymbol{\phi}$. At this stage, we have Program (12) ready to be solved for each choice of μ, ν .

The hyperparameters that remain to tune in the algorithm are μ and ν . Unless otherwise specified, we choose 5 values equally spaced in the intervals $[\mu_{min}, 0)$ for μ , and we choose ν in the range $[1/N, 0.1]$, except when $N \leq 10$, where we choose $\nu \in [0.1, 0.3]$.

In order to choose amongst these candidate values, we randomly split the dataset \mathcal{Z} into a training set \mathcal{Z}_{train} containing 70% of the DMUs and a test set \mathcal{Z}_{test} containing the remaining 30% of the data.

For each candidate pair of values of μ, ν , we train the model by solving (12) on the training set, obtaining a candidate estimator $\hat{T}(\mu, \nu)$.

We then evaluate the performance of each estimator $\hat{T}(\mu, \nu)$ on the test set \mathcal{Z}_{test} by comparing, on each DMU $\mathbf{z}_i \in \mathcal{Z}_{test}$, the predicted projections $\mathbf{z}_i + \delta\mathbf{g}_i$ of each DMU according to Program (16). This yields estimated output levels for \mathbf{z}_i , which we use for each $\mathbf{z}_i \in \mathcal{Z}_{test}$ in order to calculate the Mean Squared Error (MSE) associated with $\hat{T}(\mu, \nu)$. We then choose as the best hyperparameters those (μ^*, ν^*) that minimize this MSE on \mathcal{Z}_{test} .

Finally, once the best hyperparameters (μ^*, v^*) are determined, the model is retrained on the whole dataset \mathcal{Z} , by solving Program (12) on \mathcal{Z} with the chosen hyperparameters, yielding the final estimate of the technology $\hat{T}(\mu^*, v^*)$. We refer to the method by $1SVM_d$ when the duplicate hyperplanes strategy is used and by $1SVM_m$ when the mean hyperplanes strategy is employed.

4 Measures of efficiency

In this section, we introduce the optimization problems that we will use to obtain the various efficiency scores with respect to the estimator introduced in this paper, we linearize them, and show equivalence of the solutions to both problems. We begin with the Russell measures, as the proofs used in this case can be adapted in a straightforward manner to the other measures of efficiency.

We remark that, depending on the value of v^* , this method may leave some DMUs as outliers, and some of the ways to measure efficiency that we present will have infeasible problems in this case. This can be avoided by setting the hyperparameter $0 < v^* < 1/N$, where no outliers will be permitted, and thus this issue will not arise.

Similarly, we recall that we assume that the DMUs do not have any 0 values in their inputs or outputs. Otherwise, minor adjustments to the problems must be made to avoid issues of unboundedness or null denominators.

For every measure, we first obtain the estimated technology $\hat{T}(\mu^*, v^*)$, which we recall is defined by taking $(\mathbf{w}^*, \xi^*, \rho^*)$ to be an optimal solution of (12) after the tuning of the hyperparameters μ^*, v^* , which is given by $\hat{T}(\mu^*, v^*) = \{z \in \mathbb{R}^I_- \times \mathbb{R}^O_+ : \langle \mathbf{w}^* \cdot \phi(\mathbf{z}) \rangle \geq \rho^*\}$. Therefore, these values are fixed and are not variables of the presented programs.

4.1 Russell output

The Russell input and output measures of efficiency are non-radial measures of efficiency, and we consider them first. They can be seen as generalizations of the corresponding Farrell input or output measures which, instead of scaling every variable by the same scalar, allow that there can be slacks along some of the inputs (outputs). Thus, the arguments for these measures will also apply to the Farrell measures. We begin by introducing the output-oriented Russell measure.

The *Russell measure of output efficiency* is defined as:

$$\begin{aligned}
 R_{out}(\mathbf{z}) &= \max \left\{ \frac{1}{O} \sum_{r=I+1}^{I+O} \delta_r \cdot \delta \odot \mathbf{z} \in \hat{T}(\mu^*, v^*), \delta_r = 1 \forall r \in \{1, \dots, I\}, \delta_r \geq 1 \forall r \in \{I+1, \dots, I+O\} \right\} \\
 &= \max \left\{ \frac{1}{O} \sum_{r=I+1}^{I+O} \delta_r \cdot \langle \mathbf{w}^* \cdot \phi(\delta \odot \mathbf{z}) \rangle \geq \rho^*, \delta_r = 1 \forall r \in \{1, \dots, I\}, \delta_r \geq 1 \forall r \in \{I+1, \dots, I+O\} \right\}.
 \end{aligned}
 \tag{17}$$

Given $\mathbf{z} \in \hat{T}(\mu^*, v^*)$, the output-oriented Russell measure takes values $R_{out}(\mathbf{z}) \geq 1$, with \mathbf{z} being efficient whenever $R_{out}(\mathbf{z}) = 1$. Due to the definition of the transformation function ϕ , which has nonlinear components involving a maximum

function, program (17) is not linear. However, we can linearize it by adding a new variable $\sigma \in \mathbb{R}^H$ which attains the value of the maximum at each component, $\sigma_j = \max\{\mu^*, \langle \mathbf{p}_j \cdot \mathbf{z} \rangle + q_j\}$. In order to force σ to attain the maximum at each component, we penalize this value in the objective function, and we introduce a constant M large enough so that small changes in δ affect the objective function more than the corresponding changes in σ . The linearized model for the output-oriented Russell measure is:

$$\begin{aligned}
 & \max_{\delta \in \mathbb{R}^{I+O}, \sigma \in \mathbb{R}^H} M \frac{1}{O} \sum_{r=I+1}^{I+O} \delta_r - \sum_{j=I+O+1}^{I+O+H} \sigma_j \\
 \text{subject to} & \quad - \sum_{j=1}^{I+O} w_j^* \delta_j z(j) - \sum_{j=I+O+1}^{I+O+H} w_j^* \sigma_j \geq \rho^*, \\
 & \quad \sigma_j \geq \mu^*, \quad \forall j \in \{I+O+1, \dots, I+O+H\} \cdot \\
 & \quad \sigma_j \geq \sum_{k=1}^{I+O} [p_j(k) \delta_k z(k)] + q_j, \quad \forall j \in \{I+O+1, \dots, I+O+H\} \\
 & \quad \delta_r = 1, \quad \forall r \in \{1, \dots, I\} \\
 & \quad \delta_r \geq 1, \quad \forall r \in \{I+1, \dots, I+O\}
 \end{aligned} \tag{18}$$

Program (18) is a linear program, where M is a large number. We now prove that a solution to the linearized program yields a solution to the original definition of the measure. We first prove the following auxiliary result that, in an optimal solution, the linearizing variable σ which we introduce as proxy for the terms involving the maximum of two numbers indeed attains this maximum value. In other words, at least one of the second and third restrictions becomes equality at an optimal solution.

Lemma 4.1 *Let (δ^*, σ^*) be an optimal solution of (18). Then, $\sigma_j^* = \max\{\mu^*, \langle \mathbf{p}_j \cdot (\delta^* \odot \mathbf{z}) \rangle + q_j\}$ for all $j \in \{I+O+1, \dots, I+O+H\}$.*

Proof Suppose (δ^*, σ^*) is an optimal solution to (18). By the second and third restrictions of (18), for each $j \in \{I+O+1, \dots, I+O+H\}$, we have $\sigma_j^* \geq \max\{\mu^*, \langle \mathbf{p}_j \cdot (\delta^* \odot \mathbf{z}) \rangle + q_j\}$. Now, assume that $\sigma_{j'}^* > \max\{\mu^*, \langle \mathbf{p}_{j'} \cdot (\delta^* \odot \mathbf{z}) \rangle + q_{j'}\}$ for some $j' \in \{I+O+1, \dots, I+O+H\}$. Then, we consider the potential solution (δ^*, σ') where $\sigma'_l = \sigma_l^*$ for $l \neq j'$ and $\sigma'_{j'} = \max\{\mu^*, \langle \mathbf{p}_{j'} \cdot (\delta^* \odot \mathbf{z}) \rangle + q_{j'}\} < \sigma_{j'}^*$. This is still a feasible point of (18), as the LHS of the first restriction becomes greater, and the last two restrictions are still satisfied, so it is still a solution to the optimization problem with a larger objective, contradicting the assumption that (δ^*, σ^*) was optimal. \square

We can now prove the following link between the solutions of both programs.

Proposition 4.2 *Whenever (δ^*, σ^*) is an optimal solution of (18), then δ^* is an optimal solution of Program (17).*

Proof Let (δ^*, σ^*) be an optimal solution of (18) and suppose that δ^* is not an optimal solution of (17). Then, there exists δ' satisfying $\frac{1}{O} \sum_{r=I+1}^{I+O} \delta_{r'} > \frac{1}{O} \sum_{r=I+1}^{I+O} \delta_r^*$ such that δ' is a feasible solution of (17). That is, $\delta' \odot \mathbf{z} \in \hat{T}(\mu^*, \nu^*)$ and, by definition of $\hat{T}(\mu^*, \nu^*)$, $\langle \mathbf{w}^* \cdot \phi(\delta' \odot \mathbf{z}) \rangle \geq \rho^*$.

Define σ' by $\sigma'_j = \max\{\mu^*, \langle \mathbf{p}_j \cdot (\delta' \odot \mathbf{z}) \rangle + q_j\}$ for all $j \in \{I + O + 1, \dots, I + O + H\}$. Then, (δ', σ') is a feasible solution of (18), since the first constraint of (18) is satisfied because:

$$-\sum_{j=1}^{I+O} w_j^* \delta_{jz}(j) - \sum_{j=I+O+1}^{I+O+H} w_j^* \sigma'_j = \langle \mathbf{w}^* \cdot \phi(\delta' \odot \mathbf{z}) \rangle \geq \rho^*,$$

the second and third constraints of (18) are directly met by definition of σ' , and the last two constraints of (18) are satisfied because δ' is a feasible solution of (17). Since $\frac{1}{O} \sum_{r=I+1}^{I+O} \delta_{r'} > \frac{1}{O} \sum_{r=I+1}^{I+O} \delta_r^*$ and M is large enough (to offset the effect of the change in σ'_j), we have:

$$M \frac{1}{O} \sum_{r=I+1}^{I+O} \delta'_{r'} - \sum_{j=I+O+1}^{I+O+H} \sigma'_j > M \frac{1}{O} \sum_{r=I+1}^{I+O} \delta_r^* - \sum_{j=I+O+1}^{I+O+H} \sigma_j^*,$$

hence (δ^*, σ^*) is not an optimal solution of (18), contradicting our assumption. Thus, whenever (δ^*, σ^*) is an optimal solution of (18), δ^* is an optimal solution of (17). □

Thus, we can solve the linear program (18) to obtain the output-oriented Russell measure scores when the underlying technology is estimated by our proposal based on OneClasSVM. Next, we similarly linearize the other programs, and the same proof, adapted, holds true for the other measures.

4.2 Russell input

The Russell measure of input efficiency, analogously to the output one, is defined as:

$$R_{in}(\mathbf{z}) = \min \left\{ \frac{1}{I} \sum_{r=1}^I \delta_r : \delta \odot \mathbf{z} \in \hat{T}(\mu^*, \nu^*), \delta_r \in [0, 1] \forall r \in \{1, \dots, I\}, \delta_r = 1 \forall r \in \{I + 1, \dots, I + O\} \right\}. \tag{19}$$

In this case, $0 \leq R_{in}(\mathbf{z}) \leq 1$, and a DMU is efficient when $R_{in}(\mathbf{z}) = 1$, which happens when $\delta = \mathbf{1}$; in other words, when we cannot decrease any input without making the DMU infeasible. That is, any decrease in any input would result in an infeasible DMU.

As before with the output Russell distance, we can linearize this problem. The arguments above hold mutatis mutandis, as the only changes are the change of maximizing to minimizing in the objective function (hence the penalization term

for σ has different sign), the δ that appears in the objective function, and the range of possible values for δ , whereas everything else stays the same:

$$\begin{aligned}
 & \min_{\delta \in \mathbb{R}^{I+O}, \sigma \in \mathbb{R}^H} M \frac{1}{I} \sum_{r=1}^I \delta_r + \sum_{j=I+O+1}^{I+O+H} \sigma_j \\
 \text{subject to} \quad & - \sum_{j=1}^{I+O} w_j^* \delta_j z(j) - \sum_{j=I+O+1}^{I+O+H} w_j^* \sigma_j \geq \rho^*, \\
 & \sigma_j \geq \mu^*, \quad \forall j \in \{I+O+1, \dots, I+O+H\} \cdot \\
 & \sigma_j \geq \sum_{k=1}^{I+O} [p_j(k) \delta_k z(k)] + q_j, \quad \forall j \in \{I+O+1, \dots, I+O+H\} \\
 & 0 \leq \delta_r \leq 1, \quad \forall r \in \{1, \dots, I\} \\
 & \delta_r = 1, \quad \forall r \in \{I+1, \dots, I+O\}
 \end{aligned} \tag{20}$$

Thus, we obtain the following result:

Proposition 4.3 *If (δ^*, σ^*) is an optimal solution of (20) then δ^* is an optimal solution of Program (19).*

Proof The proof in Proposition 4.2 holds with the appropriate changes. □

4.3 Farrell output

The *output-oriented radial measure* or output-oriented Farrell measure for our technology estimator is:

$$\begin{aligned}
 F_{out}(\mathbf{z}) &= \max \{ \lambda : (-\mathbf{x}, \lambda \mathbf{y}) \in \hat{T}(\mu^*, v^*) \} = \max \{ \lambda : (z(1), \\
 & \dots, z(I), \lambda z(I+1), \lambda z(I+2), \dots, \lambda z(I+O)) \in \hat{T}(\mu^*, v^*) \} \\
 &= \max \{ \lambda : \langle \mathbf{w}^* \cdot \phi(\mathbf{z} + (\lambda - 1)(\mathbf{0}, \mathbf{y})) \rangle \geq \rho^*, (\mathbf{z} + (\lambda - 1)(\mathbf{0}, \mathbf{y})) \in \mathbb{R}_-^I \times \mathbb{R}_+^O \} \\
 &= \max \{ \delta : \langle \mathbf{w}^* \cdot \phi(\mathbf{z} + \delta(\mathbf{x}, \mathbf{0})) \rangle \geq \rho^*, (\mathbf{z} + \delta(\mathbf{x}, \mathbf{y})) \in \mathbb{R}_-^I \times \mathbb{R}_+^O \} + 1.
 \end{aligned} \tag{21}$$

Regarding the various equalities, the first formulation is the definition of the Farrell output distance, the second expression expands the values in the vector involved, implicitly showing that this is a particular case of the Russell output measure. Finally, the third and fourth formulations show how, via the transformation $\delta = \lambda - 1$, this can be seen as a particular case of the DDF with directional vector $\mathbf{g}(\mathbf{z}) = (\mathbf{0}, \mathbf{y})$.

The Farrell output distance, given $\mathbf{z} \in \hat{T}(\mu^*, v^*)$, satisfies $\lambda \geq 1$, with $\lambda = 1$ whenever \mathbf{z} is efficient. However, in case of outliers, this value $\lambda = 1$ does not

yield a feasible solution, and so, to prevent leaving the quadrant, we will add a restriction of the form $\lambda \geq 0$ to the corresponding linearized program.

As before, program (21) is not linear, due to the dependence on a maximum in the definition of $\phi(\mathbf{z})$, but we can linearize it by using the same big-M technique as before. The resulting linear program, with variables λ, σ , is:

$$\begin{aligned}
 & \max_{\lambda \in \mathbb{R}, \sigma \in \mathbb{R}^{I+O+H}} M\lambda - \sum_{j=I+O+1}^{I+O+H} \sigma_j \\
 & \text{subject to } - \sum_{j=1}^I w_j^* z(j) - \lambda \sum_{j=I+1}^{I+O} w_j^* z(j) - \sum_{j=I+O+1}^{I+O+H} w_j^* \sigma_j \geq \rho^*, \\
 & \quad \sigma_j \geq \mu^*, \quad \forall j \in \{I+O+1, \dots, I+O+H\} \\
 & \quad \sigma_j \geq \sum_{k=1}^I p_j(k) z(k) + \lambda \left[\sum_{k=I+1}^{I+O} p_j(k) z(k) \right] + q_j, \quad \forall j \in \{I+O+1, \dots, I+O+H\} \\
 & \quad \lambda \geq 0
 \end{aligned} \tag{22}$$

As before, we can obtain an optimal solution to (21) from one of (22):

Proposition 4.4 *If (λ^*, σ^*) is an optimal solution of (22) then λ^* is an optimal solution of Program (21).*

Proof The Farrell output measure is a special case of the output-oriented Russell measure, so this statement is a particular case of Proposition 4.2. □

4.4 Farrell input

The input-oriented Farrell measure of efficiency is analogous to the Farrell output measure, but with the scaling factor on the inputs instead of the outputs. In our setting, it is defined by:

$$\begin{aligned}
 F_{in}(\mathbf{z}) &= \min\{\theta: \langle \theta(-\mathbf{x}), \mathbf{y} \rangle \in \hat{T}(\mu^*, \nu^*)\} = \min\{\theta: \\
 & (\theta z(1), \dots, \theta z(I), z(I+1), \dots, z(I+O)) \in \hat{T}(\mu^*, \nu^*)\} \\
 & = \min\{\theta: \langle \mathbf{w}^* \cdot \phi(\mathbf{z} + (1-\theta)(\mathbf{x}, \mathbf{0})) \rangle \geq \rho^*, (\mathbf{z} + (1-\theta)(\mathbf{x}, \mathbf{0})) \in \mathbb{R}_-^I \times \mathbb{R}_+^O\} \\
 & = 1 - \max\{\delta: \langle \mathbf{w}^* \cdot \phi(\mathbf{z} + \delta(\mathbf{x}, \mathbf{0})) \rangle \geq \rho^*, (\mathbf{z} + \delta(\mathbf{x}, \mathbf{0})) \in \mathbb{R}_-^I \times \mathbb{R}_+^O\}.
 \end{aligned} \tag{23}$$

In this case, $F_{in}(\mathbf{z}) \in (0, 1]$ whenever $\mathbf{z} \in \hat{T}(\mu^*, \nu^*)$, with $F_{in} = 1$ whenever \mathbf{z} is efficient. The last equality in (23) shows that the Farrell input distance can be seen as a special case of the Directional Distance Function with $\mathbf{g} = (\mathbf{x}, \mathbf{0}) = (-z(1), \dots, -z(I), 0, \dots, 0)$. Note that, in this latter case, the problem is to minimize, which is due to the relationship $\delta = 1 - \theta$, which inverts the goal of the problem.

In the case of this distance, we can linearize the nonlinear problem above as in the output case, to obtain the following linear program:

$$\begin{aligned}
 & \min_{\theta \in \mathbb{R}, \sigma \in \mathbb{R}^{I+O+H}} M\theta + \sum_{j=I+O+1}^{I+O+H} \sigma_j \\
 & \text{subject to } -\theta \sum_{j=1}^I w_j^* z(j) - \sum_{j=I+1}^{I+O} w_j^* z(j) - \sum_{j=I+O+1}^{I+O+H} w_j^* \sigma_j \geq \rho^*, \\
 & \quad \sigma_j \geq \mu^*, \quad \forall j \in \{I+O+1, \dots, I+O+H\} \\
 & \quad \sigma_j \geq \theta \left[\sum_{k=1}^I p_j(k) z(k) \right] + \sum_{k=I+1}^{I+O} p_j(k) z(k) + q_j, \quad \forall j \in \{I+O+1, \dots, I+O+H\} \\
 & \quad \theta \geq 0
 \end{aligned} \tag{24}$$

As before, we have the following relationship between the solutions to these programs:

Proposition 4.5 *If (θ^*, σ^*) is an optimal solution of (24) then θ^* is an optimal solution of Program (23).*

Proof This is a special case of the input-oriented Russell function, so it is a particular case of Proposition 4.3. □

4.5 Directional distance function

We now consider the directional distance function (DDF), as described in Sects. 2.3 and 3.2. This is a measure of inefficiency, which measures how much \mathbf{z} can be moved along a direction $\mathbf{g} \in \mathbb{R}_+^{I+O}$ before leaving the technology. In order to obtain the inefficiency of a DMU \mathbf{z} , we solve Problem (15) in its linearized form (16).

The DDF takes values where efficient DMUs have inefficiency score $\delta = 0$, and DMUs within the technology have $\delta \geq 0$. Furthermore, the DDF also allows for those DMUs outside the technology which can be projected along \mathbf{g} into the technology to obtain inefficiency scores, but in these cases $\delta < 0$. Then, the following holds, with proof analogous to that of Proposition 4.2:

Proposition 4.6 *If (δ^*, σ^*) is an optimal solution of (16) then δ^* is an optimal solution of Program (15).*

Proof Analogous to Proposition 4.2. □

4.6 Weighted additive model

Another well-known family of measures of efficiency consists of the measures based on the Weighted Additive Model (WAM), introduced in Lovell and Pastor (1995). Through various choices of weights, a variety of measures is defined such as the Measure of Inefficiency Proportions (MIP) and Range Adjusted Measure (RAM), see Cooper et al. (1999).

Regarding notation, since the slacks vector is going to depend on the corresponding DMU, we denote it by $\mathbf{s} = (s(1), \dots, s(I+O))$, so that, when necessary, we can refer to the slacks of DMU \mathbf{z}_i by \mathbf{s}_i . This measure allows for slacks in both the inputs and the

outputs. The Weighted Additive Model in the netput notation and with respect to the estimator introduced in this article is:

$$\begin{aligned}
 WAM(\mathbf{z}, \mathbf{v}(\mathbf{z})) &= \max \left\{ \sum_{j=1}^{I+O} v_j s(j) : \mathbf{z} + \mathbf{s} \in \hat{T}(\mu^*, \mathbf{v}^*), \mathbf{s} \geq \mathbf{0} \right\} \\
 &= \max \left\{ \sum_{j=1}^{I+O} v_j s(j) : \langle \mathbf{w}^* \cdot \boldsymbol{\phi}(\mathbf{z} + \mathbf{s}) \rangle \geq \rho^*, \mathbf{s} \geq \mathbf{0}, \mathbf{z} + \mathbf{s} \in \mathbb{R}_-^I \times \mathbb{R}_+^O \right\}.
 \end{aligned}
 \tag{25}$$

We remark that, as with the property of free disposability, the change of signs of the inputs in the netput notation also allows a homogeneous treatment of the slacks, without the need to split the sum into its input and output terms. In this model, we require that the weights $\mathbf{v} > \mathbf{0}$ are strictly positive.

We linearize Program (25) with the same technique as before. Hence, once the weights are chosen, the linearized model for $WAM(\mathbf{z})$ becomes:

$$\begin{aligned}
 &\max_{\mathbf{s} \in \mathbb{R}_+^{I+O}, \boldsymbol{\sigma} \in \mathbb{R}^H} M \sum_{j=1}^{I+O} v_j s(j) - \sum_{j=I+O+1}^{I+O+H} \sigma_j \\
 \text{subject to} & \quad - \sum_{j=1}^{I+O} w_j^* (z(j) + s(j)) - \sum_{j=I+O+1}^{I+O+H} w_j^* \sigma_j \geq \rho^*, \\
 &\sigma_j \geq \mu^*, \quad \forall j \in \{I+O+1, \dots, I+O+H\}, \\
 &\sigma_j \geq \sum_{k=1}^{I+O} [p_j(k)(z(k) + s(k))] + q_j, \quad \forall j \in \{I+O+1, \dots, I+O+H\} \\
 &\mathbf{s} \geq \mathbf{0}, \\
 &z(j) + s(j) \leq 0, \quad \forall j \in \{1, \dots, I\} \\
 &z(j) + s(j) \geq 0, \quad \forall j \in \{I+1, \dots, I+O\}
 \end{aligned}
 \tag{26}$$

We remark that the last constraint in Model (26) is redundant whenever $\mathbf{z} \in \mathbb{R}_-^I \times \mathbb{R}_+^O$, since $\mathbf{s} \geq \mathbf{0}$, but we include it for parallelism with the other programs. The same proofs as before also work here to prove the following relationship between solutions to both programs.

Proposition 4.7 *If $(\mathbf{s}^*, \boldsymbol{\sigma}^*)$ is an optimal solution of (26) then \mathbf{s}^* is an optimal solution of Program (25).*

4.7 Slacks-based measure

The last measure that we adapt to the context of our estimator is the Slacks-Based Measure (SBM), see Pastor et al. (1999), Tone (2001). Denominated Enhanced Russell Graph Measure in Pastor et al. (1999), its original formulation has fractional terms in the objective, and we adapt the original linearization procedure to our context. We take the formulation in terms of the additive model as a starting point, i.e.

model (7) in Tone (2001), where the ratio terms appear in terms involving the input and output slacks.

The fractional program to be solved with respect to the estimator, in the netput notation, can be expressed as:

$$\begin{aligned} \min_{\mathbf{s} \in \mathbb{R}^{I+O}} \quad & \frac{1 + \frac{1}{I} \sum_{i=1}^I \frac{s(i)}{z(i)}}{1 + \frac{1}{O} \sum_{i=I+1}^{I+O} \frac{s(i)}{z(i)}} \\ \text{subject to} \quad & \mathbf{z} + \mathbf{s} \in \hat{T}(\mu^*, \nu^*), \\ & \mathbf{s} \geq \mathbf{0} \end{aligned} \tag{27}$$

In order to linearize this program, we first use the following substitution of variables, following Charnes and Cooper (1962). This change of variables is also mentioned in Cooper et al. (2006) as an exercise:

$$\beta = \left(1 + \frac{1}{O} \sum_{i=I+1}^{I+O} \frac{s(i)}{z(i)} \right)^{-1}, \tag{28}$$

$$\mathbf{t} = \beta \mathbf{s}. \tag{29}$$

We remark that $\beta > 0$. With these variables in mind, we notice that $1/\beta$ is the denominator of the objective function of (27), so that this objective function can be rewritten as $\beta \left(1 + \frac{1}{I} \sum_{i=1}^I \frac{s(i)}{z(i)} \right) = \beta + \frac{1}{I} \sum_{i=1}^I \frac{\beta s(i)}{z(i)} = \beta + \frac{1}{I} \sum_{i=1}^I \frac{t(i)}{z(i)}$, which is linear in β and \mathbf{t} . This is the new, linear, objective function.

In order to ensure that β takes the desired value, we add an extra restriction of the form $\beta \frac{1}{\beta} = 1$ to (27), i.e. $\beta \frac{1}{\beta} = \beta \left(1 + \frac{1}{O} \sum_{i=I+1}^{I+O} \frac{s(i)}{z(i)} \right) = \beta + \frac{1}{O} \sum_{i=I+1}^{I+O} \frac{t(i)}{z(i)} = 1$.

Furthermore, we expand the definition of the technology, taking into account that $\mathbf{s} = \mathbf{t}/\beta$. The effect of this change of variables on the first restriction is $\mathbf{z} + \mathbf{s} = \mathbf{z} + \mathbf{t}/\beta = \frac{1}{\beta}(\beta \mathbf{z} + \mathbf{t}) \in \hat{T}(\mu^*, \nu^*)$, yielding the following intermediate program:

$$\begin{aligned} \min_{\beta > 0, \mathbf{t} \in \mathbb{R}^{I+O}} \quad & \beta + \frac{1}{I} \sum_{i=1}^I \frac{t(i)}{z(i)} \\ \text{subject to} \quad & \beta + \frac{1}{O} \sum_{i=I+1}^{I+O} \frac{t(i)}{z(i)} = 1, \\ & \left\langle \mathbf{w}^* \cdot \boldsymbol{\phi} \left(\frac{1}{\beta}(\beta \mathbf{z} + \mathbf{t}) \right) \right\rangle \geq \rho^*, \\ & \frac{1}{\beta}(\beta \mathbf{z} + \mathbf{t}) \in \mathbb{R}_-^I \times \mathbb{R}_+^O, \\ & \mathbf{t} \geq \mathbf{0} \end{aligned} \tag{30}$$

At this stage, we have linearized the objective function, and it remains to linearize the maximum function in the PWL mapping, which we do using the same big-M technique that we have used previously. We begin by expanding the definition of $\boldsymbol{\phi}$:

$$\left\langle \mathbf{w}^* \cdot \boldsymbol{\phi} \left(\frac{1}{\beta} (\beta \mathbf{z} + \mathbf{t}) \right) \right\rangle = - \sum_{j=1}^{I+O} w_j^* \left(\frac{1}{\beta} (\beta z(j) + t(j)) \right) - \sum_{j=I+O+1}^{I+O+H} w_j^* \max \left\{ \mu^*, \left\langle \mathbf{p}_j \cdot \left(\frac{1}{\beta} (\beta \mathbf{z} + \mathbf{t}) \right) \right\rangle + q_j \right\}. \tag{31}$$

Then, we introduce the usual auxiliary variable $\boldsymbol{\sigma} \in \mathbb{R}^H$ and, for each hyperplane component of $\boldsymbol{\phi}$, we add, for each $j \in \{I + O + 1, \dots, I + O + H\}$, the restrictions:

$$\sigma_j \geq \mu^*, \tag{32}$$

$$\sigma_j \geq \left\langle \mathbf{p}_j \cdot \left(\frac{1}{\beta} (\beta \mathbf{z} + \mathbf{t}) \right) \right\rangle + q_j. \tag{33}$$

We also add a penalization term with big M in the objective function to ensure that at least one of these bounds is tight, that is, σ_j takes the maximum value of the two compared expressions. Restriction (33) is not linear, since it contains terms involving both \mathbf{t} and β nonlinearly, as \mathbf{t}/β . We linearize it by multiplying both sides by β , to obtain:

$$\tau_j := \beta \sigma_j \geq \langle \mathbf{p}_j \cdot (\beta \mathbf{z} + \mathbf{t}) \rangle + \beta q_j. \tag{34}$$

Thus, we also change variable from $\boldsymbol{\sigma}$ to $\boldsymbol{\tau} = \beta \boldsymbol{\sigma} \in \mathbb{R}^H$. Notice that we also multiply restriction (31), the one describing $\boldsymbol{\phi}$, by β , in order to get rid of the $\frac{1}{\beta}$ term (which creates a nonlinear term of the form $t(j)/\beta$), as well as the $\sigma_j \geq \mu^*$ term in order to express it with respect to the same variables. After all these changes, the final linear problem, in $\beta, \mathbf{t}, \boldsymbol{\tau}$, is:

$$\begin{aligned} \min_{\beta > 0, \mathbf{t} \in \mathbb{R}^{I+O}, \boldsymbol{\tau} \in \mathbb{R}^H} & M \left(\beta + \frac{1}{I} \sum_{i=1}^I \frac{t(i)}{z(i)} \right) + \sum_{j=I+O+1}^{I+O+H} \tau_j \\ \text{subject to} & \beta + \frac{1}{O} \sum_{i=I+1}^{I+O} \frac{t(i)}{z(i)} = 1, \\ & - \sum_{j=1}^{I+O} w_j^* (\beta z(j) + t(j)) - \sum_{j=I+O+1}^{I+O+H} w_j^* \tau_j \geq \beta \rho^*, \\ & \tau_j \geq \beta \mu^*, \quad \forall j \in \{I + O + 1, \dots, I + O + H\} \\ & \tau_j \geq \sum_{k=1}^{I+O} [p_j(k) (\beta z(k) + t(k))] + \beta q_j, \quad \forall j \in \{I + O + 1, \dots, I + O + H\} \\ & (\beta z(j) + t(j)) \leq 0, \quad \forall j \in \{1, \dots, I\} \\ & (\beta z(j) + t(j)) \geq 0, \quad \forall j \in \{I + 1, \dots, I + O\} \\ & \mathbf{t} \geq \mathbf{0} \end{aligned} \tag{35}$$

After solving program (35), we obtain the efficiency measure, in terms of β, \mathbf{t} , as the optimal value of (30), that is, $\beta + \frac{1}{I} \sum_{i=1}^I \frac{t(i)}{z(i)}$. Finally, we have the following result relating the solutions to both programs:

Proposition 4.8 *If $(\mathbf{t}^*, \beta^*, \boldsymbol{\tau}^*)$ is an optimal solution of (35) then $\mathbf{s}^* = \mathbf{t}^* / \beta^*$ is an optimal solution of (27)).*

Proof This proof has two stages, one where the equivalence with the changes of variables are performed, and another one where the linearization is performed. \square

5 Computational experiments: a finite-sample study

In order to evaluate the new method introduced in this paper, this section shows the results obtained from a computational experience, based on a finite-sample analysis, comparing the DEA method with the duplicate and mean hyperplanes strategies, denoted by $1SVM_d$ and $1SVM_m$, respectively. We summarize the results of the simulated technologies in Table 1.

We compare them with the 2 input, 2 output technology proposed by Perelman and Santín (2009) created in a way to ensure the satisfaction of microeconomic behavioral regularity conditions. In this simulation context, the input values are generated from a uniform distribution $Uni[5, 50]$, while the values for the output variables are generated according to the formula:

$$\begin{aligned} -\ln(y_1) = & -1 + 0.5 \left(\frac{\ln y_2}{\ln y_1} \right) + 0.25 \left(\frac{\ln y_2}{\ln y_1} \right)^2 \\ & - 1.5(\ln x_1) - 0.6(\ln x_2) + 0.2(\ln x_1)^2 \\ & + 0.05(\ln x_2)^2 - 0.1(\ln x_1)(\ln x_2) \\ & + 0.05(\ln x_1) \left(\frac{\ln y_2}{\ln y_1} \right) - 0.05(\ln x_2) \left(\frac{\ln y_2}{\ln y_1} \right). \end{aligned} \quad (36)$$

Following Perelman and Santín (2009), we generate points on the frontier of the technology via Equation (36), and then we introduce an inefficiency term $u \sim |N(0, \sqrt{0.3})|$ with a half-normal distribution. We also incorporate random noise. This is indicated in the “Noise” column of Table 1. Furthermore, we allow for a proportion of 0%, 10% and 25% of the simulated DMUs to be on the true frontier. Moreover, we ran 100 trials for each combination of sample size, presence or absence of noise, and percentage of units on the frontier, and we report in Table 1 the average values for the Mean Squared Error (MSE) and bias, as well as % of improvement over the DEA estimator. We tested with sample sizes of 30, 50, 70, 100 and 200 DMUs.

Regarding the results in Table 1, the improvements in MSE range from 20% to 72% in the duplicate hyperplanes strategy, while the improvements in MSE ranged from 15% to 53% in the mean hyperplanes strategy, depending on the sample size and the number of DMUs on the true frontier. On the other hand, the improvements in bias were up to 47% in the first strategy, while the largest improvement in bias was 41%, depending on the sample size and the number of DMUs on the true frontier. Furthermore, the absence of noise in the data

increases the MSE and bias in both $1SVM_m$ and $1SVM_d$ strategies. 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, the $1SVM_d$ technique with the strategy of duplicating hyperplanes used 5.16s for calculating all the estimations, while the DEA technique utilized 0.63s (approximately eight times less than $1SVM_d$). The strategy $1SVM_m$ of averaging the

Table 1 Comparison of the proposed strategies

Num.	Front.	Noise	MSE			Bias		
			DEA	$1SVM_d$	$1SVM_m$	DEA	$1SVM_d$	$1SVM_m$
Obs.								
30	0	No	0.299	0.126 (58%)	0.175 (42%)	0.406	0.256 (37%)	0.272 (33%)
	0	Yes	0.258	0.160 (38%)	0.140 (46%)	0.363	0.260 (28%)	0.239 (34%)
	10	No	0.210	0.118 (44%)	0.174 (17%)	0.315	0.205 (35%)	0.229 (27%)
	10	Yes	0.311	0.141 (55%)	0.149 (52%)	0.334	0.211 (37%)	0.233 (30%)
	25	No	0.172	0.118 (31%)	0.120 (30%)	0.210	0.195 (7%)	0.196 (7%)
	25	Yes	0.179	0.144 (20%)	0.091 (49%)	0.229	0.164 (28%)	0.161 (29%)
50	0	No	0.212	0.081 (62%)	0.119 (44%)	0.332	0.192 (42%)	0.209 (37%)
	0	Yes	0.212	0.086 (59%)	0.125 (41%)	0.314	0.198 (37%)	0.211 (33%)
	10	No	0.173	0.086 (51%)	0.099 (43%)	0.265	0.158 (40%)	0.158 (41%)
	10	Yes	0.171	0.077 (55%)	0.145 (15%)	0.245	0.161 (34%)	0.182 (26%)
	25	No	0.094	0.039 (59%)	0.054 (43%)	0.182	0.102 (44%)	0.118 (35%)
	25	Yes	0.102	0.053 (48%)	0.078 (24%)	0.168	0.117 (31%)	0.141 (16%)
70	0	No	0.172	0.101 (42%)	0.096 (44%)	0.300	0.182 (39%)	0.183 (39%)
	0	Yes	0.140	0.092 (35%)	0.093 (33%)	0.263	0.190 (28%)	0.188 (28%)
	10	No	0.113	0.068 (40%)	0.068 (40%)	0.213	0.125 (41%)	0.132 (38%)
	10	Yes	0.124	0.090 (28%)	0.058 (53%)	0.218	0.173 (21%)	0.138 (37%)
	25	No	0.082	0.029 (64%)	0.052 (36%)	0.143	0.096 (32%)	0.111 (22%)
	25	Yes	0.095	0.065 (31%)	0.049 (48%)	0.151	0.128 (16%)	0.124 (18%)
100	0	No	0.130	0.048 (63%)	0.074 (43%)	0.255	0.148 (42%)	0.159 (38%)
	0	Yes	0.113	0.069 (39%)	0.063 (44%)	0.235	0.162 (31%)	0.148 (37%)
	10	No	0.084	0.047 (44%)	0.051 (39%)	0.179	0.118 (34%)	0.129 (28%)
	10	Yes	0.075	0.052 (31%)	0.048 (36%)	0.169	0.127 (25%)	0.133 (21%)
	25	No	0.065	0.032 (51%)	0.045 (31%)	0.133	0.087 (35%)	0.120 (9%)
	25	Yes	0.052	0.036 (30%)	0.037 (29%)	0.117	0.109 (7%)	0.096 (18%)
200	0	No	0.131	0.037 (72%)	0.069 (47%)	0.249	0.148 (41%)	0.161 (35%)
	0	Yes	0.101	0.069 (32%)	0.062 (39%)	0.230	0.162 (30%)	0.148 (36%)
	10	No	0.089	0.036 (60%)	0.049 (45%)	0.157	0.118 (25%)	0.138 (12%)
	10	Yes	0.080	0.052 (35%)	0.048 (40%)	0.157	0.113 (28%)	0.128 (18%)
	25	No	0.069	0.033 (52%)	0.043 (38%)	0.148	0.079 (47%)	0.132 (11%)
	25	Yes	0.056	0.033 (42%)	0.037 (34%)	0.129	0.112 (13%)	0.104 (19%)

Table 2 Descriptive statistics of the USA PISA 2015 dataset

Variable	ESCS	SCMATEDU	TEACHERS	PVMATH	PVREAD
Mean	4.165	4.565	7.530	479.314	495.658
Median	4.191	4.590	6.055	485.665	501.365
Std. Dev	0.560	0.560	8.954	48.812	52.413
Max	5.250	5.650	113.895	586.230	599.750
90%	4.899	5.299	9.020	537.489	555.567
75%	4.591	4.990	7.791	513.610	529.108
25%	3.821	4.223	5.376	447.013	464.213
10%	3.506	3.910	4.407	422.461	430.290
Min	2.195	2.590	0.849	283.110	274.690

hyperplanes used 7.56s in the same experiment (approximately twelve times the time taken by DEA, and 1.5 times the time spent with the duplication strategy).

We conclude that both strategies seem to obtain better results than DEA in the finite-sample study carried out, with the duplicate hyperplanes strategy performing better than the mean hyperplanes strategy in MSE, bias and runtime, so we choose it for the empirical application and as the superior strategy.

6 Empirical illustration: USA schools from PISA report 2015

In this section, we present the results of the application of the estimator introduced in this paper in an empirical database from the literature, using in particular the Slacks-Based Measure (SBM) for illustration. This database consists of results from USA schools participating in the PISA (Programme for International Student Assessment) in 2015, used in Aparicio et al. (2019), and further details can be found in OECD (2017). The dataset that we use contains 162 DMUs (schools), and we report the descriptive statistics in Table 2. We used the following variables for inefficiency estimation: three inputs and two outputs. The three inputs were the Economic, Social and Cultural Status (ESCS), the school's educational resources (SCMATEDU) and the number of teachers per 100 students (TEACHERS). The outputs were their scores in math and reading (PVMATH and PVREAD). Regarding the hyperparameters used in $1SVM_d$, for this application we fix $v^* = 0.001$. This is because the SBM, as well as some of the other measures introduced, are not defined for units outside the estimated technology, so by fixing a small value for v^* we weigh the estimator towards the exclusion of outliers. In fact, there are no outliers with this setting. We consider μ as a hyperparameter, and tune it via the usual train-test (70%-30%).

With this dataset, we estimate the technology by using the standard DEA methodology and our introduced estimator with the duplicate hyperplanes strategy ($1SVM_d$). We then calculate the efficiencies estimated by DEA and $1SVM_d$ for each school with respect to the SBM, and we compare them by means of the Li test (Simar and Zelenyuk 2006).

Table 3 Descriptive statistics of the SBM efficiency scores, USA PISA 2015 dataset

Estimator	1SVM _d	DEA
Mean	0.697	0.739
Median	0.680	0.720
Std. Dev	0.089	0.089
Max	1.000	1.000
90%	0.812	0.863
75%	0.726	0.768
25%	0.645	0.682
10%	0.613	0.645
Min	0.512	0.574

Table 4 SBM Efficiency scores for selected DMUs

DMU	ESCS	SCMATEDU	TEACHERS	PVMATH	PVREAD	SBM	
						1SVM _d	DEA
18	2.884	3.28	3.431	402.66	390.20	1.000	1.000
39	4.610	5.01	4.492	586.23	590.58	0.889	1.000
104	2.881	3.28	4.268	437.91	462.13	1.000	1.000
115	2.518	2.92	5.963	428.42	435.69	1.000	1.000
120	4.301	4.70	4.662	565.90	587.02	1.000	1.000
128	2.195	2.59	4.810	283.11	274.69	0.919	1.000
143	4.633	5.03	0.849	536.67	568.18	1.000	1.000
<i>Top 5 least efficient DEA DMUs</i>							
25	3.863	4.26	12.781	429.20	437.12	0.588	0.602
42	3.735	4.13	113.895	439.60	523.44	0.531	0.574
90	4.317	4.72	19.616	498.57	501.79	0.593	0.613
117	4.579	4.98	5.928	386.29	407.50	0.512	0.622
139	4.930	5.33	11.297	500.94	536.77	0.593	0.611
<i>Top 5 biggest changes in efficiency</i>							
46	3.901	4.30	5.963	369.65	351.00	0.546	0.682
84	4.844	5.24	4.990	581.31	591.39	0.730	0.922
132	4.059	4.46	2.718	445.77	449.24	0.737	0.889
159	4.593	4.99	4.275	566.83	574.19	0.768	0.897
162	3.319	3.72	8.097	346.88	297.16	0.530	0.677

The train-test process selects the value $\mu^* = -0.1813$ as the best one. The average efficiency reported by DEA is 0.739, whereas 1SVM_d attains 0.697, which indicates lower estimates of the efficiency. We report descriptive statistics of these efficiency scores in Table 3. The DEA estimate of the technology classifies 7 DMUs as efficient under the application of the SBM, whereas the proposed approach estimate indicates only 5 units as efficient, all of which were already deemed efficient by DEA. The remaining 2 are deemed not completely efficient, attaining efficiency scores of

Table 5 SBM efficiency scores

DMU	$1SVM_d$	DEA	DMU	$1SVM_d$	DEA	DMU	$1SVM_d$	DEA
1	0.660	0.751	55	0.632	0.704	109	0.776	0.829
2	0.811	0.838	56	0.683	0.703	110	0.677	0.697
3	0.694	0.726	57	0.716	0.741	111	0.660	0.706
4	0.695	0.747	58	0.724	0.755	112	0.621	0.689
5	0.619	0.642	59	0.674	0.741	113	0.550	0.641
6	0.591	0.630	60	0.750	0.790	114	0.803	0.897
7	0.676	0.717	61	0.678	0.690	115	1.000	1.000
8	0.612	0.637	62	0.689	0.723	116	0.607	0.624
9	0.733	0.801	63	0.676	0.711	117	0.512	0.622
10	0.660	0.727	64	0.675	0.693	118	0.666	0.744
11	0.611	0.637	65	0.654	0.741	119	0.799	0.836
12	0.696	0.728	66	0.674	0.729	120	1.000	1.000
13	0.707	0.752	67	0.701	0.747	121	0.641	0.682
14	0.708	0.746	68	0.671	0.767	122	0.683	0.725
15	0.741	0.779	69	0.602	0.668	123	0.684	0.707
16	0.647	0.677	70	0.619	0.669	124	0.724	0.759
17	0.708	0.743	71	0.673	0.692	125	0.636	0.657
18	1.000	1.000	72	0.639	0.675	126	0.815	0.874
19	0.570	0.636	73	0.644	0.689	127	0.647	0.679
20	0.796	0.835	74	0.639	0.690	128	0.919	1.000
21	0.726	0.768	75	0.637	0.662	129	0.657	0.677
22	0.773	0.795	76	0.678	0.721	130	0.688	0.730
23	0.756	0.783	77	0.634	0.674	131	0.700	0.723
24	0.676	0.695	78	0.649	0.698	132	0.737	0.889
25	0.588	0.602	79	0.688	0.728	133	0.691	0.736
26	0.694	0.757	80	0.707	0.763	134	0.676	0.693
27	0.702	0.722	81	0.767	0.803	135	0.765	0.788
28	0.630	0.658	82	0.645	0.700	136	0.827	0.851
29	0.703	0.747	83	0.676	0.704	137	0.829	0.847
30	0.697	0.777	84	0.730	0.922	138	0.703	0.737
31	0.680	0.717	85	0.624	0.686	139	0.593	0.611
32	0.634	0.656	86	0.652	0.711	140	0.658	0.674
33	0.778	0.836	87	0.682	0.709	141	0.638	0.647
34	0.690	0.708	88	0.704	0.746	142	0.614	0.645
35	0.829	0.864	89	0.576	0.630	143	1.000	1.000
36	0.722	0.747	90	0.593	0.613	144	0.661	0.679
37	0.683	0.712	91	0.662	0.687	145	0.849	0.866
38	0.836	0.845	92	0.821	0.911	146	0.728	0.795
39	0.889	1.000	93	0.735	0.770	147	0.660	0.683
40	0.671	0.726	94	0.649	0.670	148	0.646	0.697
41	0.641	0.656	95	0.888	0.920	149	0.760	0.790
42	0.531	0.574	96	0.645	0.661	150	0.642	0.661

Table 5 (continued)

DMU	1SVM _d	DEA	DMU	1SVM _d	DEA	DMU	1SVM _d	DEA
43	0.712	0.754	97	0.732	0.768	151	0.607	0.635
44	0.813	0.852	98	0.693	0.717	152	0.621	0.707
45	0.720	0.760	99	0.919	0.972	153	0.660	0.686
46	0.546	0.682	100	0.637	0.677	154	0.643	0.666
47	0.680	0.701	101	0.620	0.672	155	0.693	0.717
48	0.688	0.720	102	0.606	0.629	156	0.660	0.693
49	0.617	0.636	103	0.679	0.698	157	0.652	0.711
50	0.751	0.780	104	1.000	1.000	158	0.704	0.750
51	0.690	0.722	105	0.741	0.767	159	0.768	0.897
52	0.710	0.749	106	0.661	0.679	160	0.785	0.794
53	0.703	0.734	107	0.701	0.733	161	0.651	0.689
54	0.668	0.708	108	0.669	0.694	162	0.530	0.677

0.889 and 0.919. The efficiencies of these and other selected DMUs are reported in Table 4. We also report on the 5 DMUs where the change in the efficiency score is the largest between the 1SVM_d and the DEA scores with respect to the SBM, as well as the 5 DMUs deemed least efficient by DEA. Furthermore, Table 5 reports the SBM efficiency obtained by each DMU in the dataset with respect to both 1SVM_d and DEA, where values in bold indicate that the DMU is efficient.

In order to compare the vectors of efficiencies associated with DEA and the new approach, we use the Li test, following Simar and Zelenyuk (2006), which considers whether there is a statistically significant difference between two random samples Z_A, Z_B , with distribution densities f_A, f_B . It considers as the Null Hypothesis that $H_0 : f_A(Z_A) = f_B(Z_B)$, and calculates the corresponding p -value for its possible rejection. We apply the Li test to compare the SBM efficiency scores obtained by DEA and 1SVM_d in order to check whether the differences are statistically significant, and obtain a p -value of 0.0008, indicating that the estimated scores show statistically significant differences.

From the density distribution curves associated with the SBM efficiency scores (see Fig. 2), we observe that the efficiency scores obtained by 1SVM_d on the dataset classify fewer DMUs as technically efficient, and have consistently slightly higher inefficiencies than the corresponding DEA scores. This indicates results consistent with our goal of estimating slightly larger technologies than those estimated by DEA, which suffers from overfitting (Esteve et al. 2020).

7 Conclusions and future work

In this paper, we have explored methods to estimate a production technology by adapting the 1SVM algorithm, with a piecewise linear feature mapping. We have thus built a bridge between the fields of unsupervised machine learning and efficiency estimation via DEA in the context of multi-input multi-output production

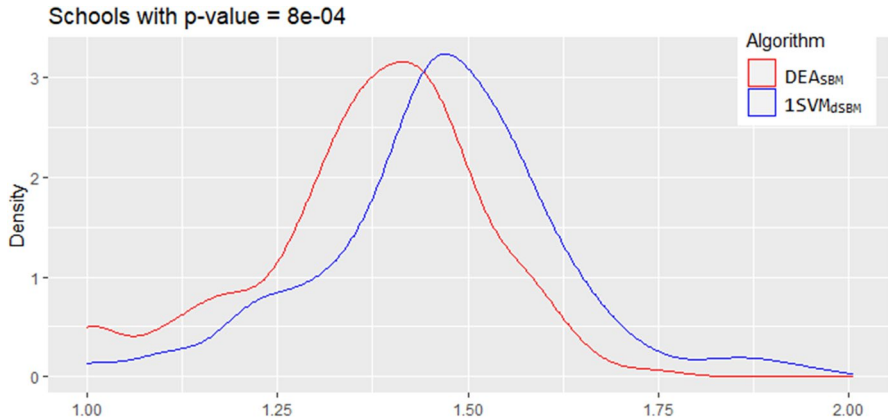


Fig. 2 Density distribution of the efficiency scores with respect to the SBM

processes. We have introduced the corresponding methodology and some variations on the hyperplane parameters involved in the feature mapping in order to estimate the technology.

We have evaluated the performance of the proposed estimators by comparing the results obtained in a finite-sample simulated environment using multiple inputs and outputs. From our results, we conclude that the approach which duplicates the DEA hyperplanes ($1SVM_d$) is superior to that which calculates the mean of the existing hyperplanes ($1SVM_m$), in addition to being less computationally expensive. Also, it is worth mentioning that both proposed approaches seem to obtain better results than the standard DEA approach regarding MSE and bias under our finite-sample analysis. However, this superiority cannot be claimed in general. From a statistical point of view, the frontier estimators could also be compared regarding some properties such as consistency, which indicates whether the estimator converges to the true target value as the sample size increases. Additionally, when the objective is to report the average efficiency score, the satisfaction of the central limit theorem is also a relevant property. Regarding the DEA estimator, these properties have been studied in detail, even establishing the rate of convergence (see, for example, Kneip et al. (1998, 2008, 2011, 2015)). One of the advantages of this knowledge is that, in the case of DEA, it is relatively easy to correct the potential bias of the estimator to determine, for example, suitable confidence intervals through bootstrapping. Unfortunately, we cannot reach the same conclusion regarding the new approach. Consequently, a complete comparison of our technique and the standard DEA model is neither possible nor fair. This way, our approach could be seen as a complementary technique to DEA when the data sample is not large.

Moreover, we have introduced and adapted multiple measures of efficiency found in the non-parametric literature on performance measurement to our context. For illustrative purposes, we have also shown the results obtained by applying, in particular, the Slacks-Based Measure to an empirical database involving schools in the USA from the PISA study. We have compared the scores determined by our proposed approach to those obtained by classical DEA. In our empirical application, the

new approach yields fewer efficient DMUs; thus showing a higher discriminating power than DEA, although this topic deserves further exploration.

Finally, we mention some possible avenues for further research. The choices we have made for the hyperplanes will allow for different approaches and exploration, as well as different transformation functions altogether. Different choices of hyperplanes, or even of other types of transformation functions may be worth pursuing to further improve the methodology. Furthermore, these parameters could be considered as hyperparameters to tune, although this would be very computationally expensive. This contribution takes part in the larger context of machine learning algorithms adapted to estimate production technologies, and another possible future line of research would be the comparison amongst them and their performance. Another possible area of interest is the enrichment of this approach with feature selection methods, since one of the ever-arising issues is the curse of dimensionality, and these methods may be extended to approach problems where direct computation may not be possible or may yield insufficient results. In addition, the possibility of using the new approach 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 further explorations. And, as was pointed out above, the study of the asymptotic properties of the new frontier estimator can be understood as one of the most important possible extensions of the method.

Funding 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 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 partially supported this work. Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Aigner D, Lovell CAK, Schmidt P (1977) Formulation and estimation of stochastic frontier production function models. *J Econom* 6(1):21–37. [https://doi.org/10.1016/0304-4076\(77\)90052-5](https://doi.org/10.1016/0304-4076(77)90052-5)
- Aparicio J, Cordero JM, Ortiz L (2019) Measuring efficiency in education: the influence of imprecision and variability in data on dea estimates. *Socio-Economic Planning Sci* 68(100):698. <https://doi.org/10.1016/j.seps.2019.03.004>, <https://www.sciencedirect.com/science/article/pii/S0038012118302684>
- Aparicio J, Esteve M (2022) How to peel a data envelopment analysis frontier: a cross-validation-based approach. *J Op Res Soc* 1–15

- Aparicio J, Perelman S, Santín D (2020) Comparing the evolution of productivity and performance gaps in education systems through dea: an application to latin american countries. *Op Res* 1–35
- Banker RD, Charnes A, Cooper WW (1984) Some models for estimating technical and scale inefficiencies in data envelopment analysis. *Manag Sci* 30(9):1078–1092
- Briec W (1999) Hölder distance function and measurement of technical efficiency. *J Prod Anal* 11(2):111–131. <https://doi.org/10.1023/A:1007764912174>
- Chambers RG, Chung Y, Färe R (1998) Profit, directional distance functions, and nerlovian efficiency. *J Optim Theory Appl* 98(2):351–364
- Charnes A, Cooper WW (1962) Programming with linear fractional functionals. *Naval Res Logist Quart* 9(3–4):181–186
- Charnes A, Cooper WW, Rhodes E (1978) Measuring the efficiency of decision making units. *Eur J Oper Res* 2(6):429–444
- Charnes A, Cooper WW, Golany B et al (1985) Foundations of data envelopment analysis for pareto-koopmans efficient empirical production functions. *J Econom* 30(1–2):91–107
- Cooper WW, Park KS, Pastor JT (1999) Ram: a range adjusted measure of inefficiency for use with additive models, and relations to other models and measures in dea. *J Prod Anal* 11(1):5–42
- Cooper WW, Seiford LM, Tone K (2006) Introduction to data envelopment analysis and its uses: with DEA-solver software and references. Springer Science & Business Media, Berlin
- Daouia A, Noh H, Park BU (2016) Data envelope fitting with constrained polynomial splines. *J R Stat Soc: Ser B (Stat Methodol)* 78(1):3–30
- Daraio C, Simar L (2007) Advanced robust and nonparametric methods in efficiency analysis: Methodology and Applications. Studies in Productivity and Efficiency, Springer, US, <https://books.google.es/books?id=QAtGqmOwyIwC>
- Ebrahimi B, Tavana M, Charles V (2021) A note and new extensions on “interval efficiency measures in data envelopment analysis with imprecise data”. *Oper Res Int J* 21(4):2719–2737
- Esteve M, Aparicio J, Rabasa A et al (2020) Efficiency analysis trees: A new methodology for estimating production frontiers through decision trees. *Expert Syst Appl* 162(113):783
- Esteve M, Aparicio J, Rodriguez-Sala JJ et al (2022) Random forests and the measurement of super-efficiency in the context of free disposal hull. *Eur J Op Res* 304(2):729–744
- Färe R, Grosskopf S (2000) Theory and application of directional distance functions. *J Prod Anal* 13(2):93–103. <https://doi.org/10.1023/A:1007844628920>
- Färe R, Lovell CK (1978) Measuring the technical efficiency of production. *J Econ Theory* 19(1):150–162
- Färe R, Grosskopf S, Lovell CK (1985) The measurement of efficiency of production. Springer Science & Business Media, Berlin
- Farrell MJ (1957) The measurement of productive efficiency. *J R Stat Soc Ser A (General)* 120(3):253–290
- Gong BH, Sickles RC (1992) Finite sample evidence on the performance of stochastic frontiers and data envelopment analysis using panel data. *J Econ* 51(1–2):259–284
- Huang X, Mehrkanoon S, Suykens J (2013) Support vector machines with piecewise linear feature mapping. *Neurocomputing* 117:118–127. <https://doi.org/10.1016/j.neucom.2013.01.023>
- Kneip A, Park BU, Simar L (1998) A note on the convergence of nonparametric DEA estimators for production efficiency scores. *Economet Theor* 14(6):783–793
- Kneip A, Simar L, Wilson PW (2008) Asymptotics and consistent bootstraps for DEA estimators in non-parametric frontier models. *Economet Theor* 24(6):1663–1697
- Kneip A, Simar L, Wilson PW (2011) A computationally efficient, consistent bootstrap for inference with non-parametric DEA estimators. *Comput Econ* 38(4):483
- Kneip A, Simar L, Wilson PW (2015) When bias kills the variance: Central limit theorems for DEA and FDH efficiency scores. *Economet Theor* 31(2):394–422
- Kuosmanen T, Johnson AL (2010) Data envelopment analysis as nonparametric least-squares regression. *Oper Res* 58(1):149–160
- Liao Z, Dai S, Kuosmanen T (2022) Convex support vector regression. arXiv preprint [arXiv:2209.12538](https://arxiv.org/abs/2209.12538)
- Lovell CK, Pastor JT (1995) Units invariant and translation invariant dea models. *Oper Res Lett* 18(3):147–151
- Luenberger DG (1992) Benefit functions and duality. *J Math Econom* 21(5):461–481. [https://doi.org/10.1016/0304-4068\(92\)90035-6](https://doi.org/10.1016/0304-4068(92)90035-6)
- Luenberger DG (1992) New optimality principles for economic efficiency and equilibrium. *J Optim Theory Appl* 75(2):221–264. <https://doi.org/10.1007/BF00941466>

- Meeusen W, van Den Broeck J (1977) Efficiency estimation from cobb-douglas production functions with composed error. *Int Econ Rev* 18(2):435–444
- OECD (2017) Pisa 2015 technical report
- Olesen OB, Ruggiero J (2018) An improved afriat-diewert-parkan nonparametric production function estimator. *Eur J Oper Res* 264(3):1172–1188
- Olesen O, Ruggiero J (2022) The hinging hyperplanes: An alternative nonparametric representation of a production function. *Eur J Op Res* 296(1):254–266. <https://doi.org/10.1016/j.ejor.2021.03.054>
- Parmeter CF, Racine JS (2013) Smooth constrained frontier analysis. In: Chen X, Swanson NR (eds) *Recent Advances and Future Directions in Causality, Prediction, and Specification Analysis: Essays in Honor of Halbert L. White Jr.* Springer, New York, NY, pp 463–488. https://doi.org/10.1007/978-1-4614-1653-1_18
- Pastor JT, Ruiz JL, Sirvent I (1999) An enhanced dea russell graph efficiency measure. *Eur J Oper Res* 115(3):596–607
- Pastor JT, Lovell CK, Aparicio J (2012) Families of linear efficiency programs based on Debreu's loss function. *J Product Anal* 38(2):109–120. <https://doi.org/10.1007/s11123-011-0216-4>
- Perelman S, Santín D (2009) How to generate regularly behaved production data? a monte carlo experimentation on dea scale efficiency measurement. *Eur J Oper Res* 199(1):303–310
- Schölkopf B, Platt JC, Shawe-Taylor J et al (2001) Estimating the support of a high-dimensional distribution. *Neural Comput* 13(7):1443–1471
- Simar L, Zelenyuk V (2006) On testing equality of two distribution functions of efficiency scores estimated from dea. *Economet Rev* 25(497):522
- Thaker K, Charles V, Pant A et al (2022) A DEA and random forest regression approach to studying bank efficiency and corporate governance. *J Op Res Soc* 73(6):1258–1277
- Tone K (2001) A slacks-based measure of efficiency in data envelopment analysis. *Eur J Oper Res* 130(3):498–509
- Tsionas MG (2022) Efficiency estimation using probabilistic regression trees with an application to chilean manufacturing industries. *Int J Product Econ* 249:108492
- Tsolas IE, Charles V, Gherman T (2020) Supporting better practice benchmarking: A DEA-ANN approach to bank branch performance assessment. *Expert Syst Appl* 160(113):599
- Valero-Carreras D, Aparicio J, Guerrero NM (2021) Support vector frontiers: A new approach for estimating production functions through support vector machines. *Omega* 104(102):490. <https://doi.org/10.1016/j.omega.2021.102490>
- Valero-Carreras D, Aparicio J, Guerrero NM (2022) Multi-output support vector frontiers. *Comput Op Res* 143(105):765
- Vapnik V (1998) *Statistical learning theory*. A Wiley-Interscience publication, Wiley, <https://books.google.es/books?id=GowoQAAMAAJ>
- Vapnik V (2013) *the nature of statistical learning theory*. Information science and statistics, Springer, New York, <https://books.google.es/books?id=EgqACAAAQBAJ>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.