Support Vector Frontiers: a New Approach for Estimating Production Functions through Support Vector Machines

Daniel Valero-Carreras, Juan Aparicio* and Nadia M. Guerrero

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

* Corresponding author: j.aparicio@umh.es (Tel.: +34 966658517; Fax: +34 966658715).

Abstract

In microeconomics, a topic of interest is the estimation of production functions. By definition, a production function is a non-decreasing function that (upper) envelops all the observations (firms) in the input-output space, capturing the extreme behavior of the data. These characteristics are far from the usual ones assumed by machine learning techniques like Support Vector Regression (SVR) in Support Vector Machines, where the function to be estimated relates the response variable to the covariables in terms of the mean instead of the maximum and, additionally, it tries to fit the data as much as possible, determining a function that increases and decreases following a data-driven process. In this paper, we introduce an adaptation of SVR for the first time, denominated Support Vector Frontiers (SVF), with the objective of estimating production functions. To do so and seeking meeting points between SVR and the standard non-parametric techniques for estimating production functions, mainly Free Disposal Hull (FDH) and Data Envelopment Analysis (DEA), a step function is defined in this paper through a specific input transformation function. However, and in contrast to FDH and DEA, SVF overcomes the problem of overfitting that these techniques suffer. Additionally, we show in this paper that standard FDH and DEA could be reinterpreted, in some sense, as Support Vector Regression techniques. Moreover, a new robust notion of efficiency is introduced, called ε-insensitive technical efficiency, directly inherited from Support Vector Machines. Finally, the performance of SVF is measured via Monte Carlo simulations, showing that the new approach considerably reduces the bias and mean squared error associated with the estimation of the true production function in comparison with standard FDH and DEA.

Keywords: Technical Efficiency, Support Vector Regression, Free Disposal Hull, Data Envelopment Analysis

1. Introduction

Support Vector Machines (SVM) is a well-known learning machine technique based on advances in Statistical Learning. It is mainly rooted in the principle of structural risk minimization. In particular, the technique aims to minimize the bound on the generalization error (i.e., the error made by the learning machine on data outside the training set) rather than directly minimizing the empirical error, such as the traditional mean square error over the observed data set. The introduction of SVM in Cortes and Vapnik (1995) and Vapnik (1995, 1998) has led to a torrent of applications and theoretical analysis, which has now established SVM as one of the standard tools for machine learning. In the last few years, there have been very significant developments in the theoretical understanding of SVM as well as the application of the approach to many different empirical contexts (Nayak et al. 2015). The technique has reached the point at which it is clearly viewed as one of the most fruitful research subareas within machine learning, a subarea which has now reached a very high degree of maturity in both theoretical insights and practical usefulness.

Regardless of reaching this stage of development, we are aware that not all the significant contexts had yet been addressed from a methodological point of view. Nowadays, there is no adaptation of SVM in the literature for approximating production function in microeconomics, a field which is related to the measurement of technical efficiency of firms. This is a special framework that requires specific needs regarding the surface to be estimated, as we will go on to explain.

Technical efficiency assessment is concerned with measuring the performance of firms, which convert inputs into outputs. An example of a firm is a shoe manufacturing factory that uses materials, labor and capital (inputs) to produce shoes (output). The measurement of technical efficiency may be of interest for a great variety of industrial sectors, from private sector firms producing goods, such as a manufacturing factory, to service industries, such as travel agencies. The approach may also be used for analyzing the relative performance of units within an organization (e.g., bank branches or fast food chains). It can also be applied to non-profit organizations, such as schools or hospitals, at a macro and micro level (Coelli et al., 2005). Indeed, efficiency evaluation in production has been and is a relevant topic for managers and policy makers, as well as an area that is worthy of attention from a practical and methodological point of view in both Engineering and Economics (see, for example, Arnaboldi et al., 2014, Aparicio et al., 2017, O'Donnell, 2018). From a methodological perspective, the main aim of such assessment is to analyze the technical efficiency of a set of observations, generally known as Decision Making Units (DMUs), by comparing their performance with respect to the so-called production possibility set or technology, which is unknown and must be estimated from a data sample (a learning sample in the terminology of machine learning). In this context, the cornerstone for the efficiency analysis of DMUs is the notion of production function. A production function represents the maximum product obtainable from the input combination at the existing state of technical knowledge. Its estimation allows calculating the corresponding technical inefficiency value as the deviation of each DMU to the (upper) boundary of the technology, characterized by the production function. In fact, given a level of input consumption, the most usual and natural measure of technical efficiency for firms is defined as the ratio of the actual produced output and the maximal producible output, determined by the production function.

As for the estimation of production functions in practice, before Farrell's (1957) seminal contribution, economists used to specify the corresponding production functions parametrically, e.g., a Cobb-Douglas function (Cobb and Douglas, 1928), relying on Ordinary Least Squares (OLS) regression analysis to estimate an 'average' production function, and assuming that disturbance terms had zero mean. This was a clearly unsatisfactory estimation, as it did not follow the accepted definition of production functions in microeconomics as the 'maximal' feasible output for each resource combination considered. In this sense, Farrell (1957) was the first in showing, for a single output and multiple inputs, how to estimate a surface upper enveloping all the observations. Farrell's contribution was based on the construction of a technology that satisfied two usual axioms in production theory: convexity and monotonicity (free disposability). Convexity establishes that if two input-output bundles are feasible (producible), then any convex combination of them is also feasible; whereas monotonicity states that if inputs increase, then the producible output must, at least, not decrease. However, many estimators meet both properties and, consequently, additional requirements are needed. In particular, the most conservative estimation of the production function would be that associated with a surface enveloping the data and, at the same time, as close as possible to them. This is the principle of conservation, known also as 'minimal extrapolation', which in the case of Farrell's estimator leads to a piece-wise linear surface. Additionally, his contribution constitutes the first implementation of Debreu's coefficient of resource utilization (Debreu, 1951) and Shephard's input distance function (Shephard, 1953).

Farrell's approach can be categorized in the current area of non-parametric techniques for estimating production functions since it is not necessary to identify, a priori, the specific mathematical formulation of the production function to be estimated. This line of research, initiated by Farrell, was later taken up by Charnes et al. (1978) and Banker et al. (1984), resulting in the development of the Data Envelopment Analysis (DEA) approach, in which the determination of the frontier is only restricted via its axiomatic foundation (mainly convexity, free disposability and minimal extrapolation). In this case, the axiom of convexity is translated into an additional requirement of the production function: concavity. Another paper working in this same line, is that by Afriat (1972), showing how to determine a production function with the property P (e.g., non-decreasing concavity) that represents the set of observations to be as nearly efficient as possible. A more natural sequel than the DEA literature of the previous work done by econometricians, even before Farrell's contribution, would be Aigner and Chu (1968), who showed how to estimate a 'parametric' Cobb-Douglas production function upper enveloping the data cloud. Furthermore, in a parallel way, Deprins et al. (1984) introduced the alternative technique known as Free Disposal Hull (FDH), which relies exclusively on monotonicity and minimal extrapolation in contrast to DEA, which additionally assumes convexity.

Nowadays, two famous non-parametric approaches for estimating production functions are DEA and FDH (Daraio and Simar, 2007; O'Donnell, 2018). In the case of DEA, the estimator is a piece-wise linear function, while in the case of FDH, the estimator is a step function. It is worth mentioning that FDH may be considered the 'skeleton' of DEA since the convex hull of the frontier estimated by FDH coincides with the DEA frontier (Daraio and Simar, 2005). Moreover, the Data Generating Process (DGP) assumed in both techniques can be summarized as follows. If we assume that m inputs are involved in the production process, an unknown non-decreasing function $f(\boldsymbol{x}): \Re_{+}^{m} \rightarrow \Re_{+}$ indicates the maximal output that is producible from an input bundle $\boldsymbol{x}$. However, in

practice, technical inefficiency can occur. Following Farrell (1957), technical inefficiency reflects the inability of a DMU to obtain maximal output from a given set of inputs, due, for example, to a mismanagement of resources. Therefore, the actual observed output for a DMU coincides with $y = f(\boldsymbol{x}) - u \geq 0$, where u is conceptualized as a non-negative random variable linked to technical inefficiency. Note that then $y \leq f(\boldsymbol{x})$, which is associated with the upper enveloping nature of the production function described above in the text. In this way, in practice, for each DMU we really observe y instead of $f(\boldsymbol{x})$, but we need to determine the ratio $y/f(\boldsymbol{x})$, which measures the level of technical inefficiency of the considered unit. This is the methodological problem that is faced by researchers in the field of efficiency measurement. Additionally, other recent alternative non-parametric techniques for estimating production frontiers are those that apply Kernel-based approaches and local regression techniques. See, for example, Du et al. (2013), where the authors propose a kernel smoothing method that can handle multiple shape constraints (e.g., monotonicity) for multivariate functions, generalizing Hall and Huang (2001). Another interesting contribution is Parmeter et al. (2014), who showed how constraint weighted bootstrapping may be applied to impose smoothness conditions on linear estimates. In particular, these authors estimated an input distance function both parametrically and non-parametrically, resorting in the last case to local linear generalized kernel regression. See also Henderson and Parmeter (2009).

Analyzed as a machine learning technique, by construction, the FDH estimator $\hat{f}_{FDH}$ is not endowed with generalization (out-of-sample) capability. In particular, the principle of minimal extrapolation focuses all its attention on minimizing the empirical error over the observed data set (i.e., minimizing the quantities $\hat{f}_{FDH}(\boldsymbol{x}) - y)$, provided that $\hat{f}_{FDH}(\boldsymbol{x}): \Re_{+}^{m} \rightarrow \Re_{+}$ is a non-decreasing function upper enveloping the data sample. In this sense, FDH suffers from the usual overfitting problem. FDH underestimates the real technical inefficiency of the observations, as this technique yields estimators always located below the (underlying) theoretical frontier f. Nevertheless, Free Disposal Hull is able to correctly 'describe' the situation of a particular set of observations from a 'relative' of the sample or sample-specific efficiency evaluation point of view, standing out as a descriptive statistic tool with little inferential capability for smaller samples (see its asymptotic properties and rates of convergence in Simar and Wilson, 2008). It is worth mentioning that something similar happens with respect to the DEA estimator $\hat{f}_{DEA}$. Consequently, despite the recognized data-driven nature of FDH and DEA, a conceptual gap remains in the literature between these common techniques in the field of efficiency evaluation and the machine learning techniques world. Anyway, there are certain key previous papers in the literature that have dealt with FDH and DEA techniques in order to transform them into inferential tools. Chronologically speaking, Banker and Maindiratta (1992) and Banker (1993) showed that DEA can be interpreted as a Maximum Likelihood estimator. Later, Simar and Wilson (1998) and Simar and Wilson (2000a, 2000b) introduced how to determine confidence intervals for the efficiency score of each DMU, the ratio $y/f(\boldsymbol{x})$ in our context, through adapting the bootstrapping methodology by Efron (1979) to the context of Free Disposal Hull and Data Envelopment Analysis. More recently, Kuosmanen and Johnson (2010) and Kuosmanen and Johnson (2017) have shown that DEA may be interpreted as non-parametric least-squares regression subject to shape constraints on the production frontier and sign constraints on residuals. Additionally, these authors introduced the Corrected Concave Non-

parametric Least Squares and showed that, if the data-generating process is deterministic and the inefficiency terms are identically and independently distributed, their estimator is consistent and asymptotically unbiased. However, none of them addresses the problem through machine learning techniques, despite being one of the most natural theoretical frameworks to be considered given the data-driven nature of DEA and FDH. One recent exception is Esteve et al. (2020), where Classification and Regression Trees (CART) are adapted for estimating production functions through step functions, so competing against the standard Free Disposal Hull technique. See also Misiunas et al. (2016), Zhu (2019), Charles et al. (2020) and Lee and Cai (2020) to read about the recent interest of the DEA community in bridging the gap between Data Envelopment Analysis and data science, machine learning and big data.

In this paper, for the first time, the SVM technique is adapted for estimating production functions, satisfying the usual postulates established in microeconomics textbooks. The new approach will be named Support Vector Frontiers (SVF). SVF will allow estimating production functions by applying the structural risk minimization principle. In particular, the technique will also consider the generalization and empirical errors, and not just the minimization of the empirical error as happens with FDH and DEA. In this way, this paper contributes to bridging the conceptual gap, pointed out above, by demonstrating that FDH and DEA can fit well within a more complex machine learning technique. More specifically, we prove that the FDH and DEA estimators are always feasible production functions generated by the optimization model linked to SVF, although they are not necessarily the optimal ones, except under very restrictive conditions. To prove that, we introduce a specific input transformation function that transforms the original regressor hyperplane in order to provide a suitable step function as an estimation surface. In this way, the SVF method offers an FDH-type estimator. At the same time, the convex hull of our step function yields a convex estimation of the production possibility set, therefore, generating a DEA-type estimator. The notion of ε-insensitive technical efficiency will be also introduced in this paper. This concept is directly related to the estimated (SVM) margin through cross-validation and endows the traditional notion of efficiency with more robustness than usual. Additionally, the performance of SVF is checked via Monte Carlo simulations, showing that the new approach significantly reduces the mean squared error and bias associated with the estimation of the true frontier in comparison with standard FDH and DEA techniques.

As far as we are aware, this paper represents the first contribution that really adapts the SVM technique, from a methodological point of view, for estimating production functions following the usual postulates in production theory. The previous contributions that considered these two types of worlds, SVM and production frontiers, did not really redefine SVM but combined it with the calculation of the standard DEA model at some stage. This means that the final efficiency estimations generated by these approaches inherit the same overfitting problems of traditional DEA. Examples of that are Song and Zhang (2009), where, in a first stage, DEA is used to evaluate the efficiency scores of a set of oil refining enterprises (DMUs), and, in a second stage, traditional SVM for regression is applied on a database consisting of all inputs and outputs as predictors and the DEA efficiency score as response variable; Yeh et al. (2010), where the objective is predicting business failure (the response variable) using the DEA efficiency score and other information as predictive variables; Poitier and Cho (2011) use standard SVM for regression to predict the DEA score of best and worst DMUs in the data sample and aggregate all the information through an average function; Kao et al. (2013) use a similar methodology to that

introduced by Song and Zhang (2009) in a multiclass classification problem, where each class is associated with a certain level of technical efficiency (estimated again by applying standard DEA); another similar paper is Farahmand et al. (2014), which applies an analogous methodology to that of Song and Zhang (2009); or, finally, Chen et al. (2018), where in a first stage, within the ambit of the satisficing Data Envelopment Analysis model, the probabilities of achieving a minimal performance threshold are computed, and in a second subsequent stage, SVM regression is applied to discriminate between high/low efficiency groups within each performance threshold.

The paper is organized as follows. Section 2 is devoted to briefly introducing the backgrounds: the FDH and DEA approaches, and the standard SVM technique. In Section 3, we extend Support Vector Machines to the context of estimating production functions, developing a new technique called Support Vector Frontiers (SVF). Performance of SVF is investigated via Monte Carlo simulations in Section 4. Finally, Section 5 concludes.

2. Background

In this section, we briefly review the main notions related to Free Disposal Hull, Data Envelopment Analysis and Support Vector Machines. Additionally, we will need to introduce some notation.

2.1. Free Disposal Hull (FDH)

Let us consider the observation of n Decision Making Units (DMUs). DMU_i consumes $\boldsymbol{x}_i = \left(x_i^{(1)},...,x_i^{(m)}\right) \in R_+^m$ amounts of inputs for the production of $\boldsymbol{y}_i = \left(y_i^{(1)},...,y_i^{(s)}\right) \in R_+^s$ amounts of outputs[1]. The relative efficiency of each DMU in the sample is assessed with reference to the so-called production possibility set or technology, which is the set of technically feasible combinations of $(\boldsymbol{x},\boldsymbol{y})$. It is defined in general terms as:

$$T = \left\{(\boldsymbol{x},\boldsymbol{y}) \in R_+^{m+s} : \boldsymbol{x} \text{ can produce } \boldsymbol{y}\right\} \tag{1}$$

Certain assumptions are done on this set, such as free disposability of inputs and outputs, meaning that if $(\boldsymbol{x},\boldsymbol{y}) \in T$, then $(\boldsymbol{x}',\boldsymbol{y}') \in T$, as long as $\boldsymbol{x}' \geq \boldsymbol{x}$ and $\boldsymbol{y}' \leq \boldsymbol{y}$[2]. Often convexity of T is also assumed (see, e.g., Färe and Primont, 1995).

When $s = 1$, this context is restricted to the key notion of production function. Accordingly, m input variables are used to produce a univariate output. In this way, the technology is defined as

$$T = \left\{(\boldsymbol{x},y) \in R_+^{m+1} : y \leq f(\boldsymbol{x})\right\} \tag{2}$$

In this context, the property of monotonicity is translated so that the production function f is supposed to be monotone non-decreasing: if $\boldsymbol{x} \leq \boldsymbol{x}'$, then $f(\boldsymbol{x}) \leq f(\boldsymbol{x}')$. Hereinafter, we will focus our attention on the estimation of technical efficiency in the contexts of production functions.

[1] We use bold for denoting vectors, and non-bold for scalars. The different components of a vector are denoted by a superscript and parenthesis.

[2] Let $\boldsymbol{z} = \left(z^{(1)},...,z^{(m)}\right)$ and $\boldsymbol{q} = \left(q^{(1)},...,q^{(m)}\right)$. Hereinafter, $\boldsymbol{z} \leq \boldsymbol{q}$ means $z^{(j)} \leq q^{(j)}$ for all $j = 1,...,m$.

As far as the measurement of technical efficiency was the concern, a certain part of the boundary of T is of interest. Specifically, we are referring to the efficient frontier of T, defined as $\partial(T) := \{(\boldsymbol{x}, y) \in T : y = f(\boldsymbol{x})\}$. In this way, technical inefficiency is defined as the distance from an interior point to this boundary. If $(\boldsymbol{x}', y') \in T$, then its technical efficiency score is determined by the ratio $y' / f(\boldsymbol{x}')$.

Nowadays, there are two main non-parametric methods based on envelopment techniques for estimating the efficient frontier of T: FDH and DEA. The FDH estimator was introduced by Deprins et al. (1984) and relies only on the free disposability (monotonicity) assumption and the minimal extrapolation principle. In contrast, the DEA estimator requires stronger assumptions, such as convexity of the set T (see subsection 2.2). The convexity assumption is widely used in Economics, but it is not always valid (Kerstens et al., 2019). The production possibility set might admit increasing returns to scale (i.e. output increases faster than the inputs, which graphically cannot be modelled by convexity), or there might be lumpy goods (i.e. fractional values of inputs or outputs do not exist). Hence, the FDH can yield a more general and flexible estimator than DEA (see Aragon et al., 2005).

The non-parametric models, such as FDH, are particularly appealing since they do not rely on restrictive hypothesis on the Data Generating Process, a feature shared with usual machine learning techniques, which are clearly data-driven approaches. In particular, Deprins et al. (1984) proposed the Free Disposal Hull of the set of observations (DMUs) to estimate T, defined as follows:

$$\hat{T}_{FDH} = \left\{(\boldsymbol{x}, \boldsymbol{y}) \in R_+^{m+s} : \boldsymbol{y} \leq \boldsymbol{y}_i, \boldsymbol{x} \geq \boldsymbol{x}_i, i = 1, ..., n\right\}. \tag{3}$$

In the univariate output case, the production function would be estimated by $f_{FDH}(\boldsymbol{x}) = \max_{i:\boldsymbol{x} \geq \boldsymbol{x}_i} \{y_i\}$. Next, we present a graphical example of the FDH estimator of a production function, showing its typical non-decreasing step shape.

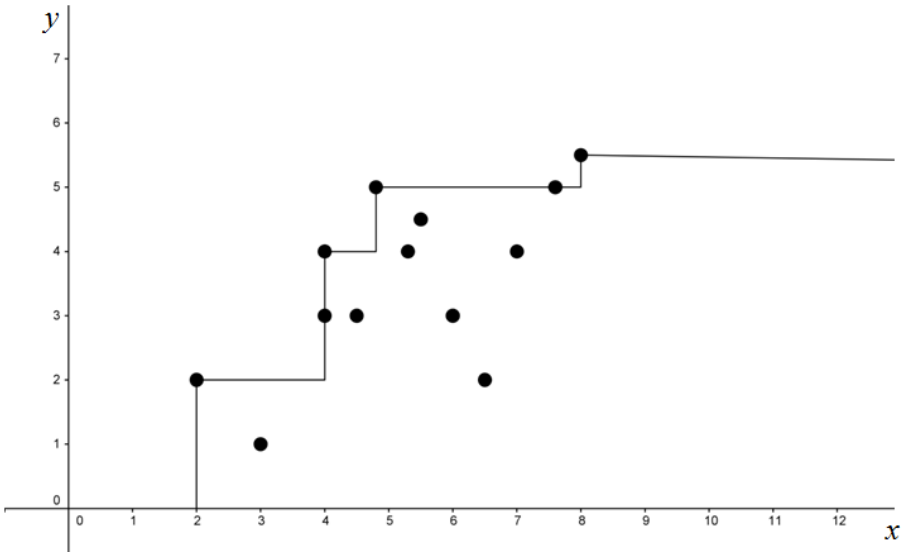


Figure 1: Example of standard FDH

The FDH technique is very engaging because it relies on very few assumptions, but, by construction, it suffers overfitting due to the minimal extrapolation principle. This principle forces the non-decreasing step function to be as close as possible to the data cloud, whereas it (upper) envelops all the observations. Note, in Figure 1, that the estimated frontier by FDH fits like a glove to the data sample, thereby additional satisfying monotonicity.

2.2. Data Envelopment Analysis (DEA)

In contrast to Free Disposal Hull, Data Envelopment Analysis additionally assumes convexity. This means that if $(\boldsymbol{x}, y)$ and $(\boldsymbol{x}', y')$ belong to T, then $\lambda(\boldsymbol{x}, y)+(1-\lambda)(\boldsymbol{x}', y') \in T$, for all $\lambda \in [0,1]$. Banker et al. (1984) proposed the DEA estimator of the production possibility set T as follows:

$$\hat{T}_{DEA} = \left\{ (\boldsymbol{x}, \boldsymbol{y}) \in R_+^{m+s} : y^{(r)} \leq \sum_{i=1}^{n} \lambda_i y_i^{(r)}, \forall r, x^{(j)} \geq \sum_{i=1}^{n} \lambda_i x_i^{(j)}, \forall j = 1, \sum_{i=1}^{n} \lambda_i = 1, \lambda_i \geq 0, \forall i \right\}. \quad (4)$$

Next, we present a graphical example of the DEA estimator of a production function (a single output), showing its typical piece-wise linear shape. Convexity of the production possibility set (the shaded area) implies, in this case, concavity of the production function. Note also that the convexification of the FDH estimator yields the DEA estimator (Daraio and Simar, 2005).

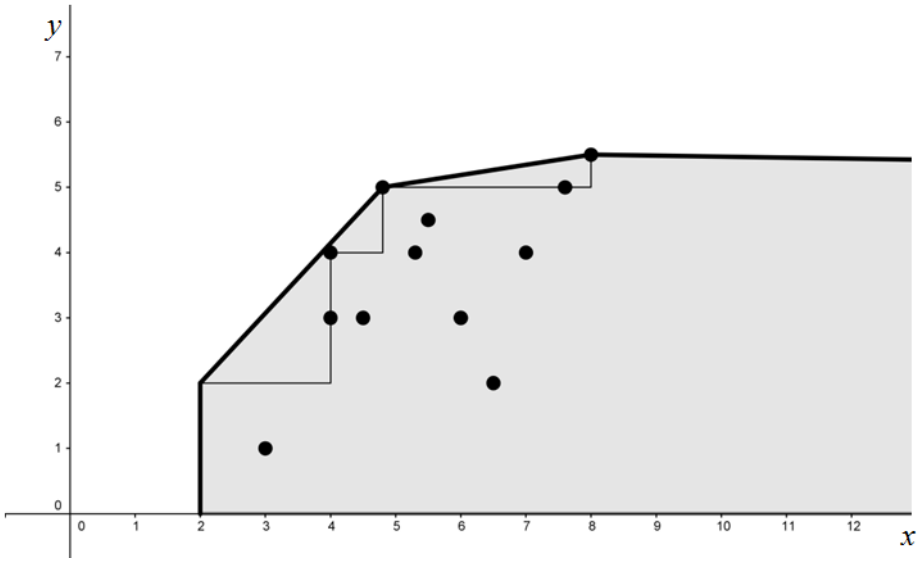


Figure 2: Example of standard DEA

As can be noted from the figure, DEA suffers the same problem of overfitting as FDH. Again, the minimal extrapolation principle compels the determination of a 'perfect' description of the observed sample at a frontier level. It is done by calculating the smallest convex set that contains the data cloud and satisfies free disposability (monotonicity).

2.3. Support Vector Regression (SVR)

Support Vector Machines (Cortes and Vapnik, 1995, Vapnik, 1995, 1998) is a machine learning technique with two objectives, depending on the nature of the response variable. SVM works like a non-parametric classification model when the response variable is categorical and as a regression model when it is numerical. In

this paper, we focus on the latter approach, due to the nature of our response variable (the produced output by a firm).

From a theoretical viewpoint, SVM is a constructive learning procedure grounded on statistical learning theory and the principle of structural risk minimization. It aims to minimize the bound on the generalization error instead of exclusively minimizing the empirical error such as the mean square error over the data set (Vapnik, 1995, 1998). This results in good generalization capability: SVM tends to perform well when applied to data outside the training set. Because SVM is suitable for dealing with a limited number of samples, regardless of the dimension of the examples (the number of feature variables), it has been widely used in machine learning, data mining, pattern recognition, function approximation, regression, etc.

Support Vector Regression (SVR) is a particular model in the family of Support Vector Machines. As the rest of the regression procedures, SVR tries to construct a function that predicts the behavior of the response variable that is involved in the study, receiving, in this case, the main benefits of machine learning. Standard Support Vector Regression is constructed through a set of techniques whose aim is to predict the value of a response variable $y \in \mathbb{R}$ given a vector of covariables $\boldsymbol{x} \in \mathbb{R}_+^m$. On this way, SVR sets a function $\hat{f} : \mathbb{R}_+^m \to \mathbb{R}$ such that $\hat{f}(\boldsymbol{x}) = \hat{y}$, where $\hat{y}$ constitutes the prediction of the response variable. Under SVR, the predictor $\hat{f}$ is defined as $\hat{f}(\boldsymbol{x}) = \boldsymbol{w}^* \phi(\boldsymbol{x}) + b^*$, where $\boldsymbol{w}^* \in \mathbb{R}^q$ and $b^* \in \mathbb{R}$ are optimal solutions of model (5), $\phi(\cdot)$ is a transformation function of the covariable space and the values of $C \in \mathbb{R}_+$ and $\varepsilon \in \mathbb{R}_+$ are obtained by a cross-validation process.

$$
\begin{aligned}
\underset{\boldsymbol{w},b,\xi_i',\xi_i}{Min} \quad & \|\boldsymbol{w}\| + C\sum_{i=1}^{n}\left(\xi_i' + \xi_i\right) & & \\
& y_i - \left(\boldsymbol{w}\phi(\boldsymbol{x}_i) + b\right) \le \varepsilon + \xi_i', & i = 1,\ldots,n & \quad (5.1) \\
& \left(\boldsymbol{w}\phi(\boldsymbol{x}_i) + b\right) - y_i \le \varepsilon + \xi_i, & i = 1,\ldots,n & \quad (5.2) \\
& \xi_i', \xi_i \ge 0, & i = 1,\ldots,n & \quad (5.3)
\end{aligned}
\qquad (5)
$$

Note that $\hat{f}(\boldsymbol{x}) = \boldsymbol{w}^* \phi(\boldsymbol{x}) + b^*$ has the structure of a hyperplane in the transformed space $(\phi(\boldsymbol{x}), y)$. The SVR generates a predictor $\hat{f}(\boldsymbol{x})$ of the response variable for vector $\boldsymbol{x}$ as well as lower and upper 'correcting' surfaces, defined as $\hat{f}(\boldsymbol{x}) - \varepsilon$ and $\hat{f}(\boldsymbol{x}) + \varepsilon$, where ε is a certain margin that endows the estimator linked to SVR with robustness (see Figure 3). Additionally, observations below the surface $\hat{f}(\boldsymbol{x}) - \varepsilon$ have an associated (empirical) error of $\xi_i > 0$ (with $\xi_i' = 0$), while observations above the surface $\hat{f}(\boldsymbol{x}) + \varepsilon$ present an (empirical) error of $\xi_i' > 0$ (with $\xi_i = 0$). Observations between the surfaces $\hat{f}(\boldsymbol{x}) - \varepsilon$ and $\hat{f}(\boldsymbol{x}) + \varepsilon$ have an error of zero (with $\xi_i = \xi_i' = 0$). Regarding the objective function in (5), it represents the combination of regression and regularization that SVR involves, mixing the empirical error term $\sum_{i=1}^{n}\left(\xi_i' + \xi_i\right)$ and the regularization term $\|\boldsymbol{w}\|$ through a weight C, which balances the two components (Vazquez and Walter, 2003). Moreover, although hyperplanes have linear shapes, SVR is able to yield prediction functions that are not necessarily linear in the

original $(\boldsymbol{x}, y)$ space (see Figure 3). It is due to the function ϕ, which is a transformation of the covariable space, $\phi: \mathbb{R}_+^m \to Z$. It is important to remark that $Z = \phi\left(\mathbb{R}_+^m\right)$, known as feature space, can have different dimensions, even infinite. This transformation is the cause of the different nonlinear approaches in $\mathbb{R}_+^m$, while the prediction function generates a linear hyperplane in Z.

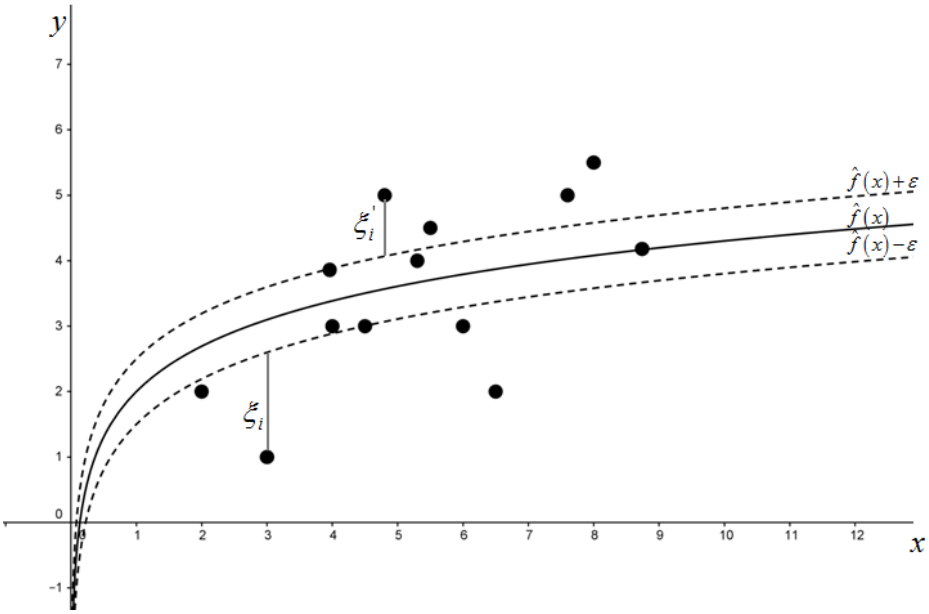


Figure 3. Example of the standard SVR

Sometimes, in the literature, it is easier or advisable to work with the dual model of program (5) and kernel functions, defined as $K(\boldsymbol{x}, \boldsymbol{x}') = \phi(\boldsymbol{x})\phi(\boldsymbol{x}')$. However, in this paper, the primal model and the transformation function $\phi(\cdot)$ will be enough for the development of the new approach.

Both the transformation function and the kernel function depend on several hyperparameters, say τ, which must be determined together with C and ε by cross-validation. Cross-validation is a standard technique in machine learning and statistics for adjusting hyperparameters of predictive models. In V-fold cross-validation, the learning sample Ω is randomly divided into $\Omega_1, ..., \Omega_V$ disjoint subsamples with the same sample size or as close as possible. Typical values for V are 5 or 10 (Friedman et al., 2001). Let the v-th learning subsample be $\Omega^{(v)} = \Omega - \Omega_v$ and let $\Omega_i = \Omega - \Omega_{v(i)}$, where $\Omega_{v(i)}$ is the subsample such that $i \in \Omega_{v(i)}$. For each $v = 1, ..., V$, an individual model is built by applying the algorithm to the training data $\Omega^{(v)}$. This model is then evaluated by means of a cost function using the test data in Ω_v. In particular, given a set of hyperparameters (C, ε, τ), the prediction error is determined as $\frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{f}_{\Omega_i}(\boldsymbol{x}_i; C, \varepsilon, \tau)\right)^2$, where $\hat{f}_{\Omega_i}(\boldsymbol{x}_i; C, \varepsilon, \tau)$ is the SVR predictor evaluated at $\boldsymbol{x}_i$ when the learning data Ω_i and the hyperparameters (C, ε, τ) have been considered. To choose the best combination of hyperparameters using cross-validation, we compute the prediction error for different values of hyperparameters (C, ε, τ). Finally, one selects the combination $(C^*, \varepsilon^*, \tau^*)$ with the lowest prediction error and uses it for training a SVR model on the complete data set Ω.

3. Support Vector Regression for estimating production functions

In this section, we introduce a new technique based on a methodological adaptation of the standard SVR for the estimation of production functions, which will be denominated Support Vector Frontiers (SVF). The new approach will allow the determination of production functions, satisfying the usual axioms of microeconomics, through a data-driven approach that does not assume any particular random distribution on the data and generates, at a first stage, a step function as a predictor. At a second stage, the convexification of this step function will allow us to provide a piece-wise linear predictor of the production function. These shapes are shared with the estimators derived from FDH and DEA techniques. However, while the latter suffer the problem of overfitting, the new method will try to overcome this drawback through minimizing the structural risk as SVR does. To start with, we will show how the standard SVR model must be modified in order to estimate (upper) enveloping surfaces, satisfying certain classical axioms in production theory (see, for example, Färe et al., 1985): (A1) if $\boldsymbol{x} = \boldsymbol{0}_m$, then $f(\boldsymbol{x}) = 0$; (A2) if $\boldsymbol{x} \leq \boldsymbol{x}'$, then $f(\boldsymbol{x}) \leq f(\boldsymbol{x}')$; (A3) f is a concave function. Postulate A1 means that if a firm does not consume any input, it cannot produce anything. Axiom A2 means that if a firm consumes more resources, then, it is always possible to produce more output; while A3 is associated with the convexity of the production possibility set T in (2).

In order to be clear in our exposition, we will adapt the standard SVR model step by step in this section. It means that we will incorporate the satisfaction of each production axiom A1-A3 gradually throughout the text. Additionally, we will show the existing relationship between FDH and DEA and the new machine learning framework. In fact, it is one of our objectives to show under which hypothesis these two standard techniques in efficiency measurement may be reinterpreted as SVM models. Finally, we will introduce a new robust definition of technical efficiency, called ε-insensitive technical efficiency, as a natural application of the notion of margin in SVM to the efficiency measurement field.

3.1. The estimation of enveloping surfaces

In this subsection, we will show how to adapt the standard SVR model to estimate upper enveloping structures of the data. Additionally, we will adjust trivially the model to meet A1.

By the Data Generating Process mentioned in the Introduction, we know that our target function to be estimated, f, and the response variable (the output in our production context) have the following relationship with respect to each learning example i, $i = 1,...,n$: $y_i \leq f(\boldsymbol{x}_i)$. Therefore, it seems natural to force the predictor to meet the same association and add it to the constraints of the SVR model (5): $y_i \leq \hat{f}(\boldsymbol{x}_i) = \boldsymbol{w}\boldsymbol{\phi}(\boldsymbol{x}_i) + b$, $i = 1,...,n$. Nevertheless, this new type of restriction directly implies the satisfaction of the constraint (5.1), $y_i - (\boldsymbol{w}\boldsymbol{\phi}(\boldsymbol{x}_i) + b) \leq \varepsilon + \xi_i'$, since $y_i \leq \boldsymbol{w}\boldsymbol{\phi}(\boldsymbol{x}_i) + b \Leftrightarrow y_i - (\boldsymbol{w}\boldsymbol{\phi}(\boldsymbol{x}_i) + b) \leq 0$ and, additionally, we have that $\varepsilon + \xi_i' \geq 0$, for all $i = 1,...,n$. Consequently, constraint type (5.1) can be removed from the new optimization model if the conditions $y_i \leq \boldsymbol{w}\boldsymbol{\phi}(\boldsymbol{x}_i) + b$, $i = 1,...,n$, are incorporated. And, for the same reason, the decision variables ξ_i', $i = 1,...,n$, can be also deleted. This has geometrical implications regarding the margin: the upper

correcting surface disappears. It really collapses to the predictor surface $\hat{f}(\boldsymbol{x})$, which now upper envelops the data cloud.

Moreover, a simple manner of satisfying axiom A1 consists of forcing the term b to be directly zero as long as $\phi(\mathbf{0}_m)=\mathbf{0}_q$. It guarantees that $\hat{f}(\mathbf{0}_m)=\boldsymbol{w}\phi(\mathbf{0}_m)+b=0$. In this way, at this point, the adapted SVR model would be as follows:

$$\begin{aligned} \underset{\boldsymbol{w},\xi_i}{Min} \quad & \|\boldsymbol{w}\|+C\sum_{i=1}^{n}\xi_i \\ & y_i-\boldsymbol{w}\phi(\boldsymbol{x}_i)\le 0, \qquad i=1,\ldots,n \quad (6.1) \\ & \boldsymbol{w}\phi(\boldsymbol{x}_i)-y_i\le \varepsilon+\xi_i, \quad i=1,\ldots,n \quad (6.2) \\ & \xi_i\ge 0, \qquad i=1,\ldots,n \quad (6.3) \end{aligned} \tag{6}$$

The predictor linked to the modified SVR model is equivalent to that associated with the standard SVR model, but now with a new meaning of upper 'frontier' of the data cloud thanks to constraint (6.1).

Regarding the shape of the predictor, it is worth mentioning that standard efficiency measurement approaches like FDH and DEA resort to certain particular types of surfaces. FDH uses step functions and DEA resorts to the convexification of the FDH step function to get a concave predictor of the production function. Although, in this respect, standard SVM is really flexible, allowing to use several complex transformation families of functions (polynomial, radial, Laplacian kernels, among others), we will focus our attention in this paper on a step function as the transformation function $\phi(\cdot)$, seeking to establish a bridge between the standard approaches for measuring technical efficiency and the SVR method. In this way, and by analogy with FDH and DEA, we suggest in this paper to apply a transformation function in such a way that our estimator is a step function, which could be subsequently transformed into a concave function by convexification. To do that, we will base our approach on the spline approximation of the predictor constructed from a finite number of knots by Vapnik (1998, p. 464).

Next, we introduce the necessary definitions and notation for establishing the transformation function $\phi_{SVF}(\cdot)$ that will be used in this paper. First, we introduce the notion of knot and how the input space will be split.

Definition 1 [Knots]. For each input dimension j, $j=1,...,m$, the set of knots is defined as $T_j=\left\{t_{l_j}^{(j)}: l_j=1,...,k_j\right\}$ satisfying $0<t_1^{(j)}<t_2^{(j)}<...<t_{k_j}^{(j)}$.

Definition 2 [Grid]. The disjoint subsets $C_{1...1},...,C_{k_1...k_m}$ such that

$$C_{l_1...l_m}:=\left\{\boldsymbol{x}\in\mathbb{R}_+^m / t_{l_j}^{(j)}\le x^{(j)}<t_{l_j+1}^{(j)}, j=1,...m\right\},\ l_1\in\{1,...,k_1\},...,l_m\in\{1,...,k_m\}, \tag{7}$$

with $t_{k_j+1}^{(j)}:=\infty,\ \forall j=1,...,m$, define a grid G on $\mathbb{R}_+^m$. Each subset $C_{l_1...l_m}$ is called a cell of the grid G.

By Definition 2, the total number of cells of a grid G equals $k_1 \cdot k_2 \cdot \ldots \cdot k_m$, where k_j represents the maximum number of knots in the dimension j, $j = 1,...,m$. Moreover, each cell in the grid may be characterized by two 'extreme' points, introduced in the following definition.

Definition 3 [Extreme knot-points of a cell]. For each cell $C_{l_1 \ldots l_m}$ of a grid G, $\boldsymbol{a}_{l_1 \ldots l_m} = \left(t_{l_1}^{(1)}, \ldots, t_{l_m}^{(m)}\right)$ and $\boldsymbol{b}_{l_1 \ldots l_m} = \left(t_{l_1+1}^{(1)}, \ldots, t_{l_m+1}^{(m)}\right)$ are the lower extreme knot-point and the upper extreme knot-point, respectively.

By Definition 3, each cell can be equivalently rewritten as $C_{l_1 \ldots l_m} = \left\{ \boldsymbol{x} \in \mathbb{R}_+^m \,/\, a_{l_1 \ldots l_m}^{(j)} \le x^{(j)} < b_{l_1 \ldots l_m}^{(j)}, j = 1,...m \right\}$.

Now, given an input vector $\boldsymbol{x} \in \mathbb{R}_+^m$, we need to identify the cell where it is located. The following binary 'activation' function defined for each cell $C_{l_1 \ldots l_m}$ will indicate with a value of one if $\boldsymbol{x} \in C_{l_1 \ldots l_m}$.

Definition 4 [Activation function of a cell $C_{l_1 \ldots l_m}$]. A function $L_{l_1 \ldots l_m} : \mathbb{R}_+^m \to \{0,1\}$ defined as

$$\boldsymbol{x} \to L_{l_1 \ldots l_m}(\boldsymbol{x}) = \prod_{j=1}^{m} B\left(\boldsymbol{x}^{(j)} - t_{l_j}^{(j)}\right), \tag{8}$$

where
$$B\left(\boldsymbol{x}^{(j)} - t_{l_j}^{(j)}\right) = \begin{cases} 1, & if \quad \boldsymbol{x}^{(j)} - t_{l_j}^{(j)} \ge 0 \\ 0, & if \quad \boldsymbol{x}^{(j)} - t_{l_j}^{(j)} < 0 \end{cases}, \forall l_j = 1, \ldots, k_j, \forall j = 1, \ldots, m, \tag{9}$$

is the activation function associated with the cell $C_{l_1 \ldots l_m}$ of the grid G.

Note also that, by Definition 4, when a cell $C_{l_1 \ldots l_m}$ is activated, then all the cells $C_{s_1 \ldots s_m}$ with $s_j \le l_j$, $j = 1,...,m$, are also activated, i.e., if $L_{l_1 \ldots l_m}(\boldsymbol{x}) = 1$, then $L_{s_1 \ldots s_m}(\boldsymbol{x}) = 1$, $\forall s_j \le l_j$, $j = 1,...,m$. We will say that all these cells are cells dominated by $C_{l_1 \ldots l_m}$. Figure 4 shows how the activation of cells in a grid of $\mathbb{R}_+^m$ for two input dimensions works.

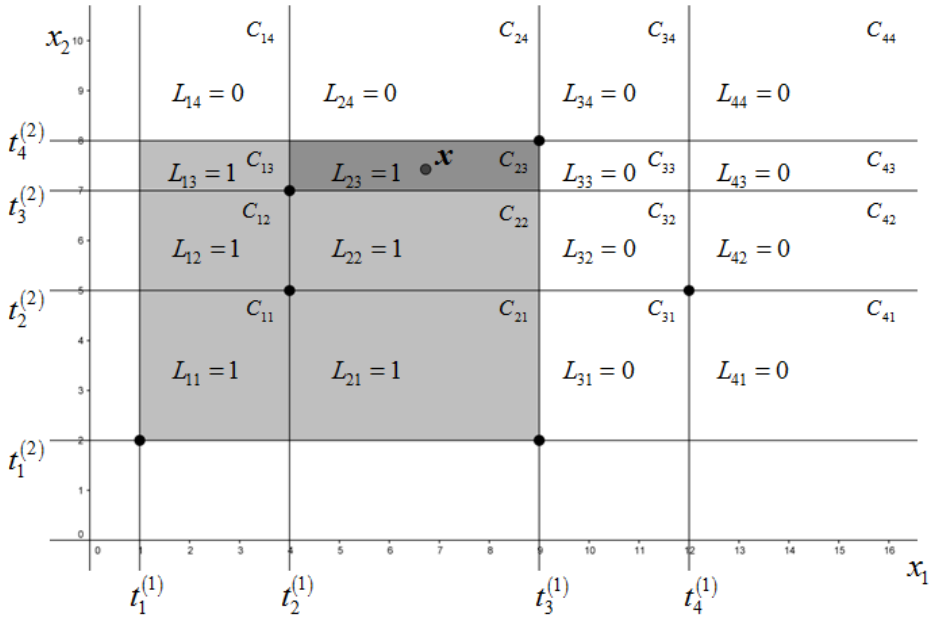


Figure 4. Example of the activation of cells in a grid

In this context, given a grid G, we define the transformation function ϕ_{SVF}^{G}, which will be used in our approach, as the mapping of the space $\mathbb{R}_{+}^{m}$ given by

$$\boldsymbol{x} \to \phi_{SVF}^{G}(\boldsymbol{x}) = \left(L_{11..11}(\boldsymbol{x}), L_{11..12}(\boldsymbol{x}), \ldots, L_{11..1k_m}(\boldsymbol{x}), L_{11..21}(\boldsymbol{x}), L_{11..22}(\boldsymbol{x}), \ldots, L_{11..2k_m}(\boldsymbol{x}), \ldots, L_{k_1 k_2 \ldots k_m}(\boldsymbol{x})\right) \tag{10}$$

For each input vector $\boldsymbol{x} \in \mathbb{R}_{m}^{+}$, $\phi_{SVF}^{G}(\boldsymbol{x})$ corresponds to a vector of ones and zeros generated from the $k_1 \cdot k_2 \cdot \ldots \cdot k_m$ activation functions on grid G. Following our previous discussion, the components of the vector $\phi_{SVF}^{G}(\boldsymbol{x})$ value one for the cell where $\boldsymbol{x}$ is located and all its corresponding dominated cells. In this way, at this point, the adapted SVR model for estimating production functions would be as follows:

$$\begin{aligned}
\underset{\boldsymbol{w},\xi_i}{Min} \quad & \|\boldsymbol{w}\| + C\sum_{i=1}^{n}\xi_i && && (11.0)\\
& y_i - \boldsymbol{w}\phi_{SVF}^{G}(\boldsymbol{x}_i) \le 0, && i=1,\ldots,n && (11.1)\\
& \boldsymbol{w}\phi_{SVF}^{G}(\boldsymbol{x}_i) - y_i \le \varepsilon + \xi_i, && i=1,\ldots,n && (11.2)\\
& \xi_i \ge 0, && i=1,\ldots,n && (11.3)
\end{aligned} \tag{11}$$

The predictor linked to (11) would be $\hat{f}(\boldsymbol{x}) = \boldsymbol{w}\phi_{SVF}^{G}(\boldsymbol{x})$. Note that $\hat{f}(\boldsymbol{0}_m) = 0$ since $\phi_{SVF}^{G}(\boldsymbol{0}_m) = \boldsymbol{0}_q$, with $q = k_1 \cdot k_2 \cdot \ldots \cdot k_m$. It guarantees the satisfaction of axiom A1. Moreover, by the definition of the transformation function, if $\boldsymbol{x} \in C_{l_1 \ldots l_m}$, then we have that $\hat{f}(\boldsymbol{x}) = \boldsymbol{w}\phi_{SVF}^{G}(\boldsymbol{x}) = \sum_{\substack{s_1=1,\ldots,l_1 \\ \vdots \\ s_m=1,\ldots,l_m}} w_{s_1 \ldots s_m}$.

Additionally, it is worth noting that the number of components of the vector of parameters $\boldsymbol{w}$ coincides with the total number of cells of the corresponding grid G. Explicitly, the parametric vector $\boldsymbol{w}$ can be written as follows.

$$\boldsymbol{w}=\left(w_{11..11},w_{11..12},\ldots,w_{11..1k_m},w_{11..21},w_{11..22},\ldots,w_{11..2k_m},\ldots,w_{k_1k_2..k_m}\right) \tag{12}$$

Each component of $\boldsymbol{w}$ is linked to a cell on the grid and the order of the components is the same as that followed by the vector $\phi_{SVF}^{G}(\boldsymbol{x})$ in (10). This relationship allows introducing the notion of a parametrized grid.

Definition 5 [Parameterized grid]. A grid G in which is assigned a parameter $w_{l_1\ldots l_m}\in\mathbb{R}$ for each cell $C_{l_1\ldots l_m}$, $l_j=1,\ldots,k_j, j=1,\ldots,m$, is a parameterized grid with parameters $\boldsymbol{w}$.

Next, we are going to show that the predictor derived from model (11) is a step function and, additionally, any step function $f(\boldsymbol{x})$ defined on a grid G may be equivalently rewritten in the way $f(\boldsymbol{x})=\boldsymbol{w}\phi_{SVF}^{G}(\boldsymbol{x})$ for a certain parametric vector $\boldsymbol{w}$. Nevertheless, we first need to introduce the definition of a step function on a grid and the notion of a recoverable function through a parametrized grid.

Definition 6 [Step function on a grid]. A function $f:\mathbb{R}_{+}^{m}\to\mathbb{R}$ is a step function on a grid G if $f(\boldsymbol{x})$ is constant $\forall\boldsymbol{x}\in C_{l_1\ldots l_m}$, $\forall l_1=1,...,k_1$, …, $\forall l_m=1,...,k_m$. The value of the function f for any $\boldsymbol{x}\in C_{l_1\ldots l_m}$ will be denoted as $f\left(C_{l_1\ldots l_m}\right)$.

Definition 7 [Recoverable function through a parameterized grid]. A function $f:\mathbb{R}_{+}^{m}\to\mathbb{R}$ is a recoverable function through a parameterized grid G with parameters $\boldsymbol{w}$ if and only if $\forall\boldsymbol{x}\in\mathbb{R}_{+}^{m}$ $f(\boldsymbol{x})=\boldsymbol{w}\phi_{SVF}^{G}(\boldsymbol{x})=\sum_{\substack{s_1=1,\ldots,l_1\\ \vdots\\ s_m=1,\ldots,l_m}} w_{s_1\ldots s_m}$, with $\boldsymbol{x}\in C_{l_1\ldots l_m}$.

Given a grid G, it is clear that $\hat{f}(\boldsymbol{x})=\boldsymbol{w}\phi_{SVF}^{G}(\boldsymbol{x})=\sum_{\substack{s_1=1,\ldots,l_1\\ \vdots\\ s_m=1,\ldots,l_m}} w_{s_1\ldots s_m}$ is constant for each $\boldsymbol{x}\in C_{l_1\ldots l_m}$.

Consequently, the predictor derived from (11) is a step function defined on G. Additionally, it can be proved that, given a step function $f(\boldsymbol{x})$ defined on a grid G, there will always be parameters $\boldsymbol{w}$ such that $f(\boldsymbol{x})$ may be equivalently expressed as $f(\boldsymbol{x})=\boldsymbol{w}\phi_{SVF}^{G}(\boldsymbol{x})=\sum_{\substack{s_1=1,\ldots,l_1\\ \vdots\\ s_m=1,\ldots,l_m}} w_{s_1\ldots s_m}$.

Proposition 1. Any step function on a grid G is a recoverable function through the parameterized grid G for a certain unique parametric vector $\boldsymbol{w}$.

Proof. Let $f:\mathbb{R}_{+}^{m}\to\mathbb{R}$ a step function on a grid G. Then, we have that $f(x)=f\left(C_{l_1\ldots l_m}\right)\ \forall x\in C_{l_1\ldots l_m}$, $\forall l_1=1,...,k_1,...,\forall l_m=1,...,k_m$. Now, we need to find out whether the following system has a solution or not.

$$\sum_{\substack{s_1=1,\ldots,l_1\\ \vdots\\ s_m=1,\ldots,l_m}} w_{s_1\ldots s_m}=f\left(C_{l_1\ldots l_m}\right), l_1=1,...,k_1,...,l_m=1,...,k_m \tag{13}$$

The expression in (13) represents $k_1 \cdot ... \cdot k_m$ equations. The equations corresponding to $l_1 = l_2 = ... = l_{m-1} = 1$ and $l_m = 1,...,k_m$ are as follows.

$$\begin{cases} w_{1...11} & & & & & = & f\left(C_{1...11}\right) \\ w_{1...11} & + & w_{1...12} & & & = & f\left(C_{1...12}\right) \\ & & & \vdots & & & \\ w_{1...11} & + & w_{1...12} & + \ \ldots \ + & w_{1...1k_m} & = & f\left(C_{1...1k_m}\right) \end{cases} \tag{14}$$

In matrix format, (14) can be written as (15):

$$\underbrace{\begin{pmatrix} 1 & 0 & & 0 \\ 1 & 1 & & 0 \\ \vdots & \vdots & \ddots & \\ 1 & 1 & \ldots & 1 \end{pmatrix}}_{A_{1...1l_m}} \underbrace{\begin{pmatrix} w_{1...11} \\ w_{1...12} \\ \vdots \\ w_{1...1k_m} \end{pmatrix}}_{\tilde{W}_{1...1l_m}} = \underbrace{\begin{pmatrix} f\left(C_{1...11}\right) \\ f\left(C_{1...12}\right) \\ \vdots \\ f\left(C_{1...1k_m}\right) \end{pmatrix}}_{F\left(C_{1...1l_m}\right)}, \tag{15}$$

with $\left|A_{1...1l_m}\right| = 1$ because $A_{1...1l_m}$ is a lower triangular matrix (the determinant of a triangular matrix is the product of the elements of its diagonal).

Regarding the system of equations for $l_1 = l_2 = ... = l_{m-2} = 1$, $l_{m-1} = 1,...,k_{m-1}$ and $l_m = 1,...,k_m$, we have the following format:

$$\underbrace{\begin{pmatrix} \underbrace{\begin{pmatrix} 1 & 0 & \ldots & 0 \\ 1 & 1 & \ldots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 1 & 1 & \ldots & 1 \end{pmatrix}}_{A_{1...1l_m}} & O & & \ldots & O \\ \underbrace{\begin{pmatrix} 1 & 0 & \ldots & 0 \\ 1 & 1 & \ldots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 1 & 1 & \ldots & 1 \end{pmatrix}}_{A_{1...1l_m}} & \underbrace{\begin{pmatrix} 1 & 0 & \ldots & 0 \\ 1 & 1 & \ldots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 1 & 1 & \ldots & 1 \end{pmatrix}}_{A_{1...2l_m}} & & \ldots & O \\ \vdots & \vdots & & \ddots & \\ \underbrace{\begin{pmatrix} 1 & 0 & \ldots & 0 \\ 1 & 1 & \ldots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 1 & 1 & \ldots & 1 \end{pmatrix}}_{A_{1...1l_m}} & \underbrace{\begin{pmatrix} 1 & 0 & \ldots & 0 \\ 1 & 1 & \ldots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 1 & 1 & \ldots & 1 \end{pmatrix}}_{A_{1...2l_m}} & & \ldots & \underbrace{\begin{pmatrix} 1 & 0 & \ldots & 0 \\ 1 & 1 & \ldots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 1 & 1 & \ldots & 1 \end{pmatrix}}_{A_{1...k_{m-1}l_m}} \end{pmatrix}}_{A_{1...1l_{m-1}l_m}} \underbrace{\begin{pmatrix} w_{1...11} \\ w_{1...12} \\ \vdots \\ w_{1...1k_m} \\ w_{1...21} \\ w_{1...22} \\ \vdots \\ w_{1...2k_m} \\ \vdots \\ w_{1...k_{m-1}1} \\ w_{1...k_{m-1}2} \\ \vdots \\ w_{1...k_{m-1}k_m} \end{pmatrix}}_{\tilde{W}_{1...1l_{m-1}l_m}} = \underbrace{\begin{pmatrix} f\left(C_{1...11}\right) \\ f\left(C_{1...12}\right) \\ \vdots \\ f\left(C_{1...1k_m}\right) \\ f\left(C_{1...21}\right) \\ f\left(C_{1...22}\right) \\ \vdots \\ f\left(C_{1...2k_m}\right) \\ \vdots \\ f\left(C_{1...k_{m-1}1}\right) \\ f\left(C_{1...k_{m-1}2}\right) \\ \vdots \\ f\left(C_{1...k_{m-1}k_m}\right) \end{pmatrix}}_{F\left(C_{1...1l_{m-1}l_m}\right)}, \tag{16}$$

with $\left|A_{1\ldots 1 l_{m-1} l_m}\right| = \prod_{s_{m-1}=1,\ldots,k_{m-1}} \left|A_{1\ldots 1 s_{m-1} l_m}\right| = 1$ because $A_{1\ldots 1 l_{m-1} l_m}$ is a lower triangular matrix by blocks, and the determinant of those kind of matrices is equal to the product of the determinants of the blocks of its diagonal.

Sequentially, we can write all $k_1 \cdot \ldots \cdot k_{m-1} \cdot k_m$ equations of the system given in (13) in matrix format as

$$A_{l_1\ldots l_{m-1} l_m} \cdot \tilde{W}_{l_1\ldots l_{m-1} l_m} = F\left(C_{l_1\ldots l_{m-1} l_m}\right) \tag{17}$$

with $\left|A_{l_1\ldots l_{m-1} l_m}\right| = \prod_{s_1=1,\ldots,k_1} \left|A_{s_1 l_2\ldots l_{m-1} l_m}\right| = 1$ because $A_{l_1\ldots l_{m-1} l_m}$ is a lower triangular matrix by blocks.

Given that $\left|A_{l_1\ldots l_{m-1} l_m}\right| = 1 \neq 0$, we have that $A_{l_1\ldots l_{m-1} l_m}$ is invertible. Hence, the system given in (13) is compatible and determined (it has a unique solution). ■

This result shows the flexibility of the transformation suggested in this paper, at least, with respect to any step function that can be defined on a grid in the input space. Later in the text, this proposition will be the key for showing that the step function associated with the FDH technique is always a feasible step function linked to the new approach, as long as we resort to a specific 'empirical' grid.

At this point, the predictor derived from model (11) is able to generate step functions upper enveloping the data cloud and satisfying axiom A1. However, the model can yield non-monotonic predictors, which are not valid when the objective is to determine a (production) function meeting axiom A2. In Figure 5, we show an example of this possible situation. Additionally, concavity of the predictor is not assured, which contradicts axiom A3. In Figure 5, this fact is linked to the non-convexity of the associated technology or production possibility set (the shaded area).

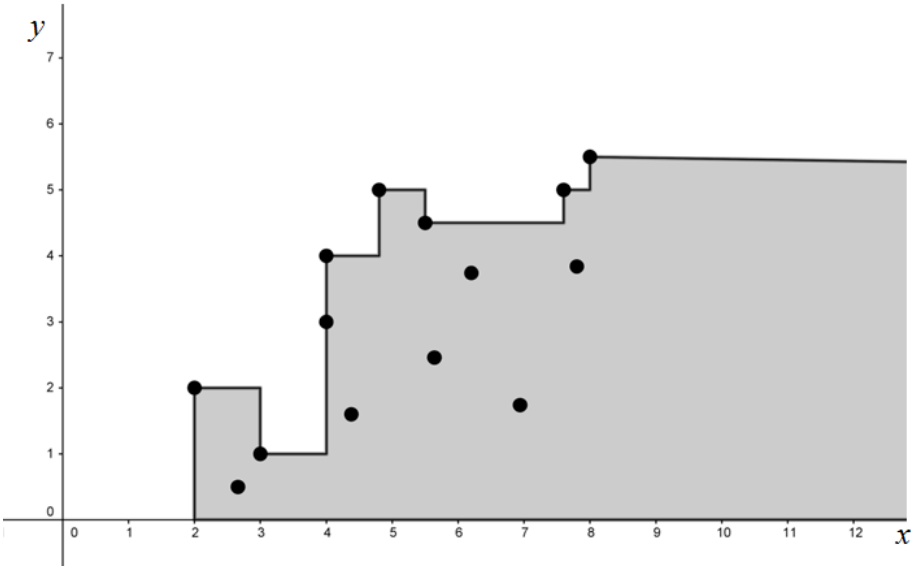


Figure 5. A possible predictor obtained from model (11)

We will try to overcome these drawbacks of the technique in the next section.

3.2. The axioms of monotonicity and concavity of the production function

Monotonicity and concavity are two basic axioms of production functions in microeconomics. Consequently, in this section, we try to endow the predictor derived from the new approach based on SVR with the satisfaction of these two properties. Accordingly, the first part of this subsection is devoted to defining the necessary conditions for assuring a monotonic non-decreasing step function as a predictor from the SVF approach. In the final part of this subsection, we will show how to trivially determine concave functions as predictors of the production functions by directly applying convexification of the previously defined step function.

We start by establishing the conditions that assure that the final predictor derived from SVF meets monotonicity for any recoverable function defined on a parametric grid G. To do that, we first need a technical lemma.

Lemma 1. Let G a grid and let $\boldsymbol{x} \in C_{h_1 \ldots h_m}$ and $\boldsymbol{z} \in C_{l_1 \ldots l_m}$ with $\boldsymbol{x} \leq \boldsymbol{z}$. Then, $h_j \leq l_j, \forall j = 1, \ldots, m$.

Proof. Let $\boldsymbol{x} \in C_{h_1 \ldots h_m}$ and $\boldsymbol{z} \in C_{l_1 \ldots l_m}$ with $\boldsymbol{x} \leq \boldsymbol{z}$. By reduction to the absurd, let us suppose that there exists $j' \in \{1, \ldots, m\}$ with $h_{j'} > l_{j'}$. By the definition of cell in grid a G, we have that $t_{h_{j'}}^{(j')} \leq x^{(j')} < t_{h_{j'}+1}^{(j')}$ and $t_{l_{j'}}^{(j')} \leq z^{(j')} < t_{l_{j'}+1}^{(j')}$. But note that if $h_{j'} > l_{j'}$, by the definition of knot in a grid G, then $t_{h_{j'}}^{(j')} \geq t_{l_{j'}+1}^{(j')}$. Therefore, $x^{(j')} \geq t_{l_{j'}+1}^{(j')}$. Hence, $x^{(j')} > z^{(j')}$ since $z^{(j')} < t_{l_{j'}+1}^{(j')}$, which is a contradiction with the hypothesis $\boldsymbol{x} \leq \boldsymbol{z}$. ■

Before showing the result on monotonicity, let us introduce the following notation: $W_{l_1 \ldots l_m} := \sum_{\substack{s_1 = 1, \ldots, l_1 \\ \vdots \\ s_m = 1, \ldots, l_m}} w_{s_1 \ldots s_m}$.

Theorem 1. Let $f : \mathbb{R}_+^m \to \mathbb{R}_+$ a recoverable function through the parameterized grid G. Then, f is a monotonic non-decreasing function if and only if

$$W_{l_1 \ldots l_m} \geq W_{h_1 \ldots h_m}, \forall h_j \leq l_j, \forall j = 1, \ldots, m, \text{ with } h_j, l_j = 1, \ldots, k_j. \tag{18}$$

Proof. (i) For $f : \mathbb{R}_+^m \to \mathbb{R}_+$, a recoverable function through the parameterized grid G, we start by proving that $W_{l_1 \ldots l_m} \geq W_{h_1 \ldots h_m}, \quad \forall l_1 = 1, \ldots, k_1, \ldots, \forall l_m = 1, \ldots, k_m, \forall h_1 \leq l_1, \ldots, \forall h_m \leq k_m$ implies f is a monotonic non-decreasing function. Let $\boldsymbol{x} \in C_{h_1 \ldots h_m}$ and $\boldsymbol{z} \in C_{l_1 \ldots l_m}$ with $\boldsymbol{x} \leq \boldsymbol{z}$. We need to prove that $f(\boldsymbol{x}) \leq f(\boldsymbol{z})$. By Lemma 1, we have that $h_j \leq l_j, \forall j = 1, \ldots, m$. Additionally, by hypothesis, f is a recoverable function through the parameterized grid G with parameters $\boldsymbol{w}$. In this way, $f(\boldsymbol{x}) = \sum_{\substack{s_1 = 1, \ldots, h_1 \\ \vdots \\ s_m = 1, \ldots, h_m}} w_{s_1 \ldots s_m} = W_{h_1 \ldots h_m}$ for $\boldsymbol{x} \in C_{h_1 \ldots h_m}$ and $f(\boldsymbol{z}) = \sum_{\substack{s_1 = 1, \ldots, l_1 \\ \vdots \\ s_m = 1, \ldots, l_m}} w_{s_1 \ldots s_m} = W_{l_1 \ldots l_m}$ for $\boldsymbol{z} \in C_{l_1 \ldots l_m}$. Finally, $W_{l_1 \ldots l_m} \geq W_{h_1 \ldots h_m}, \quad \forall l_1 = 1, \ldots, k_1, \ldots, \forall l_m = 1, \ldots, k_m, \forall h_1 \leq l_1, \ldots, \forall h_m \leq k_m$ implies that $f(\boldsymbol{x}) \leq f(\boldsymbol{z})$. (ii) Let f be a monotonic non-decreasing function. We have to prove that $W_{l_1 \ldots l_m} \geq W_{h_1 \ldots h_m}, \quad \forall l_1 = 1, \ldots, k_1, \ldots, \forall l_m = 1, \ldots, k_m, \forall h_1 \leq l_1, \ldots, \forall h_m \leq k_m$. Let consider, without loss of generality, $l_1 \ldots l_m$ and $h_1 \ldots h_m$ with $h_j \leq l_j, \forall j = 1, \ldots, m$. Let prove that $a_{h_1 \ldots h_m} \leq a_{l_1 \ldots l_m}$. Suppose that there exists $j' \in \{1, \ldots, m\}$

such that $a^{(j')}_{h_1\ldots h_m} > a^{(j')}_{l_1\ldots l_m}$. By the definition of cell in a grid, we have that $t^{(j')}_{h_{j'}} = a^{(j')}_{h_1\ldots h_m} \leq x^{(j')} \quad \forall \boldsymbol{x} \in C_{h_1\ldots h_m}$ and $t^{(j')}_{l_{j'}} = a^{(j')}_{l_1\ldots l_m} \leq z^{(j')} \quad \forall \boldsymbol{z} \in C_{l_1\ldots l_m}$. Therefore, $t^{(j')}_{h_{j'}} > t^{(j')}_{l_{j'}}$. By Definition 1, we now have that the last inequality is only possible if $h_{j'} > l_{j'}$, which is a contradiction with the fact that $h_j \leq l_j, \forall j = 1,\ldots,m$. Then, $a_{h_1\ldots h_m} \leq a_{l_1\ldots l_m}$ with $h_j \leq l_j, \forall j = 1,\ldots,m$. Moreover, $a_{h_1\ldots h_m} \in C_{h_1\ldots h_m}$, $a_{l_1\ldots l_m} \in C_{l_1\ldots l_m}$ and f is a recoverable function through the parameterized grid G with parameters $\boldsymbol{w}$, which means that $f\left(a_{h_1\ldots h_m}\right) = W_{h_1\ldots h_m}$ and $f\left(a_{l_1\ldots l_m}\right) = W_{l_1\ldots l_m}$. Finally, since f is a monotonic non-decreasing function and $a_{h_1\ldots h_m} \leq a_{l_1\ldots l_m}$, we can conclude that $f\left(a_{h_1\ldots h_m}\right) \leq f\left(a_{l_1\ldots l_m}\right)$, that is, $W_{h_1\ldots h_m} \leq W_{l_1\ldots l_m}$. ■

It is possible to state a similar result but using a smaller number of constraints than in (18). This statement is formally established in Proposition 2.

Proposition 2. The system of inequalities $W_{h_1\ldots h_m} \leq W_{l_1\ldots l_m}$, $\forall h_j \leq l_j, \forall j = 1,...,m,$ with $h_j, l_j = 1,...,k_j$ is equivalent to the system of inequalities $W_{l_1 l_2 \ldots s_j \ldots l_m} \leq W_{l_1 l_2 \ldots l_m}$, $\forall s_j = l_j - 1, \forall l_j = 1,...,k_j, \forall j = 1,\ldots,m$.

Proof. (i) $W_{h_1\ldots h_m} \leq W_{l_1\ldots l_m}$, $\forall h_j \leq l_j, \forall j = 1,...,m,$ with $h_j, l_j = 1,...,k_j$, trivially implies that $W_{l_1 l_2 \ldots s_j \ldots l_m} \leq W_{l_1 l_2 \ldots l_m}$ $\forall s_j = l_j - 1, \forall l_j \leq k_j, \forall j = 1,\ldots,m$. (ii) Let us assume that $W_{l_1 l_2 \ldots s_j \ldots l_m} \leq W_{l_1 l_2 \ldots l_m}$, $\forall s_j = l_j - 1, \forall l_j = 1,...,k_j, \forall j = 1,\ldots,m$. Let then $h_j, l_j \in \{1,\ldots,k_j\}$, $\forall j = 1,\ldots,m$ with $h_j \leq l_j$, $\forall j = 1,\ldots,m$. Then, we have that $W_{l_1 l_2 \ldots l_m} \geq W_{l_1 - 1 l_2 \ldots l_m} \geq W_{l_1 - 2 l_2 \ldots l_m} \geq \ldots \geq W_{h_1 l_2 \ldots l_m} \geq W_{h_1 l_2 - 1 \ldots l_m} \geq W_{h_1 l_2 - 2 \ldots l_m} \geq \ldots \geq W_{h_1 h_2 \ldots l_m} \geq W_{h_1 h_2 \ldots l_{m-1}} \geq W_{h_1 h_2 \ldots l_{m-2}} \geq \ldots \geq W_{h_1 \ldots h_m}$. ■

Once we have identified the conditions that characterize monotonicity for step functions defined on a parametrized grid, we are able to introduce the final optimization model corresponding to the new technique for estimating production functions, the so-called Support Vector Frontiers (SVF):

$$
\begin{array}{lll}
\underset{\boldsymbol{w},\xi_i}{Min} & \|\boldsymbol{w}\|_1 + C\sum_{i=1}^{n}\xi_i & (19.0) \\
s.t. & y_i - \boldsymbol{w}\phi^G_{SVF}(\boldsymbol{x}_i) \leq 0, \quad i = 1,\ldots,n & (19.1) \\
 & \boldsymbol{w}\phi^G_{SVF}(\boldsymbol{x}_i) - y_i \leq \varepsilon + \xi_i, \quad i = 1,\ldots,n & (19.2) \\
 & W_{l_1 l_2 \ldots s_j \ldots l_m} \leq W_{l_1 l_2 \ldots l_m}, \quad \forall l_1,...,l_m, \forall s_j = l_j - 1, \forall j = 1,\ldots,m & (19.3) \\
 & \xi_i \geq 0, \quad i = 1,\ldots,n & (19.4)
\end{array}
\qquad (19)
$$

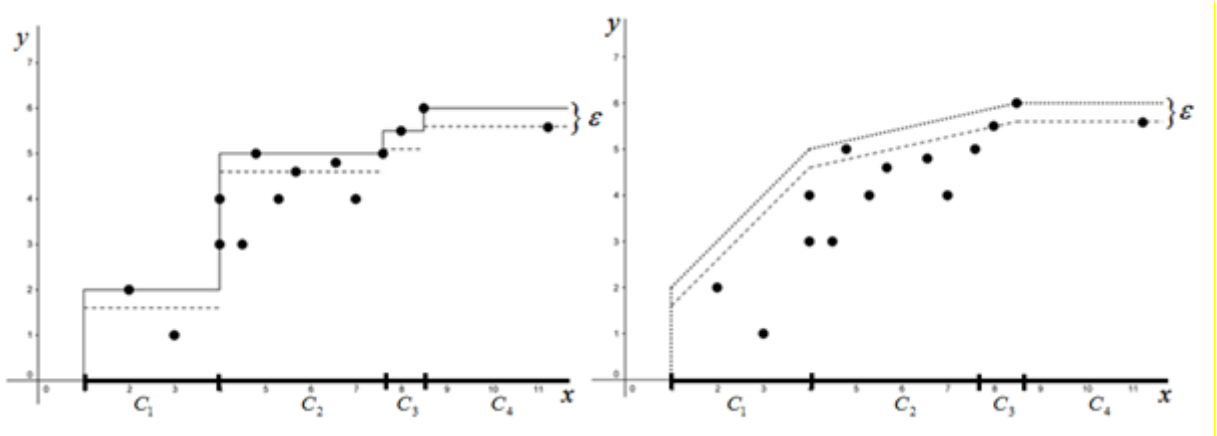
The constraints (19.1)-(19.4) correspond to the restrictions of model (11) with the new set of constraints (19.3) for ensuring monotonicity. Regarding the objective function, (19.0) coincides with (11.0) but, in the final model (19), a specific norm is chosen. Although the Euclidean norm is the most usual metric when standard SVM is utilized, other possibilities exist in the literature (see, for example, Blanco et al., 2020). In particular, in this paper, we consider the norm ℓ_1 because one of our objectives is comparing the new approach with standard FDH

and DEA, which are based on linear programming. Note that, by assuming the norm ℓ_1, and given that $\boldsymbol{w}\phi^G_{SVF}(\boldsymbol{x}_i) = \sum_{\substack{s_1=1,\dots,l_1 \\ \vdots \\ s_m=1,\dots,l_m}} w_{s_1\dots s_m}$ with $\boldsymbol{x}_i \in C_{l_1\dots l_m}$, model (19) is a linear program.

Given an optimal solution of (19), $(\boldsymbol{w}^*, \xi_1^*, \dots, \xi_n^*)$, the predictor of the production frontier linked to SVF is defined as $f_{SVF}(\boldsymbol{x}) = \boldsymbol{w}^* \phi^G_{SVF}(\boldsymbol{x})$, which is equal to $\sum_{\substack{s_1=1,\dots,l_1 \\ \vdots \\ s_m=1,\dots,l_m}} w^*_{s_1\dots s_m}$ if $\boldsymbol{x} \in C_{l_1\dots l_m}$. By the previous discussion, $f_{SVF}(\boldsymbol{x})$ is a step monotonic non-decreasing function that (upper) envelops the learning data. In order to additionally achieve a concave function, which in production theory is translated to convex technologies, it is enough to convexificate the predictor $f_{SVF}(\boldsymbol{x})$. A similar strategy is followed to get a convex production possibility set in DEA from a step production frontier estimated by FDH. To do that, in the case of SVF, we must identify a certain set of extreme points. In particular, each cell $C_{l_1\dots l_m}$ in the grid generates one of these points: one with input vector $\boldsymbol{a}_{l_1\dots l_m}$, the lower extreme knot-point of cell $C_{l_1\dots l_m}$, and output equals the prediction of the response variable for $\boldsymbol{a}_{l_1\dots l_m}$, i.e., $f_{SVF}(\boldsymbol{a}_{l_1\dots l_m})$. Given the set of 'virtual' input-output points $\left\{\left(\boldsymbol{a}_{l_1\dots l_m}, f_{SVF}(\boldsymbol{a}_{l_1\dots l_m})\right)\right\}_{\substack{l_1=1,\dots,k_1 \\ \vdots \\ l_m=1,\dots,k_m}}$, a convex production possibility set can be determined by applying a DEA-type estimation by analogy with expression (4):

$$\hat{T}_{CSVF} = \left\{(\boldsymbol{x},y) \in R^{m+1}_+ : y \le \sum_{\substack{l_1=1,\dots,k_1 \\ \vdots \\ l_m=1,\dots,k_m}} \lambda_{l_1\dots l_m} f_{SVF}(\boldsymbol{a}_{l_1\dots l_m}), x^{(j)} \ge \sum_{\substack{l_1=1,\dots,k_1 \\ \vdots \\ l_m=1,\dots,k_m}} \lambda_{l_1\dots l_m} a^{(j)}_{l_1\dots l_m}, \forall j, \sum_{\substack{l_1=1,\dots,k_1 \\ \vdots \\ l_m=1,\dots,k_m}} \lambda_{l_1\dots l_m} = 1, \lambda_{l_1\dots l_m} \ge 0, \forall l_1, \dots, \forall l_m \right\}, \quad (20)$$

where the acronym CSVF denotes 'Convexificated Support Vector Frontiers'.



(a) *(b)*

Figure 6. Example of an estimation of SVF (a) and CSVF (b)

Figure 6a shows a graphical example of an estimation of a production function from a learning sample with an input and an output. On the x-axis, we show the cells of the considered grid. The solid line shows the step function linked to the predictor directly derived from the solution of model (19). It is a monotonic non-decreasing function that envelops the data. In contrast to FDH, SVF does not consider the principle of minimal extrapolation and, consequently, the SVF predictor can be located strictly above the FDH predictor with certain contact points. On the one hand, the hyperparameter C in the objective function of model (19), which will be fitted by a cross-validation process as in the standard SVM, is the value that helps to locate the frontier more or less above the data. Greater values for C give more importance to the observations and are associated with an estimated frontier located closer to the data. On the other hand, the hyperparameter ε, which will be also fitted by cross-validation, allows to consider a certain degree of robustness in the analysis. The lower 'correcting' surface, defined as $f_{SVF}(\boldsymbol{x})-\varepsilon$, is also illustrated in the figure by the dashed line. Observations between the surfaces $f_{SVF}(\boldsymbol{x})-\varepsilon$ and $f_{SVF}(\boldsymbol{x})$ have an empirical error of zero ($\xi_i = 0$). Additionally, in Figure 6b, we also show the convexification of the SVF predictor. It is the dashed dotted line, which is derived from the points associated with the lower limit of each cell in the grid. Some of these lower limits are observations and some are virtual points. The dashed dotted line graphically describes an estimated production function that is concave, which implies that the area below this frontier defines a convex set. Furthermore, Figure 6b also shows the lower correcting surface for the convex setting, which is determined in a natural way by convexification from $f_{SVF}(\boldsymbol{x})-\varepsilon$ in Figure 6a.

Figure 6 and the notion of margin ε in standard SVM inspire a new and more robust notion of technical efficiency in production theory. We are referring to Definition 8, where, in particular, the concept of ε-insensitive technical efficiency is introduced.

Definition 8. The DMU $(\boldsymbol{x}_i, y_i)$ is ε-insensitive technically efficient if and only if $f_{SVF}(\boldsymbol{x}_i)-\varepsilon \leq y_i \leq f_{SVF}(\boldsymbol{x}_i)$.

Note that, of course, the set of ε-insensitive technically efficient units include the traditional technically efficient DMUs, i.e., those located onto the (estimated) production frontier f_{SVF}. In rough terms, the model does not discriminate, regarding the degree of technical efficiency, among the units located between the predictor and the lower correcting surface. For the SVF approach, all are equally technically efficient.

Finally, to finish this subsection, it is worth mentioning that model (19) needs the prior determination of certain parameters. Both C and ε were previously mentioned in the text. However, these are not the only parameters to be determined. As in the standard SVM, the transformation function (or, instead, the associated kernel) depends on another set of parameters. In our context, the SVF transformation function ϕ_{SVF}^{G} is linked to the determination of a set of knots for each input dimension considered in the problem. In this paper, we apply a simplified search method to determine these points since their identification may turn out to be a computationally hard task if their number and position are let free. Instead, the key parameter to be determined by cross-validation will be d, the number of cells to be defined for each input dimension. In an empirical context, we observe a maximum and minimum value in each input variable, from which it is possible to calculate the range and split it into d cells with the same width. This process will be illustrated in Section 4.

3.3. Free Disposal Hull and Data Envelopment Analysis as Support Vector Frontiers

Free Disposal Hull (FDH) and Data Envelopment Analysis (DEA) are well-known non-parametric approaches to technical efficiency analysis. In this subsection, we show that FDH and DEA may be alternatively interpreted as Support Vector Regression subject to shape constraints (monotonicity and concavity) that generates upper enveloping surfaces of the learning sample. This reinterpretation reveals the nature of FDH and DEA as part of machine learning techniques. In particular, we prove that these two standard techniques are always feasible solutions of the Support Vector Frontiers optimization model, being optimal in the simplest case of working with only one input and one output.

Throughout the section, we will focus our attention on FDH, achieving similar results with respect to DEA by direct convexification. We start by showing that the step function yielded by applying the FDH technique is always feasible in the SVF model (19), when a specific grid based on the observations is considered. We are referring to the 'empirical' grid, which uses the observations as knots for each input dimension and will be formally introduced through Definition 9. Before, let us introduce new notation. Let $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n}$ be a learning sample with $\boldsymbol{x}_i \in \mathbb{R}_+^m$ and $y_i \in \mathbb{R}_+$, $\forall i = 1,\ldots,n$. Then, the set of elements $\{\tilde{x}_{l_j}^{(j)}\}_{l_j=1}^{n^{(j)}}$, for each $j = 1,\ldots,m$, is the set of ordered values of the observations once repetitions have been removed in each dimension. Consequently, $n^{(j)} \leq n$ and $\tilde{x}_1^{(j)} < \ldots < \tilde{x}_{n^{(j)}}^{(j)}$.

Definition 9 [Empirical Grid]. The empirical grid G^E is the grid defined from the following sets of knots $T_j^E = \{\tilde{x}_{l_j}^{(j)}, l_j = 1,...,n^{(j)}+1\}$, $j = 1,\ldots,m$, with $\tilde{x}_{n^{(j)}+1}^{(j)} := \infty$, $\forall j = 1,\ldots,m$.

The cells in the empirical grid will be denoted by $C_{l_1\ldots l_m}^E$ and the extreme knot-points for each one as $\boldsymbol{a}_{l_1\ldots l_m}^E$ and $\boldsymbol{b}_{l_1\ldots l_m}^E$.

Another necessary definition is one that introduces the notion of set of dominated observations.

Definition 10 [Set of dominated observations]. Given an input vector $\boldsymbol{x} \in \mathbb{R}_+^m$, the set of dominated observations is defined as $X = \{i \in \{1,\ldots,n\} / \boldsymbol{x}_i \leq \boldsymbol{x}\}$.

Next, we are going to prove that the predictor generated by FDH is a step function defined on the empirical grid G^E. This result is key for proving that the FDH estimator is always feasible of the SVF model. In order to prove that, we first need to state some previous technical results.

Lemma 2. Let $i \in \{1,\ldots,n\}$ and $\boldsymbol{x}_i \leq \boldsymbol{a}_{l_1\ldots l_m}^E$. Then, $\boldsymbol{x}_i \leq \boldsymbol{x}$, $\forall \boldsymbol{x} \in C_{l_1\ldots l_m}^E$.

Proof. Let $x \in C_{l_1\ldots l_m}^E$. Then $\boldsymbol{a}_{l_1\ldots l_m}^E \leq \boldsymbol{x}$ because $C_{l_1\ldots l_m}^E = \{\boldsymbol{x} \in \mathbb{R}_+^m / \boldsymbol{a}_{l_1\ldots l_m}^E \leq \boldsymbol{x} < \boldsymbol{b}_{l_1\ldots l_m}^E\}$. By hypothesis, we have that $\boldsymbol{x}_i \leq \boldsymbol{a}_{l_1\ldots l_m}^E$. Therefore, $\boldsymbol{x}_i \leq \boldsymbol{a}_{l_1\ldots l_m}^E \leq \boldsymbol{x}$. ■

Corollary 1. Let $\boldsymbol{x} \in C^E_{l_1 \ldots l_m}$. Then $A^E_{l_1 \ldots l_m} := \left\{ i \in \{1,\ldots,n\} / \boldsymbol{x}_i \leq \boldsymbol{a}^E_{l_1 \ldots l_m} \right\} \subset X$.

Proof. The result is a direct consequence of Lemma 2. ■

Lemma 3. Let $i \in \{1,\ldots,n\}$ and $\boldsymbol{x} \in C^E_{l_1 \ldots l_m}$. If $\boldsymbol{x}_i \leq \boldsymbol{x}$, then $\boldsymbol{x}_i \leq \boldsymbol{a}^E_{l_1 \ldots l_m}$.

Proof. Let $x \in C^E_{l_1 \ldots l_m}$ and $\boldsymbol{x}_i \leq \boldsymbol{x}$ for an $i \in \{1,\ldots,n\}$. By reduction to the absurd, let us assume that it is not true that $\boldsymbol{x}_i \leq \boldsymbol{a}^E_{l_1 \ldots l_m}$. Then, there exists $j' \in \{1,\ldots,m\}$ such that $x_i^{(j')} > a^{E(j')}_{l_1 \ldots l_m}$. By the definition of $C^E_{l_1 \ldots l_m} = \left\{ \boldsymbol{x} \in \mathbb{R}^m_+ / \boldsymbol{a}^E_{l_1 \ldots l_m} \leq \boldsymbol{x} < \boldsymbol{b}^E_{l_1 \ldots l_m} \right\}$, we have that $a^{E(j')}_{l_1 \ldots l_m} < x_i^{(j')} \leq x^{(j')} < b^{E(j')}_{l_1 \ldots l_m}$. Therefore, there exist $\tilde{x}^{(j')}_{l_{j'}}$ and $\tilde{x}^{(j')}_{l_{j'}+1}$, successive elements in the ordered set of values $\left\{ \tilde{\boldsymbol{x}}^{(j')}_{l_{j'}} \right\}_{l_{j'}=1}^{n^{(j')}}$ such that $\tilde{x}^{(j')}_{l_{j'}} < x_i^{(j')} < \tilde{x}^{(j')}_{l_{j'}+1}$, with $\tilde{x}^{(j')}_{l_{j'}} = a^{E(j')}_{l_1 \ldots l_m}$ and $\tilde{x}^{(j')}_{l_{j'}+1} = b^{E(j')}_{l_1 \ldots l_m}$. However, this is a contradiction with the fact that $x_i^{(j')} \in \left\{ \tilde{\boldsymbol{x}}^{(j')}_{l_{j'}} \right\}_{l_{j'}=1}^{n^{(j')}}$. Consequently, $\boldsymbol{x}_i \leq \boldsymbol{a}^E_{l_1 \ldots l_m}$. ■

Corollary 2. Let $\boldsymbol{x} \in C^E_{l_1 \ldots l_m}$. Then $X \subset A^E_{l_1 \ldots l_m}$.

Proof. The result is a direct consequence of Lemma 3. ■

Proposition 3. Let $\boldsymbol{x} \in C^E_{l_1 \ldots l_m}$. Then $X = A^E_{l_1 \ldots l_m}$.

Proof. This result is a consequence of Corollaries 1 and 2. ■

The next theorem proves that the predictor associated with the FDH technique is a step function defined on the empirical grid. Note that it is well-known in the literature that the FDH methodology always generates step functions. However, we also need to prove that it is a step function defined on a grid (following Definition 6). In this case, we identify such grid with the empirical one.

Theorem 2. The predictor yielded from the FDH technique is a step function defined on the grid G^E. In particular, $f_{FDH}(\boldsymbol{x}) = f_{FDH}\left(\boldsymbol{a}^E_{l_1 \ldots l_m} \right),\ \forall \boldsymbol{x} \in C^E_{l_1 \ldots l_m}$.

Proof. If $\boldsymbol{x} \in C^E_{l_1 \ldots l_m}$, then $f_{FDH}(\boldsymbol{x}) = \max_{i:\boldsymbol{x} \geq \boldsymbol{x}_i} \{y_i\} = \max_{i \in X} \{y_i\} = \max_{i \in A^E_{l_1 \ldots l_m}} \{y_i\} = \max_{i / \boldsymbol{x}_i \leq \boldsymbol{a}^E_{l_1 \ldots l_m}} \{y_i\} = f_{FDH}\left(\boldsymbol{a}^E_{l_1 \ldots l_m} \right)$, where the third equality is true by Proposition 3, the last equality by the monotonicity of the FDH function and the other equalities by definition. ■

By Theorem 2 and Proposition 1, we know that there is a vector $\boldsymbol{w}^{FDH}$ such that the predictor associated with the FDH technique can be equivalently expressed as $f_{FDH}(\boldsymbol{x}) = \boldsymbol{w}^{FDH} \phi^{G^E}_{SVF}(\boldsymbol{x})$. Moreover, by Definition 7, f_{FDH} is a recoverable function through the parameterized grid G^E with parameters $\boldsymbol{w}^{FDH}$. Additionally, if we define the empirical errors as

$$\xi_i^{FDH} := \begin{cases} \left(f_{FDH}(\boldsymbol{x}) - \varepsilon \right) - y_i, & \text{if } \left(f_{FDH}(\boldsymbol{x}) - \varepsilon \right) - y_i \geq 0 \\ 0, & \text{if } \left(f_{FDH}(\boldsymbol{x}) - \varepsilon \right) - y_i < 0 \end{cases}, \forall i = 1,\ldots,n, \qquad (21)$$

then we can state that FDH always generates a feasible solution of model (19). Expression (21) may be equivalently written in a compact way as $\xi_i^{FDH} = \max\left\{\left(f_{FDH}(\boldsymbol{x}) - \varepsilon\right) - y_i, 0\right\}$.

Theorem 3. $\left(\boldsymbol{w}^{FDH}, \xi_1^{FDH}, ..., \xi_n^{FDH}\right)$ is a feasible solution of model (19).

Proof. Conditions (19.1) are trivially satisfied because $\boldsymbol{w}^{FDH}\phi_{SVF}^{G^E}(\boldsymbol{x}) = f_{FDH}(\boldsymbol{x})$ is an upper enveloping function of the data. Constraints (19.2) and (19.4) hold thanks to the definition of the empirical errors ξ_i^{FDH}, $\forall i = 1, ..., n$. Additionally, f_{FDH} is a recoverable function through the parameterized grid G^E with parameters $\boldsymbol{w}^{FDH}$. Then, given that it is well-known that f_{FDH} is a monotonic non-decreasing function, invoking Theorem 1 and Proposition 2, conditions (19.3) are satisfied by the vector of parameters $\boldsymbol{w}^{FDH}$. ■

Accordingly, the step function generated by the standard FDH technique can be always recovered from the new approach, the so-called Support Vector Frontiers technique. The classical FDH estimated production function is always considered by model (19), given any parameters C and ε, as a possible SVF predictor. Another thing is that the FDH step function was optimal. This is not always true. Nevertheless, we are able to prove that, for any parameters C and ε, in the simplest framework, i.e. when the number of inputs equals one, the FDH predictor is always optimal of model (19) when the empirical grid is considered. It will be Theorem 4. Before proving it, we will establish a useful lemma.

Lemma 4. Let $m = 1$ and let $\left(\boldsymbol{w}', \xi_1', ..., \xi_n'\right)$ be a feasible solution of model (19) when the empirical grid is considered. Then, $\boldsymbol{w}'\phi_{SVF}^{G^E}\left(\max_{1 \le i \le n}\{x_i\}\right) \ge \max\{y_1, ..., y_n\}$.

Proof. Let $i' = 1, ..., n$ be such that $y_{i'} = \max\{y_1, ..., y_n\}$. Then, by (19.1), $\boldsymbol{w}'\phi_{SVF}^{G^E}(x_{i'}) \ge y_{i'}$. By (19.3), $h(x) := \boldsymbol{w}'\phi_{SVF}^{G^E}(x)$ is a monotonic non-decreasing function. Therefore, $\max_{1 \le i \le n}\{x_i\} \ge x_{i'}$ implies $\boldsymbol{w}'\phi_{SVF}^{G^E}\left(\max_{1 \le i \le n}\{x_i\}\right) \ge \boldsymbol{w}'\phi_{SVF}^{G^E}(x_{i'}) \ge y_{i'} = \max\{y_1, ..., y_n\}$. ■

Theorem 4. Let $m = 1$. Then, $\left(\boldsymbol{w}^{FDH}, \xi_1^{FDH}, ..., \xi_n^{FDH}\right) \ge 0$ is an optimal solution of model (19).

Proof. Let $\left(\boldsymbol{w}', \xi_1', ..., \xi_n'\right)$ be any feasible solution of model (19). By (19.3), $w_1' \le w_1' + w_2'$, $w_1' + w_2' \le w_1' + w_2' + w_3'$, …, $w_1' + w_2' + w_3' + ... + w_{n^{(1)}-1}' \le w_1' + w_2' + w_3' + ... + w_{n^{(1)}}'$. Therefore, $w_{l_1}' \ge 0$, $l_1 = 2, ..., n^{(1)}$. Let $i', i'' = 1, ..., n$ be such that $x_{i'} = \min_{1 \le i \le n}\{x_i\}$ and $x_{i''} = \max_{1 \le i \le n}\{x_i\}$. By (19.1), $\boldsymbol{w}'\phi_{SVF}^{G^E}(x_{i'}) \ge y_{i'}$, which is equivalent to $w_1' \ge y_{i'} \ge 0$. Regarding the objective function in (19), $\|\boldsymbol{w}'\|_1 + C\sum_{i=1}^{n}\xi_i' = w_1' + ... + w_{n^{(1)}}' + C\sum_{i=1}^{n}\xi_i' = \boldsymbol{w}'\phi_{SVF}^{G^E}(x_{i''}) + C\sum_{i=1}^{n}\xi_i'$. Additionally, it is well-known that $f_{FDH}(x_{i''}) = \max\{y_1, ..., y_n\}$. Then, $\boldsymbol{w}^{FDH}\phi_{SVF}^{G^E}(x_{i''}) = f_{FDH}(x_{i''}) = \max\{y_1, ..., y_n\}$, which is, by Lemma 4, the minimum value that $\boldsymbol{w}'\phi_{SVF}^{G^E}(x_{i''})$ can take for any feasible solution $\left(\boldsymbol{w}', \xi_1', ..., \xi_n'\right)$ of (19). Moreover,

given any feasible solution $\left(\boldsymbol{w}', \xi_1', ..., \xi_n'\right)$, $\xi_i' = \max\left\{\left(\boldsymbol{w}'\phi_{SVF}^{G^E}\left(x_i\right) - \varepsilon\right) - y_i, 0\right\}$ because we are minimizing (19.0), which includes the term $C\sum_{i=1}^{n}\xi_i'$, and also thanks to constraints (19.2). Finally, note that, by the principle of minimal extrapolation, $f_{FDH}\left(x_i\right) = \boldsymbol{w}^{FDH}\phi_{SVF}^{G^E}\left(x_i\right) \leq \boldsymbol{w}'\phi_{SVF}^{G^E}\left(x_i\right)$, $\forall i = 1, ..., n$, which is equivalent to $\left(\boldsymbol{w}^{FDH}\phi_{SVF}^{G^E}\left(x_i\right) - \varepsilon\right) - y_i \leq \left(\boldsymbol{w}'\phi_{SVF}^{G^E}\left(x_i\right) - \varepsilon\right) - y_i$, $\forall i = 1, ..., n$. These last inequalities imply that $\xi_i^{FDH} \leq \xi_i'$, $i = 1, ..., n$, for any feasible solution $\left(\boldsymbol{w}', \xi_1', ..., \xi_n'\right)$. Consequently, $\boldsymbol{w}^{FDH}\phi_{SVF}^{G^E}\left(x_{i''}\right) + C\sum_{i=1}^{n}\xi_i^{FDH} \leq \boldsymbol{w}'\phi_{SVF}^{G^E}\left(x_{i''}\right) + C\sum_{i=1}^{n}\xi_i'$ for any feasible solution $\left(\boldsymbol{w}', \xi_1', ..., \xi_n'\right)$, which means that $\left(\boldsymbol{w}^{FDH}, \xi_1^{FDH}, ..., \xi_n^{FDH}\right)$ is an optimal solution of model (19). ■

However, Theorem 4 cannot be extended to the general framework, i.e., when the number of inputs is arbitrary. To show that, it is enough to provide a counterexample of the property for two inputs. Let us consider the following learning sample for three units (A, B and C): $\left(x_1^A, x_2^A, y^A\right) = (1, 4, 2)$, $\left(x_1^B, x_2^B, y^B\right) = (2, 2, 1)$ and $\left(x_1^C, x_2^C, y^C\right) = (3, 1, 3)$. Considering the empirical grid, the solution associated with the FDH technique is as follows:

$$\begin{array}{llll} w_{14}^{FDH} = 0 & w_{24}^{FDH} = 2 & w_{34}^{FDH} = -1 & w_{44}^{FDH} = -1 \\ w_{13}^{FDH} = 0 & w_{23}^{FDH} = 0 & w_{33}^{FDH} = 1 & w_{43}^{FDH} = -1 \\ w_{12}^{FDH} = 0 & w_{22}^{FDH} = 0 & w_{32}^{FDH} = 0 & w_{42}^{FDH} = 3 \\ w_{11}^{FDH} = 0 & w_{21}^{FDH} = 0 & w_{31}^{FDH} = 0 & w_{41}^{FDH} = 0 \end{array} \quad (22).$$

with $\xi_1^{FDH} = \xi_2^{FDH} = \xi_3^{FDH} = 0$. For $C = 1$ and $\varepsilon = 0$, this solution generates a value of the objective function (19.0) equal to 9, while the optimal value of model (19) for this simple numerical example is really 4. Therefore, the FDH predictor is always a feasible step function associated with the SVF model but it does not have to be optimal, except for the one output-one input case.

By convexification, and taking into account that the standard DEA can be derived from the FDH step function by applying this same technique, all the previous results may be trivially extended to the estimation of convex production technologies. In this way, the traditional DEA estimation of the production function is always one of the possible solutions associated with the Convexified Support Vector Frontiers technique. Additionally, in the context of one input and one output, the DEA estimation is obtained from the optimal solution of the Support Vector Frontiers after applying the convexification of the SVF step function, regardless of the values of the hyperparameters C and ε.

4. Monte Carlo Simulations

This section describes simulation results that serve for the comparison of methods: FDH vs SVF and DEA vs CSVF. To do that, we present a comparison of all these methods in three alternative simulated environments. Their descriptions appear in Table 1.

Scenario	Inputs	Functional form
(A)	x_1	$y= 3+x_1^{0.5}-u$
(B)	x_1,x_2	$y= 3+x_1^{0.2}+x_2^{0.3}-u$
(C)	x_1,x_2,x_3	$y= 3+x_1^{0.05}+x_2^{0.15}+x_3^{0.3}-u$

Table 1. Description of the three scenarios

These three scenarios exemplify several Cobb-Douglas functions, widely known in the economic literature. Scenario A represents a single-input case and scenarios B and C represent multi-input cases. For all of them, we tested with data set sizes of 20, 30, 40, 50, 60, 70, 80, 90 and 100. The input data were randomly sampled from *Uni*[1,10], independently for each input and observation. Then, the efficient output level was calculated and a random inefficiency term $u \sim |N(0,0.4)|$ was subtracted to obtain the data used for the analysis. We ran 50 trials $(t = 1,...,50)$ for each combination of scenario and data set size to investigate the relative performance of the methods. Additionally, and due to the nature of SVF, as happens with SVM and SVR, the best model must be selected through the determination of the best combination of hyperparameters. This was carried out through a cross-validation process based on five folds. In our framework, the hyperparameters are C, ε and d, the number of cells with the same width to be defined for each input dimension. In practice, it is necessary to state a finite number of possible combinations of these hyperparameters, since the computational cost of testing the infinite combinations of hyperparameters would be unmanageable. Arbitrarily, we fixed the following values for each hyperparameter: $C \in \{0.1,\ 0.5,\ 1,\ 2,\ 5\}$, $\varepsilon \in \{0,\ 0.001,\ 0.01,\ 0.1,\ 0.2\}$ and $d \in \{1,2,3,4,5,6,7,8,9,10\}$. These values generate a total quantity of 250 different combinations of hyperparameters for the SVF problem. Additionally, performance of each method is evaluated by two standard criteria: the mean squared error (MSE) and the bias. The MSE statistic is defined as $\sum_{t=1}^{50}\sum_{i=1}^{n}\left(f\left(\boldsymbol{x}_i^t\right)-\hat{f}\left(\boldsymbol{x}_i^t\right)\right)^2 \Big/ 50n$, whereas the bias is calculated as $\sum_{t=1}^{50}\sum_{i=1}^{n}\left|f\left(\boldsymbol{x}_i^t\right)-\hat{f}\left(\boldsymbol{x}_i^t\right)\right| \Big/ 50n$.

Table 2 reports the mean and standard deviation (in brackets) of the best hyperparameters in our simulations. All of them were estimated by cross-validation except for scenario C, where, due to computational issues, we fixed the hyperparameter in $d = 5$. Regarding the results, the hyperparameter C was indifferent for the scenario with an input (A), because the cross-validated prediction error was always the same. However, the value of the hyperparameter C was relevant for scenarios with more inputs, as B and C. Something similar happened with the value of the hyperparameter ε in scenario A with small sample sizes (20 and 30 observations). Regarding the hyperparameter d, it determines the size of the grid G (the number of cells), and, consequently, the number of decision variables and constraints in model (19). The observed trend shows that the greater the number of observations, the larger the value of the hyperparameter d.

Scenario	Number of obs.	Hyperparameter						
		C	ε			d		
		MEAN (STD)	MEAN (STD)	MIN	MAX	MEAN (STD)	MIN	MAX
A	20	indifferent	indifferent	0	0	3.55(1.001)	1.5	5
	30			0	0	5.79(1.438)	2.25	7.5
	40		0.037(0.077)	0	0.2	7.92(1.736)	3	10
	50		0.079(0.095)	0	0.2	10.45(2.001)	6.25	12.5
	60		0.088(0.089)	0	0.2	12.27(2.302)	6	15
	70		0.073(0.088)	0	0.2	15.225(2.271)	8.75	17.5
	80		0.061(0.087)	0	0.2	18.08(2.175)	12	20
	90		0.033(0.068)	0	0.2	19.215(3.057)	11.25	22.5
	100		0.016(0.047)	0	0.2	22.45(3.051)	15	25
B	20	1.804(1.924)	0.047(0.073)	0	0.2	2.88(1.335)	1	5
	30	3.064(1.941)	0.025(0.045)	0	0.2	4.23(1.939)	1.5	7.5
	40	3.332(1.811)	0.009(0.024)	0	0.1	6.32(2.308)	2	10
	50	3.66(1.695)	0.022(0.049)	0	0.2	6.15(3.132)	1.25	12.5
	60	3.15(1.785)	0.012(0.033)	0	0.2	9.51(4.151)	3	15
	70	2.98(1.635)	0.006(0.02)	0	0.1	12.18(4.315)	3.5	17.5
	80	3.96(1.551)	0.002(0.004)	0	0.01	12.44(5.804)	4	20
	90	3.38(1.589)	0.011(0.027)	0	0.1	15.705(5.521)	4.5	22.5
	100	3.46(1.717)	0.011(0.034)	0	0.2	15.8(6.418)	2.5	25
C	20	1.9(1.897)	0.054(0.08)	0	0.2	5(0)		
	30	1.964(1.755)	0.019(0.041)	0	0.2			
	40	1.978(1.799)	0.023(0.044)	0	0.2			
	50	2.364(1.841)	0.028(0.058)	0	0.2			
	60	2.216(1.836)	0.011(0.027)	0	0.1			
	70	1.996(1.697)	0.014(0.032)	0	0.1			
	80	2.104(1.664)	0.012(0.027)	0	0.1			
	90	2.572(1.773)	0.023(0.053)	0	0.2			
	100	3.25(1.882)	0.025(0.045)	0	0.2			

Table 2: The best SVF hyperparameters

Table 3 describes the performance of the different methods studied: FDH, SVF, DEA and CSVF based on MSE. The first two columns indicate the scenario and the sample size. The next four columns show the mean and the standard deviation (in brackets) of the methods considered. The following two columns report the fraction of trials in which the SVF improves or equals the MSE of the FDH and the percentage of improvement of this method with respect to the other. The last two columns are like the previous ones but compare CSVF versus DEA. Regarding the results, all the methods were affected by the increase in dimensionality, from one to several inputs, since the MSE increases as the number of inputs increases. Likewise, the improvement of the SVF in relation to the FDH was quite substantial, giving better results in all the simulations. The improvements ranged from 14.3% to 34.4% on average. As for the comparison between CSVF and DEA, the percentages of enhancement were higher than in the previous comparison between SVF and FDH, with improvements ranging from 38.7% to 78.4% on average. One significant result is that the superiority of SVF and CSVF compared to traditional approaches is high even under contexts with a small number of observations.

Scenario	Number of obs.	Mean squared error							
		FDH	SVF	DEA	CSVF	Fraction of trials SVF<=FDH	Improvement (%) SVF vs FDH	Fraction of trials CSVF<=DEA	Improvement (%) CSVF vs DEA
A	20	0.06(0.026)	0.046(0.024)	0.021(0.017)	0.012(0.011)	1	23.126(18.476)	0.96	39.28(29.892)
	30	0.048(0.016)	0.038(0.014)	0.017(0.011)	0.01(0.008)	1	20.47(13.607)	0.98	39.604(27.292)
	40	0.036(0.01)	0.029(0.009)	0.01(0.007)	0.006(0.006)	1	18(9.163)	0.98	41.54(22.823)
	50	0.033(0.011)	0.026(0.01)	0.008(0.006)	0.004(0.005)	1	19.686(13.358)	1	49.347(24.152)
	60	0.027(0.008)	0.022(0.008)	0.007(0.006)	0.004(0.004)	1	18.776(10.464)	1	47.294(20.497)
	70	0.026(0.008)	0.021(0.007)	0.007(0.005)	0.003(0.004)	1	20.539(10.891)	1	52.575(21.751)
	80	0.022(0.007)	0.018(0.006)	0.005(0.004)	0.003(0.002)	1	17.649(10.122)	1	48.986(23.684)
	90	0.02(0.005)	0.017(0.005)	0.004(0.003)	0.002(0.002)	1	14.712(8.33)	1	40.99(17.318)
	100	0.018(0.005)	0.015(0.004)	0.004(0.003)	0.002(0.002)	1	15.18(9.632)	1	38.709(22.785)
B	20	0.093(0.031)	0.062(0.029)	0.048(0.023)	0.019(0.012)	1	32.906(21.515)	0.96	57.68(24.558)
	30	0.088(0.03)	0.065(0.027)	0.039(0.02)	0.014(0.01)	1	25.792(15.893)	0.98	62.702(21.254)
	40	0.074(0.021)	0.058(0.021)	0.031(0.013)	0.011(0.008)	1	21.86(16.536)	1	64.682(19.001)
	50	0.067(0.015)	0.052(0.016)	0.022(0.009)	0.009(0.009)	1	23.709(13.996)	0.96	55.665(45.693)
	60	0.064(0.015)	0.05(0.015)	0.021(0.01)	0.008(0.007)	1	21.534(13.903)	0.98	61.858(24.149)
	70	0.06(0.012)	0.05(0.013)	0.02(0.007)	0.008(0.007)	1	17.301(13.545)	0.98	58.134(21.477)
	80	0.055(0.01)	0.046(0.01)	0.017(0.007)	0.006(0.003)	1	15.941(11.554)	0.98	63.6(16.983)
	90	0.051(0.011)	0.044(0.011)	0.016(0.007)	0.007(0.005)	1	14.314(11.453)	0.98	60.297(19.467)
	100	0.047(0.008)	0.04(0.01)	0.014(0.005)	0.005(0.003)	1	15.497(12.1)	0.98	62.138(18.285)
C	20	0.125(0.046)	0.085(0.034)	0.078(0.036)	0.029(0.019)	1	30.401(20.041)	1	62.182(19.52)
	30	0.11(0.03)	0.075(0.028)	0.062(0.03)	0.019(0.013)	1	31.182(19.658)	0.98	66.747(18.605)
	40	0.106(0.032)	0.073(0.03)	0.054(0.02)	0.015(0.01)	1	31.681(18.429)	0.98	70.078(20.382)
	50	0.093(0.021)	0.063(0.018)	0.042(0.015)	0.011(0.006)	1	31.281(17.332)	1	73.877(15.085)
	60	0.094(0.018)	0.063(0.017)	0.042(0.014)	0.011(0.006)	1	32.396(14.453)	1	73.107(12.174)
	70	0.092(0.019)	0.061(0.018)	0.04(0.015)	0.01(0.007)	1	33.688(15.497)	1	74.851(14.008)
	80	0.082(0.017)	0.054(0.018)	0.034(0.011)	0.007(0.003)	1	34.423(15.374)	1	78.375(12.496)
	90	0.08(0.015)	0.053(0.015)	0.03(0.009)	0.009(0.013)	1	33.81(15.881)	0.96	66.782(53.812)
	100	0.079(0.013)	0.056(0.013)	0.03(0.009)	0.007(0.007)	1	29.867(11.365)	0.96	74.618(22.103)

Table 3: Relative performance of estimation methods based on the MSE performance criteria

Table 4 shows the results based on the bias instead of the MSE. The structure of Table 4 is like that of Table 3. Regarding the bias, SVF outperforms FDH for all the computational experiences carried out, with a reduction ranging from 8% to 19.9%. As for the convex technologies, CSVF works better than the traditional DEA, with improvements from 27.3% to 56.1%. For CSVF, the percentage of improvement increases as the number of inputs augments. As in the case of the MSE, the dominance of the new approaches compared to traditional ones is clear, even under frameworks based on few observations.

Finally, Figures 7 and 8 show a graphical example of the result of one of our simulations.

Scenario	Number of obs.	Bias							
		FDH_ABS	SVF_ABS	DEA_ABS	CSVF_ABS	Fraction of trials SVF_ABS<=FDH_ABS	Improvement (%) SVF_ABS vs FDH_ABS	Fraction of trials CSVF_ABS<=DEA_ABS	Improvement (%) CSVF_ABS vs DEA_ABS
A	20	0.198(0.05)	0.17(0.05)	0.104(0.042)	0.074(0.034)	1	14.211(10.371)	0.9	27.252(24.578)
	30	0.18(0.034)	0.156(0.034)	0.089(0.029)	0.06(0.023)	1	13.383(7.095)	0.96	30.993(18.712)
	40	0.155(0.024)	0.137(0.026)	0.066(0.022)	0.045(0.021)	1	12.288(6.549)	1	33.668(15.439)
	50	0.15(0.023)	0.132(0.024)	0.06(0.016)	0.04(0.016)	1	11.767(7.046)	0.98	35.347(16.288)
	60	0.134(0.018)	0.119(0.02)	0.053(0.016)	0.035(0.014)	1	11.152(5.373)	1	36.312(11.539)
	70	0.131(0.019)	0.116(0.02)	0.049(0.015)	0.032(0.015)	1	11.157(4.782)	1	37.062(13.023)
	80	0.123(0.02)	0.111(0.019)	0.046(0.015)	0.031(0.013)	1	9.347(3.65)	1	33.281(11.379)
	90	0.116(0.013)	0.106(0.014)	0.039(0.01)	0.027(0.009)	1	9.009(3.961)	1	31.853(10.586)
	100	0.11(0.015)	0.101(0.015)	0.038(0.011)	0.027(0.01)	1	8.722(3.565)	0.98	30.588(11.75)
B	20	0.249(0.045)	0.2(0.05)	0.17(0.042)	0.101(0.032)	1	19.867(13.551)	0.96	39.658(18.271)
	30	0.246(0.044)	0.212(0.046)	0.151(0.038)	0.082(0.031)	1	14.097(9.055)	0.98	44.811(18.545)
	40	0.223(0.035)	0.197(0.04)	0.132(0.029)	0.07(0.026)	1	12.283(8.81)	1	46.997(15.129)
	50	0.218(0.026)	0.189(0.032)	0.114(0.021)	0.063(0.027)	1	13.594(8.496)	0.96	43.884(22.736)
	60	0.207(0.025)	0.183(0.028)	0.104(0.02)	0.057(0.022)	1	11.829(7.956)	0.98	45.423(17.326)
	70	0.205(0.022)	0.187(0.023)	0.1(0.013)	0.061(0.021)	1	8.593(6.036)	0.98	39.713(18.1)
	80	0.193(0.017)	0.177(0.018)	0.091(0.013)	0.05(0.014)	1	8.19(5.649)	0.96	44.273(14.527)
	90	0.187(0.02)	0.173(0.022)	0.087(0.017)	0.05(0.017)	1	7.24(5.748)	0.98	42.471(15.682)
	100	0.182(0.017)	0.168(0.021)	0.081(0.013)	0.046(0.013)	1	8.035(6.362)	0.96	43.309(16.463)
C	20	0.283(0.049)	0.234(0.048)	0.214(0.045)	0.117(0.036)	1	16.94(12.771)	0.98	44.533(15.679)
	30	0.271(0.038)	0.224(0.045)	0.19(0.042)	0.098(0.032)	1	17.024(12.169)	0.98	47.769(15.344)
	40	0.264(0.039)	0.22(0.046)	0.173(0.034)	0.085(0.026)	1	17.113(10.631)	0.96	50.106(16.306)
	50	0.25(0.03)	0.206(0.03)	0.153(0.027)	0.072(0.023)	1	17.063(9.746)	1	52.36(15.557)
	60	0.25(0.027)	0.207(0.031)	0.15(0.024)	0.07(0.02)	1	17.391(8.809)	1	52.876(11.499)
	70	0.248(0.023)	0.204(0.031)	0.144(0.023)	0.069(0.022)	1	17.825(9.214)	0.96	52.017(16.303)
	80	0.235(0.026)	0.192(0.032)	0.134(0.021)	0.058(0.015)	1	18.561(8.696)	1	56.066(12.685)
	90	0.231(0.021)	0.189(0.027)	0.124(0.017)	0.064(0.035)	1	18.151(9.2)	0.94	47.215(31.566)
	100	0.232(0.019)	0.195(0.022)	0.124(0.016)	0.058(0.024)	1	15.712(6.472)	0.96	53.283(18.828)

Table 4: Relative performance of estimation methods based on the bias performance criteria

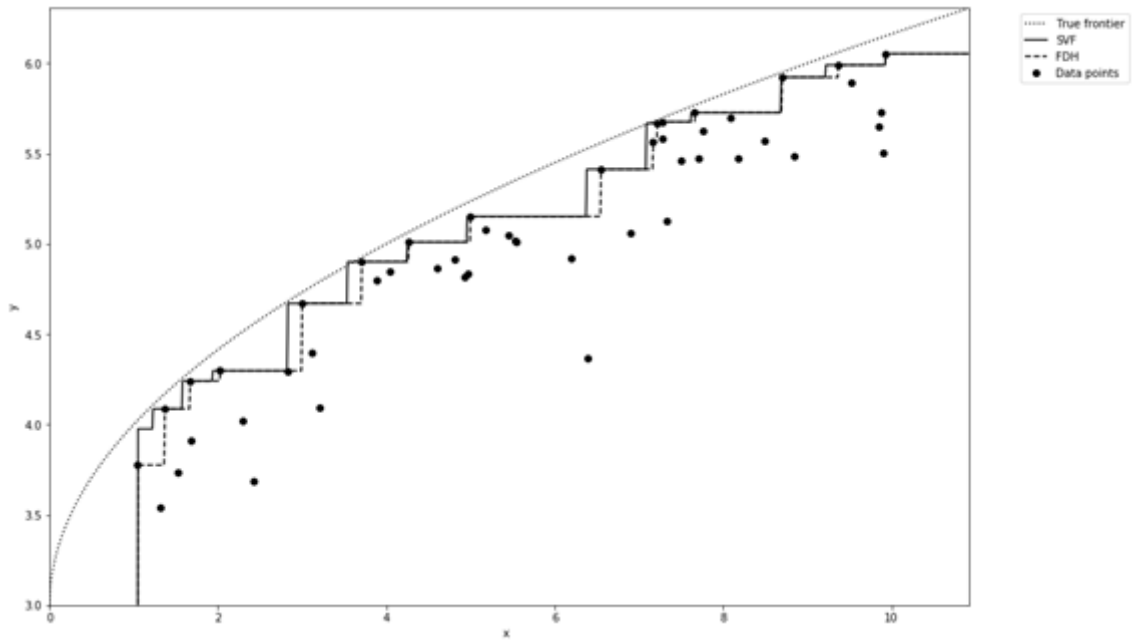


Figure 7: Graphical illustration of SVF vs FDH in a simulated dataset

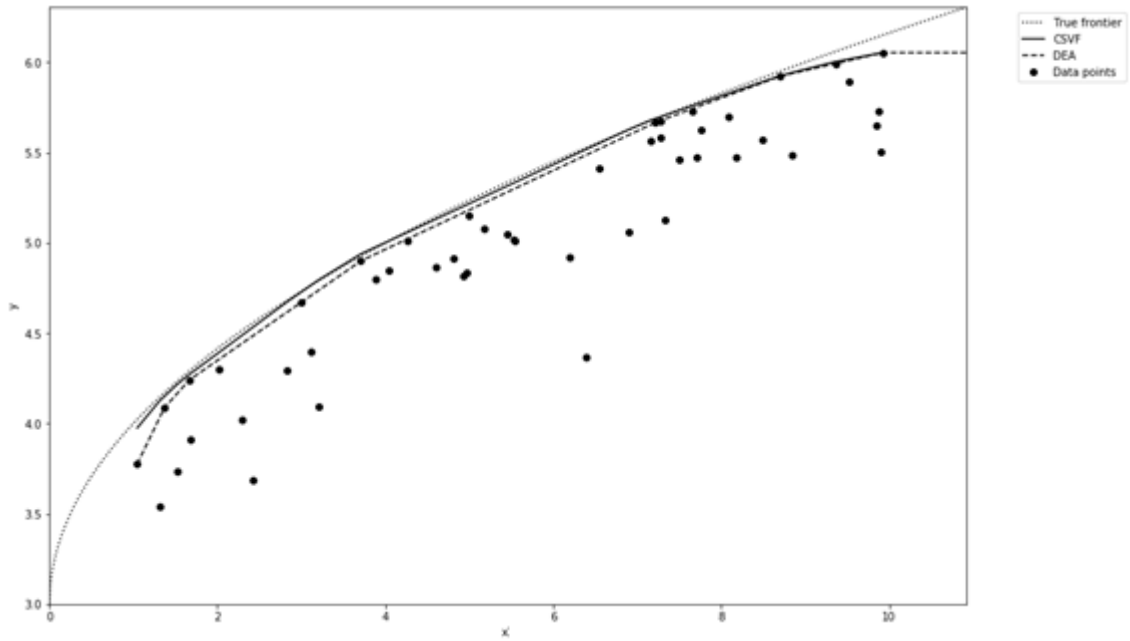


Figure 8: Graphical illustration of CSVF vs DEA in a simulated dataset

5. Conclusions and future work

In this paper, we built a new bridge between non-parametric frontier analysis and Support Vector Machines (SVM). So far, these two fields have been growing in parallel with few significant contact points. However, they present certain points in common and the new tendency in Economics, Engineering and Operations Research on big data and machine learning strongly encourages efficiency analysis researchers to join the data analytics field (see, for example, the recent papers by Khezrimotlagh et al., 2019 and Zhu, 2019). In our case, this has meant introducing a new approach to estimate production frontiers by adapting Support Vector Regression, generating the so-called Support Vector Frontiers (SVF). Specifically, standard Support Vector Regression is subjected to shape constraints and adapted in such a way that the predictor surface upper envelops all the

observations. Monotonicity is one of the key shape constraints. It is related to the classical axiom of free disposability, which any production function must meet in microeconomics. In a first stage and motivated by the Free Disposal Hull (FDH) technique, a specific transformation function of the input space was introduced that allowed determining monotonic non-decreasing step functions as predictor of the production functions. In a second stage, by convexification, we were able to yield concave predictors, which are directly linked to convex production possibility sets and Data Envelopment Analysis (DEA). This resulted in the introduction of the so-called Convexificated Support Vector Frontiers (CSVF). In contrast to standard SVM, SVF and CSVF are able to estimate production functions since they have been defined to capture maximum trends in the data instead of mean trends and to guarantee the satisfaction of monotonicity and concavity.

Both FDH and DEA are well-known non-parametric approaches in the literature for measuring technical efficiency. In this paper, we also showed that these two standard methods could be interpreted as feasible predictors of the introduced adaptation of the Support Vector Regression technique. Under the simplest context of producing one output by consuming only one input, FDH and DEA are also optimal solutions of Support Vector Frontiers. This reinterpretation reveals the nature of FDH and DEA as part of existing machine learning techniques.

Despite the similarities between the traditional non-parametric approaches for determining technical efficiency and Support Vector Frontiers, FDH and DEA suffer from a problem of overfitting, a problem inherited from the axiom of minimal extrapolation assumed in the classical literature (Farrell, 1957, Afriat, 1972, Banker et al., 1984). In contrast, the new technique does not assume this postulate, which endows SVF with more flexibility thus allowing to locate the estimated production frontier above the FDH and DEA predictors following a cross-validation process. Overall, the introduction of SVF in the literature opens up new ways for adapting and integrating machine learning techniques to the efficiency analysis world.

Furthermore, performance of the new approach was investigated in this paper via Monte Carlo simulation. Our results indicated that SVF and CSVF outperform FDH and DEA, respectively, with respect to several traditional error measures like the mean squared error (MSE) and the bias. Regarding the MSE, we observed that the determined improvements ranged from 14.3% to 78.4% in our simulations. As for the bias, the enhancement ranged from 8% to 56.1%.

The new technique presents both extra advantages and drawbacks in comparison with FDH and DEA. As for the extra returns, the adaptation of the standard SVR allowed translating certain notions of this machine learning technique to the efficiency measurement world. In particular, we introduced the robust concept of ε-insensitive technical efficiency, which incorporates the notion of margin to the definition of performance efficiency. Regarding the drawbacks, the new technique takes up more computing time in comparison with FDH and DEA. Although the optimization program that must be determined linked to SVF is a Linear Program, the high number of constraints associated with the satisfaction of monotonicity and, additionally, the cross-validation process need more computational time than the traditional FDH and DEA techniques.

We finish this section by mentioning several lines that state interesting avenues for further research with Support Vector Frontiers. The first one is the possibility of extending the new technique to the context of

producing multiple outputs. In this respect, Vazquez and Walter (2003) extended SVR by considering the so-called Cokriging method, which is a multi-output version of Kriging that exploits the correlations due to the proximity in the space of factors and outputs; while Zhang and Zhou (2013) presented a multi-output Support Vector Regression approach based on problem transformations (see also the survey by Borchani et al., 2015). Another interesting line of research could be adapting certain existing methodologies in the field of SVM for ranking the importance of covariables to the context of production functions (see, for example, Chan and Lin, 2008). Additionally, we resorted in this paper to the ℓ_1-norm to define a linear optimization program, in the line of FDH and DEA. However, it is possible to use other norms to improve the results associated with the accuracy of the SVM predictions (see, for example, Blanco et al., 2020). Finally, an evident research line to be followed is the application of the new approach to real databases in different empirical contexts, thus checking the validity of the technique in practice.

Acknowledgments

The authors are grateful for the financial support from the Spanish Ministry for Economy and Competitiveness, the State Research Agency and the European Regional Development Fund under grant PID2019-105952GB-I00. This work was also supported by the *Generalitat Valenciana* under Grant ACIF/2020/155.

References

Afriat, S. N. (1972). Efficiency estimation of production functions. International economic review, 568-598.

Aigner, D. J., & Chu, S. F. (1968). On estimating the industry production function. The American Economic Review, 58(4), 826-839.

Aparicio, J., Pastor, J. T., Vidal, F., & Zofío, J. L. (2017). Evaluating productive performance: A new approach based on the product-mix problem consistent with Data Envelopment Analysis. Omega, 67, 134-144.

Aragon, Y., Daouia, A., & Thomas-Agnan, C. (2005). Nonparametric frontier estimation: a conditional quantile-based approach. Econometric Theory, 358-389.

Arnaboldi, M., Azzone, G., & Giorgino, M. (2014). Performance measurement and management for engineers. Academic Press.

Banker, R. D., Charnes, A., & Cooper, W. W. (1984). Some models for estimating technical and scale inefficiencies in data envelopment analysis. Management Science, 30(9), 1078-1092.

Banker, R. D., & Maindiratta, A. (1992). Maximum likelihood estimation of monotone and concave production frontiers. Journal of Productivity Analysis, 3(4), 401-415.

Banker, R. D. (1993). Maximum likelihood, consistency and data envelopment analysis: a statistical foundation. Management science, 39(10), 1265-1273.

Blanco, V., Puerto, J., & Rodriguez-Chia, A. M. (2020). On lp-Support Vector Machines and Multidimensional Kernels. Journal of Machine Learning Research, 21(14), 1-29.

Borchani, H., Varando, G., Bielza, C., & Larrañaga, P. (2015). A survey on multi-output regression. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 5(5), 216-233.

Chang, Y. W., & Lin, C. J. (2008). Feature ranking using linear SVM. In Causation and Prediction Challenge (pp. 53-64).

Charles, V., Aparicio, J., & Zhu, J. (2020). Data Science and Productivity Analytics. Springer.

Charnes, A., Cooper, W. W., & Rhodes, E. (1978). Measuring the efficiency of decision making units. European journal of operational research, 2(6), 429-444.

Chen, Z., Matousek, R., & Wanke, P. (2018). Chinese bank efficiency during the global financial crisis: A combined approach using satisficing DEA and Support Vector Machines☆. The North American Journal of Economics and Finance, 43, 71-86.

Cobb, C. W., & Douglas, P. H. (1928). A theory of production. The American Economic Review, 18(1), 139-165.

Coelli, T. J., Rao, D. S. P., O'Donnell, C. J., & Battese, G. E. (2005). An introduction to efficiency and productivity analysis. Springer Science & Business Media.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. Machine learning, 20(3), 273-297.

Daraio, C., & Simar, L. (2005). Introducing environmental variables in nonparametric frontier models: a probabilistic approach. Journal of productivity analysis, 24(1), 93-121.

Daraio, C., & Simar, L. (2007). Advanced robust and nonparametric methods in efficiency analysis: Methodology and applications. Springer Science & Business Media.

Debreu, G. (1951). The coefficient of resource utilization. Econometrica: Journal of the Econometric Society, 273-292.

Deprins, D., & Simar, L. (1984). Measuring labor efficiency in post offices, The Performance of Public Enterprises: Concepts and Measurements, M. Marchand, P. Pestieau and H. Tulkens.

Du, P., Parmeter, C. F., & Racine, J. S. (2013). Nonparametric kernel regression with multiple predictors and multiple shape constraints. Statistica Sinica, 1347-1371.

Efron, B. (1979). Bootstrap methods: another look at the Jackknife. Annals of Statistics, 7(1), 1-26.

Esteve, M., Aparicio, J., Rabasa, A., & Rodriguez-Sala, J. J. (2020). Efficiency Analysis Trees: a New Methodology for Estimating Production Frontiers through Decision Trees. Expert Systems with Applications, 113783.

Farahmand, M., Desa, M. I., & Nilashi, M. (2014). A combined data envelopment analysis and support vector regression for efficiency evaluation of large decision making units. International Journal of Engineering and Technology (IJET), 2310-2321.

Färe, R., Grosskopf, S., & Lovell, C. K. (1985). The measurement of efficiencies of production. Springer Netherlands.

Färe, R., & Primont, D. (1995). Multi-Output Production and Duality: Theory and Applications. Kluwer Academic Publishers.

Farrell, M. J. (1957). The measurement of productive efficiency. Journal of the Royal Statistical Society: Series A (General), 120(3), 253-281.

Friedman, J., Hastie, T., & Tibshirani, R. (2001). The elements of statistical learning. New York: Springer series in statistics.

Hall, P., & Huang, L. S. (2001). Nonparametric kernel regression subject to monotonicity constraints. Annals of Statistics, 624-647.

Henderson, D. J., & Parmeter, C. F. (2009). Imposing economic constraints in nonparametric regression: survey, implementation, and extension. Advances in Econometrics, 25, 433-69.

Kao, H. Y., Chang, T. K., & Chang, Y. C. (2013). Classification of hospital web security efficiency using data envelopment analysis and support vector machine. Mathematical Problems in Engineering, 2013.

Kerstens, K., O'donnell, C., & Van de Woestyne, I. (2019). Metatechnology frontier and convexity: A restatement. European Journal of Operational Research, 275(2), 780-792.

Khezrimotlagh, D., Zhu, J., Cook, W. D., & Toloo, M. (2019). Data envelopment analysis and big data. European Journal of Operational Research, 274(3), 1047-1054.

Kuosmanen, T., & Johnson, A. L. (2010). Data envelopment analysis as nonparametric least-squares regression. Operations Research, 58(1), 149-160.

Kuosmanen, T., & Johnson, A. (2017). Modeling joint production of multiple outputs in StoNED: Directional distance function approach. European Journal of Operational Research, 262(2), 792-801.

Lee, C. Y., & Cai, J. Y. (2020). LASSO variable selection in data envelopment analysis with small datasets. Omega, 91, 102019.

Misiunas, N., Oztekin, A., Chen, Y., & Chandra, K. (2016). DEANN: A healthcare analytic methodology of data envelopment analysis and artificial neural networks for the prediction of organ recipient functional status. Omega, 58, 46-54.

Nayak, J., Naik, B., & Behera, H. (2015). A comprehensive survey on support vector machine in data mining tasks: applications & challenges. International Journal of Database Theory and Application, 8(1), 169-186.

O'Donnell, C. J. (2018). Productivity and Efficiency Analysis. Springer Singapore.

Parmeter, C. F., Sun, K., Henderson, D. J., & Kumbhakar, S. C. (2014). Estimation and inference under economic restrictions. Journal of productivity analysis, 41(1), 111-129.

Poitier, K., & Cho, S. (2011). Estimation of true efficient frontier of organisational performance using data envelopment analysis and support vector machine learning. International Journal of Information and Decision Sciences, 3(2), 148-172.

Shephard, G. C. (1953). Unitary groups generated by reflections. Canadian Journal of Mathematics, 5, 364-383.

Simar, L., & Wilson, P. W. (1998). Sensitivity analysis of efficiency scores: How to bootstrap in nonparametric frontier models. Management science, 44(1), 49-61.

Simar, L., & Wilson, P. W. (2000a). A general methodology for bootstrapping in non-parametric frontier models. Journal of applied statistics, 27(6), 779-802.

Simar, L., & Wilson, P. W. (2000b). Statistical inference in nonparametric frontier models: The state of the art. Journal of productivity analysis, 13(1), 49-78.

Simar, L., & Wilson, P. W. (2008). Statistical inference in nonparametric frontier models: recent developments and perspectives. The measurement of productive efficiency and productivity growth, 421-521.

Song, J., & Zhang, Z. (2009, January). Oil refining enterprise performance evaluation based on DEA and SVM. In 2009 Second International Workshop on Knowledge Discovery and Data Mining (pp. 401-404). IEEE.

Vapnik, V. (1995). The Nature of Statistical Learning Theory. Springer, New York.

Vapnik, V. (1998). Statistical Learning Theory. Wiley, New York.

Vazquez, E., & Walter, E. (2003). Multi-output suppport vector regression. IFAC Proceedings Volumes, 36(16), 1783-1788.

Yeh, C. C., Chi, D. J., & Hsu, M. F. (2010). A hybrid approach of DEA, rough set and support vector machines for business failure prediction. Expert Systems with Applications, 37(2), 1535-1541.

Zhang, M. L., & Zhou, Z. H. (2013). A review on multi-label learning algorithms. IEEE transactions on knowledge and data engineering, 26(8), 1819-1837.

Zhu, J. (2019). DEA under big data: data enabled analytics and network data envelopment analysis. Annals of Operations Research, 1-23.