Università
Ca'Foscari
Venezia

**Department
of Management**

Working Paper Series

Andrea Pontiggia and Giovanni Fasano

Data Analytics and Machine Learning Paradigm to Gauge Performances Combining Classification, Ranking and Sorting for System Analysis

**Working Paper n. 5/2021
July 2021**

ISSN: 2239-2734

Data Analytics and Machine Learning paradigm to gauge performances combining classification, ranking and sorting for system analysis

ANDREA PONTIGGIA
<*andrea.pontiggia@unive.it*>
Dept. of Management
Ca' Foscari University of Venice

GIOVANNI FASANO
<*fasano@unive.it*>
Dept. of Management
Ca' Foscari University of Venice

(July 2021)

Abstract. We consider the problem of measuring the performances associated with members of a given group of homogeneous individuals. We provide both an analysis, relying on Machine Learning paradigms, along with a numerical experience based on three conceptually different real applications. A keynote aspect in the proposed approach is represented by our data–driven framework, where guidelines for evaluating individuals' performance are derived from the data associated to the entire group. This makes our analysis and the relative outcomes quite versatile, so that a number of real problems can be studied in view of the proposed general perspective.

Keywords: Performance Analysis, Data Analytics, Support Vector Machines, Human Resources

JEL Classification Numbers: M51, C38

MathSci Classification Numbers: 91C99, 90C30

Correspondence to:

Giovanni Fasano	Dept. of Management, Ca' Foscari University of Venice S.Giobbe, Cannaregio 873 30121 Venezia, Italy
Phone:	[+39] (041)-234-6922
E-mail:	fasano@unive.it

1 Introduction

Managing the performance is a critical operational concern of nearly every organization. However, simplified assumptions about the role of measurement in organizations attribute the primary role of performance measurement of providing managers with a means of control. Many people recognize and acknowledge the shortcomings of the performance measurement systems adopted by firms [1]. Managers question the validity and effectiveness of the common practice of performance appraisal. Moreover new and emerging organizational forms emphasize the need to rethink how to manage people [2].

A useful starting point is represented by the methods applied and not only the measure or criteria adopted. There is a hidden value to apply new computational approaches. The data produced as a result of performance measurement can be used to challenge the assumptions that managers hold about how their businesses operate [3]. Machine Learning (ML) methods are a powerful toolkit for discovering and exploring consistent patterns in quantitative data. The patterns identified by ML could be used for innovating the perspective and can contribute to overcome some of current limitations in the performance appraisal processes. To demonstrate the application of ML for pattern discovery, we implement ML algorithms to study the performances in two organizational settings (case study -1- and -2-) and one market-based context (case study -3-). To our knowledge ML methods represent an under utilized set of methods in the human resources management. Adoption of these methods could be improved by testing and validating applications of ML to performance analysis.

This paper attempts to use Support Vector Machines using data from three studies, to demonstrate the application of an exploratory tool to discover robust patterns in quantitative data. It addresses two issues: first, to validate the application of supervised ML methods for measuring the performance using quantitative data. Second, to provide a guidance on further research that uses such computational methods, to improve the capabilities for allocating resources and managing the performances. Discovering new and robust empirical patterns from the performance measures might encourage the management to rethink human resources systems such as performance evaluation processes and reward and development policies.

The structure of this paper is in accordance with the next guidelines. In the next Section 2 we detail the organizational context where we embed our analysis and the real problems we aim at solving. Section 3 reports a very brief summary of standard paradigms from ML (i.e. Supervised, Unsupervised and Semi-Supervised Learning). In Section 4 we clarify our terminology, in terms of performance measures for classification, sorting and ranking, respectively. Section 5 is devoted to clearly state our Support Vector Machine (SVM)-based classification framework from semi-supervised learning. In Section 6 we report three real applications where our proposal for performance measure can be fruitfully applied, including a comparative analysis of the results. Finally, Section 7 proposes some conclusions and indicates the guidelines for future work[1].

[1] As regards the symbols adopted in the paper, we use $\mathbb{R}^p$ to represent the set of the real p-vectors, while for the sake of simplicity $\|x\|$ is used to indicate the Euclidean norm in place of $\|x\|_2$. Given the set $A \subseteq \mathbb{R}^p$, we indicate by $conv(A)$, $int(A)$, $\partial(A)$ and $|A|$, respectively the set of all the convex combinations of points in A, the interior of A, the boundary of A and the cardinality of A. The Euclidean distance of the point $\bar{x} \in \mathbb{R}^p$ from the hyperplane π of $\mathbb{R}^p$ is indicated by $d[\bar{x}, \pi]$. Given the n-real vectors x and y, with $< x, y >$ we indicate their standard inner product. Given the random variable z, with $\mathbb{E}(z)$ we represent its expected value. To avoid any possible confusion, given the sequence $\{x_i\}$, with $x_i \in \mathbb{R}^p$, the entries

	Low structuring of the task	High structuring of the task
Low dependence:		Setting of Case study -1-
High dependence:	Settings of Case study -2-	Settings of Case study -3-

Table 1: Organizational Context and Setting of the studies

2 Our problem: analysis of organizational context level

In recent years, ML has become increasingly popular for performing analysis and generating predictions based on data, in a variety of different managerial areas. In this paper, we have focused on a specific type of model personalization for estimation of the performance in three different contexts, applying the SVM perspective. We focus on measurement of performance not considering the relationship between the performance and the reward or the factors which determine the performance. Our approach is directed to measurement and not to optimization of the outcomes [4]. Two dimensions described the research setting: the *level of interdependence* and the *degree of structuring* of the task [5].

In designing our experiment and analysing the three case studies we distinguish between two different (opposite) configurations, based on the level of interdependence of subjects' task. The first one shows no dependence from others' contributions and collaboration. The performance is independent of others' activities. The individual performance is not influenced by external factors and does not require any cooperation or collaboration. The activities are mainly executed individually [6]. The second situation is the opposite: the individual performance is largely dependent on the other subjects. Collaborative behaviors are requested in order to achieve the goal.

High or low dependence is the extent to which unit personnel are dependent upon one another to perform their individual jobs. Generally speaking, the higher the number of one-person jobs and the greater the degree of collaboration and the greater the interdependence. In the case study 1 and 2 the level of interdependence is also described by the presence or absence of a set of routines, procedures and the mandatory use of specific technical devices and applications. Also it is possible to observe the intensity of interdependence analyzing the flow: in the high dependence the process is sequential and preset, in low interdependence the process is loosely coupled, reciprocal and team-based [7].

The second variable to describe the context refers to the level of structuring of task. Structuring stands for basic characteristics of the task defined by subjects' jobs [8]. High structuring is defined by a set of defined and well described, low level of variability, low level of complexity (e.g. defined by knowledge and skills requested by the task) and the presence of procedures and the absence of problem solving or innovative behaviors required. At the opposite, low structuring is described by low standardization (no routine), general behavioral codes and operation rules, high autonomy and discretion [9]. The overall organizational context can be described as in Table 1.

Another difference in the research path concerns with the focus of the application of the SVM methods. In order to validate and test the computational approach, each case study presents a different focus. The longitudinal analysis (see Table 2) measures the level of performance in dif-

of x_i will be indicated as $(x_i)_1$, $(x_i)_2$, etc.

	Main Focus of SVM application
Case study -1-	Longitudinal Analysis
Case study -2-	Simulation of performances
Case study -3-	Forecast of performances

Table 2: Application of the analysis by SVMs

ferent times, so that it allows to compare the observations. The second case study simulates the consequences of perturbations and shows the effects of this variation on the global and aggregate performances. The last case study, a market-based analysis of performance, shows the application of SVM to forecast the performances.

3 Supervised, unsupervised and semi-supervised learning

The main purpose of this section is that of clarifying the context of our proposal, in view of the current literature of ML approaches. Observe that a thorough and complete survey of ML paradigms, along with those optimization techniques for their implementation, is far beyond the scope of the present section. Nevertheless, for the convenience of the reader we are committed to give at least a clear idea about the context our analysis draws inspiration from. The interested reader can easily find additional and complete details in [21, 27, 30],

Broadly speaking we can say that ML approaches deal with the analysis of a set of N points $\{x_1,\dots,x_N\}$ (with $x_i \in \chi \subseteq \mathbb{R}^p$, $i = 1,\dots,N$), along with the set of labels $\{y_1,\dots,y_N\}$ (with $y_i \in \mathcal{Y} \subseteq \mathbb{R}$, $i = 1,\dots,N$). In this regard, following a traditional taxonomy for ML, depending on the analysis which is carried on the pairs $\{x_i, y_i\}$, we distinguish among the following frameworks: *unsupervised learning*, *supervised learning* and *semi-supervised learning*.

Unsupervised Learning ultimately deals with the assessment of the density function $p_X(x)$ which has generated the points x_i, $i = 1,\dots,N$, ignoring the labels $\{y_i\}$. This definition is quite general, but encompasses a number of practical problems where unsupervised learning is adopted. Among the last problems we find *clustering*, where the points are grouped trying to capture a likely structure behind the sequence $\{x_1,\dots,x_N\}$. In this regard, as a purely indicative example, the reader may consider that the mean point $\bar{x} = \sum_{i=1}^{N} x_i$ summarizes some of the statistical properties of the series $\{x_1,\dots,x_N\}$, and allows to identify possible outliers. Thus, identifying $\bar{x}$ as a center of a cluster for the points $\{x_i\}$ may represent a natural choice. As another example of unsupervised learning approaches, the reader may consider the *principal component analysis*. Here, an essential structure behind the sequence $\{x_1,\dots,x_N\}$ is sought, in order to summarize its statistical properties, identifying privileged lines/hyperplanes around which the points $\{x_i\}$ tend to cluster.

Supervised Learning is definitely focused on validating a structure behind the points $\{x_i\}$, strongly relying on the labels $\{y_i\}$. This task can be equivalently formulated by the goal of determining the statistical properties (i.e. the density function $p_Y(y)$) associated with a map between the sets χ and $\mathcal{Y}$. Similarly to unsupervised learning, there is plenty of applications where the use of labels $\{y_i\}$ becomes an essential tool. For instance, *regression problems* correspond to assume

$\mathcal{Y} \equiv \mathbb{R}^m$, while *classification problems* refer to the case where $\mathcal{Y}$ contains isolated points (i.e. any label y_i may assume only a discrete number of real/categorical values). In general, the assessment of the density function $p_Y(y)$ may be obtained in two distinctive ways. We can first assess $p_X(x)$ and then use the Bayes Theorem to obtain $p_Y(y)$ (*generative algorithms*), or we can directly compute/estimate $p_Y(y)$ (*discriminative algorithms*). As an example of the last class of methods we have SVMs, which will be widely adopted in this paper. They are methods which subdivide the points $\{x_i\}$ into classes (classification methods), in order to simply assess the probability for a point to belong to one of the classes (e.g. in case just two classes are considered, SVMs decide if this probability is below/above 1/2). SVMs heavily rely on optimization techniques.

Semi-supervised Learning (or more in general Partially Supervised Learning) deals with using the set $\{x_i\}$ and only a subset of $\{y_i\}$ in the learning process. This can be accomplished in different ways. In particular, in this paper we adopted a semi-supervised learning procedure which is divided into two phases. We initially apply a *clustering analysis* (unsupervised learning) to identify the subsets $L_{\max}$ and $L_{\min}$ of the set $\{x_1, \ldots, x_N\}$. Then, we solve a series of quadratic programming formulations associated with a sequence of SVMs (supervised learning), showing that a wide range of applications can be modelled using our hybrid partially supervised approach.

4 Measures of performance: classification, sorting and ranking

The following paragraphs introduce a more rigorous approach to clarify the ideas underpinning our proposal. As a result, both our analysis and our final ML scheme will be way more motivated. In this regard, to possibly simplify both the taxonomy and our analysis, in the light of the applications detailed in the sequel (see Section 6), we consider a group of individuals (*fellows*) and we say that the vector $x_i \in \mathbb{R}^p$ summarizes the performance of the i-th fellow. Moreover, the set of all the fellows will be addressed hereafter as the *unit*. Given that, we are interested about:

(1) providing guidelines which allow to gauge the i-th fellow performance, on the basis of the whole information associated with *all* the other fellows,

(2) giving perspective indications to each fellow in order to *comparatively* enhance her/his performance with respect to the other fellows,

(3) providing an indicator which possibly summarizes the performance of the overall unit,

(4) determining an effective procedure to (possibly) allow comparisons among different units,

(5) tuning a procedure which, to some extent, does not merely include standard paradigms from cluster analysis, but basically allows classification, ranking and sorting of all the fellows,

(6) allowing comparison among different time-dependent scenarios for the unit,

(7) introducing an evaluation process whose perspective relies on ML paradigms and goes beyond mere multicriteria methodologies [24],

our procedure uses information associated with the finite number of points $\{x_i\} \equiv \chi \subset \mathbb{R}^p$, $p \in \mathbb{N}$, to generate the finite sequences of sets $\{A_k\}$, $\{B_k\}$, $k = 0, \ldots, m$, such that

$$\begin{array}{ll} (i) & \emptyset \subset A_k, B_k \subseteq \chi, \\ (ii) & A_{k+1} \supseteq A_k, \\ (iii) & B_{k+1} \supseteq B_k, \\ (iv) & |A_{k+1}| + |B_{k+1}| = |A_k| + |B_k| + 1, \\ (v) & A_m \cup B_m = \chi, \end{array} \tag{1}$$

in order to evaluate the performance of all the fellows of the unit. We assess an iterative procedure, with respect to the subscript k, where the i-th fellow is eventually associated with the pair (x_i, y_i), so that eventually $x_i \in A_m$ (with $y_i = +1$) or $x_i \in B_m$ (with $y_i = -1$). Before getting into details we urge to give a better intuition behind the choice of the sequences $\{A_k\}$ and $\{B_k\}$ in (1). In particular, we aim at generating $\{A_k\}$ and $\{B_k\}$ so that each of these sequences contains sets with a non decreasing number of elements. Moreover, eventually (when $k = m$) the sets A_m and B_m will cover χ. The key aspect of this structured sequences is the following: the sets A_0 and B_0 will contain fellows of the unit endowed with *extreme performance*, i.e. namely A_0 will contain fellows with the *most* appealing performance, while B_0 will contain fellows with the *least* appealing performance. Then, a ML approach will select one of the fellows in $\chi \setminus \{A_0 \cup B_0\}$, say $\bar{z}$, and based on the elements in $A_0 \cup B_0$ will decide about one of the following two scenarios

$$\begin{cases} A_1 = A_0 \cup \{\bar{z}\} \\ B_1 = B_0 \end{cases} \qquad \begin{cases} A_1 = A_0 \\ B_1 = B_0 \cup \{\bar{z}\}. \end{cases}$$

After setting $k = k + 1$ the procedure will be iteratively repeated, so that eventually we will have $A_m \cup B_m = \chi$. We remark that in both the above scenarios, at step k the ML approach will induce *classification* for the fellow $\bar{z}$, either in A_k or B_k, uniquely based on information contained in $A_k \cup B_k$. In addition, setting $\bar{z} \in A_{k+1}$ (respectively $\bar{z} \in B_{k+1}$) the algorithm determines a *ranking* between $\bar{z}$ and the points in B_k (respectively $\bar{z}$ and the points in A_k), since the performance of $\bar{z}$ will be considered incompatible with respect to the performance of the fellows in B_k (respectively A_k). Moreover, at step k, a *sorting* system can be considered among the fellows in the set A_{k+1} (respectively B_{k+1}), based on the priority that the newly added fellow $\bar{z}$ has with respect to the other elements in the set.

We strongly remark that the above iterative procedure aims at implementing a natural process of simplification in order to learn from data. Indeed, it is based on first identifying extreme performance (i.e. A_0 and B_0) within the unit, which is an easier task with respect to classifying the performance of all the fellows at once. Then, learning from the trusted classified data contained in A_0 and B_0, it further classifies the performance of an additional fellow, so that the bunch of trusted data, now contained in A_1 and B_1, is enhanced. The process is repeated until all the fellows' performance has been classified. Equivalently, it means that on the overall, the cumbersome classification/ranking/sorting process of the whole number of $|\chi|$ fellows in the unit is pursued after just performing $|\chi| - |A_0 \cup B_0|$ binary classifications, which represent a much simpler task. In other words, the analogical system of grading/rewarding for an entire group of fellows can be reduced to a sequence of binary choices, whose reliability increases with the index k, inasmuch as the set of already classified points $A_k \cup B_k$ increases its cardinality (i.e. the size of the training dataset) at each

step. Examples of similar approaches can be found in the literature in [15, 31, 32].

As a visual representation, in Figure 1 we plot N points $x_1,\ldots,x_N$, with $x_i \in \mathbb{R}^2$, associated with a set of fellows, with respect to the two criteria C_1 and C_2. In the example we have the following range of values for the entries of x_i (see also Section 6.2)

$$\begin{cases} 2.5 \leq (x_i)_1 \leq 3.7, & i = 1,\ldots,N, \\ \\ 10 \leq (x_i)_2 \leq 100, & i = 1,\ldots,N. \end{cases}$$

Our iterative ML procedure is based on SVMs and needs to preliminarily compute the subsets $L_{\max}$ (i.e. A_0) and $L_{\min}$ (i.e. B_0) of χ. In this regard, the large red bullets ($L_{\max}$) at North-East position in Figure 1 represent a (weak) Pareto front, including those points from the maximization of both the criteria C_1 and C_2. Conversely, the large cyan points ($L_{\min}$) at South-West position are also associated with a (weak) Pareto front; however, unlike $L_{\max}$, they are computed adopting the minimization of both the criteria C_1 and C_2. For a more formal definition of these last sets of points, the reader can consider that the point (fellow) with coordinates $(\bar{x})_1,\ldots,(\bar{x})_p$ will be classified as a point in $L_{\max}$ if it satisfies the properties (*non-dominated point* with respect to joint maximization)

$$((\bar{x})_1,\ldots,(\bar{x})_p) \in L_{\max} \text{ if } \begin{array}{l} \nexists j \in \{1,\ldots,N\}, \text{ with } ((x_j)_1,\ldots,(x_j)_p) \neq ((\bar{x})_1,\ldots,(\bar{x})_p), \\ \text{s.t. } (x_j)_h > (\bar{x})_h, \quad h = 1,\ldots,p. \end{array} \tag{2}$$

Similarly, the point with coordinates $(\hat{x})_1,\ldots,(\hat{x})_p$ will be classified as a point in $L_{\min}$ if it satisfies the properties (*dominated point* with respect to joint minimization)

$$((\hat{x})_1,\ldots,(\hat{x})_p) \in L_{\min} \text{ if } \begin{array}{l} \nexists j \in \{1,\ldots,N\}, \text{ with } ((x_j)_1,\ldots,(x_j)_p) \neq ((\hat{x})_1,\ldots,(\hat{x})_p), \\ \text{s.t. } (x_j)_h < (\hat{x})_h, \quad h = 1,\ldots,p. \end{array} \tag{3}$$

As a general idea, in several practical applications it is desirable that each fellow scores as high as possible in all the criteria, so that the point associated with her/him will be non-dominated with respect to joint maximization (i.e. it belongs to $L_{\max}$, showing the *best performance*). On the contrary, it is highly undesirable that a fellow scores poorly with respect to all the criteria, because the point associated with her/him will possibly be part of dominated pairs (i.e. it belongs to $L_{\min}$, showing the *worst performance*). Therefore, in the light of the guidelines (1)-(7), all the fellows will be eventually involved in our iterative process, which includes *classification* (of all the fellows), *ranking* (between the identified classes) and *sorting* (among the fellows in each class). In particular, we are going to show that by our iterative ML procedure, at each step k we can pursue:

- *classification*: through a binary separation (partition) process, applied to a subset of the fellows, we identify a pair of classes (namely A_{k+1} and B_{k+1} using the taxonomy in (1)). The separation process uses SVMs, which represent an essential tool of ML;

- *ranking*: the fellows in each of the classes A_{k+1} and B_{k+1} are easily subject to a natural ranking. Indeed, A_{k+1} includes only fellows with the certified best performance up to step k, while B_{k+1} encompasses only fellows corresponding to the classified worst performance up to step k. Thus, each fellow in A_{k+1} retains a 'natural supremacy' with respect to all the fellows in B_{k+1}. Equivalently each of the fellows in B_{k+1} is outranked by any fellow in A_{k+1};

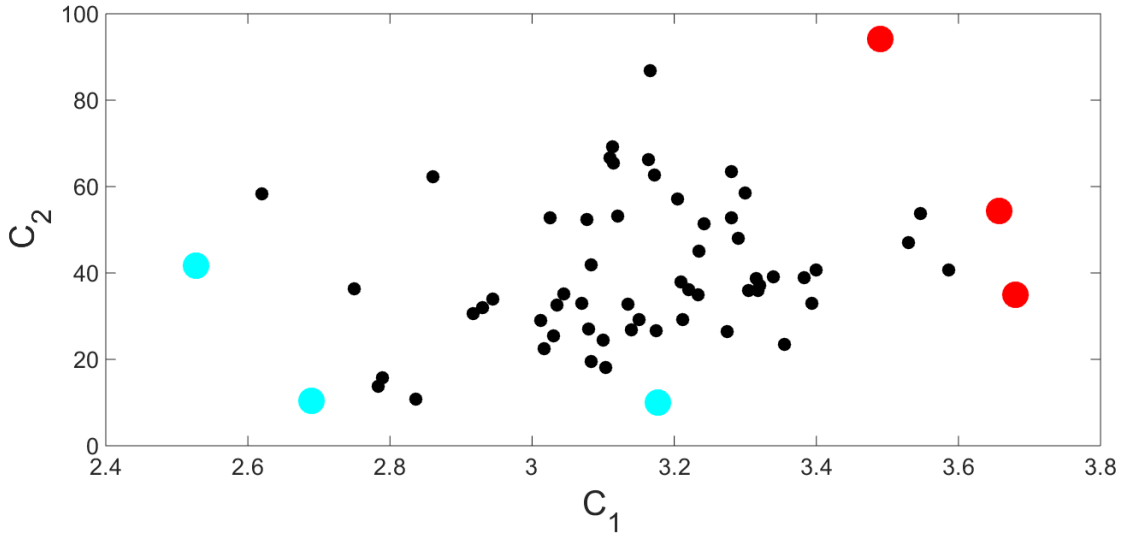

Figure 1: Each bullet represents a point associated with a unit fellow. Large bullets represent special fellows included in Pareto fronts. Red large bullets represent the Pareto front $L_{\max}$, cyan large bullets belong to the Pareto front $L_{\min}$.

- *sorting*: it is possible to couple our SVM approach with some additional metrics (endowed with the classic properties of measures), in order to discriminate among fellows within the same class. To give an example, we first recall that once the fellow $\bar{z}$ is included in a class, then it will no more be removed, by the SVM-based procedure, from that class. Thus, given the class A_{k+1} (similarly for B_{k+1}) at step k and the fellow $\bar{z} \in A_{k+1}$, a sorting process among the elements in A_{k+1} can be determined depending on the step $i \leq k$ at which $\bar{z}$ was included in A_{k+1} (e.g. if $\bar{z}$ and $\bar{w}$ were included in A_{k+1} respectively at step i and step j, with $i < j \leq k$, then we may consider $\bar{z}$ *preferable to* $\bar{w}$ - see also Lemma 5.2).

We complete this section observing that by the definitions (2) and (3), the computational cost for evaluating each of the sets $L_{\max}$ and $L_{\min}$ is polynomial, being given by $O(|\chi|(|\chi|-1)p)$ (indeed, any of the $|\chi|$ points requires to be compared with all the other points, and each comparison has a cost equal to p).

5 A Machine Learning perspective: the SVM classification framework

In the current section we describe approaches for our classification problems, by using separating hyperplane classifiers. Linear separation by hyperplane classifiers provides a procedure to construct linear decision boundaries, that possibly attempt to separate points (i.e. our fellows) into two different classes (see Figure 2), where the convex hull of the points in a given class do not include any of the points in the other class. In case the separation is not allowed (because of the relative geometry of the points), then a *misclassification measure* is in any case considered, in order to reduce misclassification errors as much as possible. Conversely, in case full separation by a linear boundary

is possible, then the use of linear classifiers also provides a *measure of successful classification*, by exploiting information associated with special points, namely *support vectors*, whose role is discussed later on (see Figure 3, too). Figure 2–*left* depicts two sets of points (green bullets and red stars) which are clearly separated by any of the dashed lines. On the contrary, in Figure 2–*right* any of the dashed lines is unable to separate green bullets from red stars.

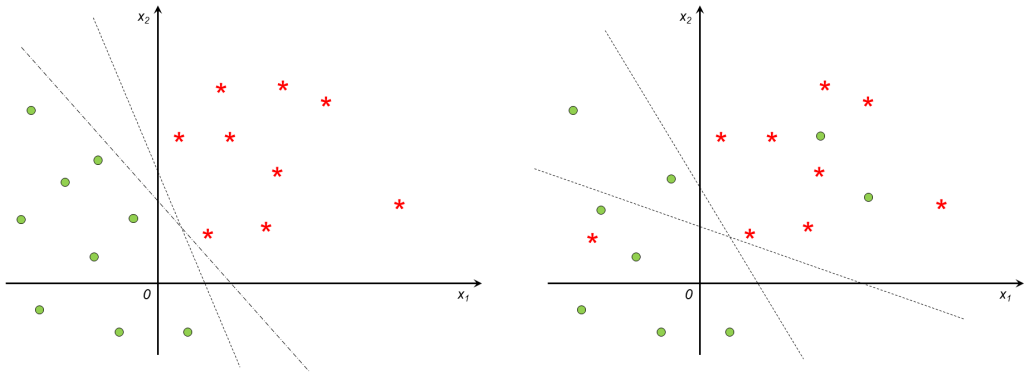

Figure 2: Examples of linearly separable (*left*) and linearly non separable (*right*) sets of points (red stars and green bullets). Both the dashed and dashed-dotted line on the left are able to separate the sets of points. Conversely, any line on the right is unable to separate the sets.

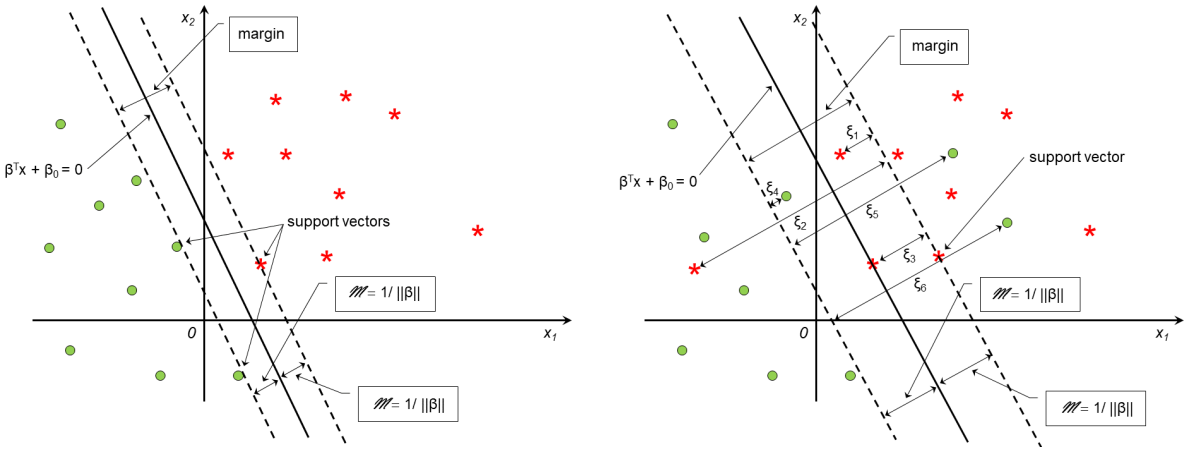

Figure 3: (*left*) The two sets of points (bullets and stars) are linearly separable by the hyperplane $\beta^T x + \beta_0 = 0$ with a margin $W = 2/\|\beta\|$. (*right*) The two sets of points are not linearly separable by the hyperplane $\beta^T x + \beta_0 = 0$, even though the quantity $2/\|\beta\|$ yet represents a suitable generalization of the separation margin $\mathcal{W}$ between the sets. The quantities $\{\xi_i\}$ represent misclassification errors.

Definition 5.1 *Given the points $x_i \in \mathbb{R}^p$, $i = 1,\ldots,N$, and the values $y_i \in \{-1,+1\}$, $i = 1,\ldots,N$, let us define the nonempty sets $A = \{x_i \ : \ y_i = +1\}$ and $B = \{x_i \ : \ y_i = -1\}$. Then, we say that A and B are linearly separable if there exists a hyperplane $H(\beta,\beta_0;x) = 0$, with coefficients $\beta \in \mathbb{R}^p$*

and $\beta_0 \in \mathbb{R}$*, such that*

$$\begin{cases} H(\beta,\beta_0;x_i) > 0, & \forall i \ : \ y_i = +1 \\ \\ H(\beta,\beta_0;x_i) < 0, & \forall i \ : \ y_i = -1. \end{cases} \tag{4}$$

The next proposition provides a fundamental result of linear separability for sets containing finitely many points.

Proposition 5.1 ([13]) *Two sets* $A, B \subset \mathbb{R}^p$ *are linearly separable if and only if*

$$conv(A) \cap conv(B) = \emptyset.$$

In Figure 3 we report standard examples of linearly separable (*left*) and linearly non-separable (*right*) sets of points, being the hyperplane $H(\beta,\beta_0;x) = \beta^T x + \beta_0 = 0$ the separation hyperplane. We can clearly observe that, according with Definition 5.1 and Proposition 5.1, in Figure 3–*left* the (green) bullets are all below the separation hyperplane, while the (red) stars are all above the separation hyperplane. Conversely, in Figure 3–*right* we are unable to locate *all* the bullets and stars in two supplementary half–spaces. Hereafter in this section we shortly describe the approach by *Support Vector Machine*, which is a standard ML technique used to possibly construct an optimal separating hyperplane between two classes of points. In our analysis we also encompass the nonseparable case, where we duly take into account those misclassified points in Figure 3–*right*, corresponding to the nonzero misclassification errors $\{\xi_i\}$.

We consider the following set of N training points, defined by their pairs

$$(x_1,y_1),\ldots,(x_N,y_N), \tag{5}$$

where $x_i \in \mathbb{R}^p$ and $y_i \in \{-1,1\}$. Then, we introduce the unit vector $\beta \in \mathbb{R}^p$ with $\|\beta\| = 1$, the constant value $\beta_0 \in \mathbb{R}$ and the hyperplane

$$\pi \ : \quad \{x \in \mathbb{R}^p \ : \ \beta^T x + \beta_0 = 0\}. \tag{6}$$

The hyperplane π possibly induces a classification rule for the pairs (x_i,y_i), by using the classifier

$$C(x) = sign\{\beta^T x + \beta_0\}.$$

I.e. in case the pair (x_i,y_i) satisfies $C(x_i) = y_i$ and $y_i = -1$, then the point x_i will belong to the class C^-; conversely, in case the pair (x_i,y_i) satisfies $C(x_i) = y_i$ and $y_i = +1$, then the point x_i will belong to the class C^+, being $C^- \cap C^+ = \emptyset$.
When C^- and C^+ are linearly separable, then regardless of the value of y_i the pair (x_i,y_i) evidently fulfills the relation

$$(\beta^T x_i + \beta_0) y_i \geq 0.$$

Hence, in the separable case we might be interested to find a separation hyperplane which yields the largest possible distance (i.e. the *margin* $\mathcal{M} \geq 0$), between the points in either the classes C^-

and $\mathcal{C}^+$ and the separation hyperplane. The resulting supervised learning problem reduces to the mathematical programming formulation[2]

$$\begin{array}{ll} \max\limits_{\beta,\,\beta_0} & \mathcal{M} \\ \text{s.t.} & (\beta^T x_i + \beta_0) y_i \geq \mathcal{M}, \qquad i = 1,\ldots,N, \\ & \|\beta\| = 1. \end{array} \tag{7}$$

As by Figure 3, when maximizing the margin $\mathcal{M}$ we attempt to determine the largest distance between any vector in the set $\{x_i\}$ and the separation hyperplane $(\beta^*)^T x_i + \beta_0^* = 0$, which solves the optimization problem (7). Being $\mathcal{M} = 1/\|\beta\|$ [13, 14], the last problem can be equivalently reformulated as

$$\begin{array}{ll} \min\limits_{\beta,\,\beta_0} & \|\beta\| \\ \text{s.t.} & (\beta^T x_i + \beta_0) y_i \geq 1, \qquad i = 1,\ldots,N, \end{array} \tag{8}$$

which represents a standard convex linearly constrained mathematical programming approach to the separation problem, in case the convex hulls of the sets $\mathcal{C}^-$ and $\mathcal{C}^+$ have an empty intersection (Figure 3–*left*). We strongly remark that the problem (8) is convex, being equivalent to a quadratic linearly constrained optimization problem, whose local solutions are therefore also global solutions.

Figure 4 describes the graphical interpretation for the solutions of problem (8), where we assume (without loss of generality) that $x_i \geq 0$, for any index i. In particular, in Figure 4 we project the feasible set of problem (8), i.e. a polyhedron, over the plane described by β_0 and β_j. Thus, any line in this picture corresponds to a hyperplane, associated with a different inequality constraint in (8). Hence, the inequality $\beta^T x_i + \beta_0 > 1$ (respectively $-\beta^T x_i - \beta_0 > 1$) corresponds to the constraint in (8) associated with the pair (x_i, y_i) where $y_i = +1$ (respectively $y_i = -1$), and addresses interior points of the feasible set. Conversely, the equality $\beta^T x_i + \beta_0 = 1$ (respectively $-\beta^T x_i - \beta_0 = 1$) again corresponds to the constraint in (8) associated with the pair (x_i, y_i) where $y_i = +1$ (respectively $y_i = -1$). However, now this hyperplane identifies points on the boundary of the feasible set of (8). The dashed arrows in Figure 4 represent normal vectors to the constraints, indicating those points which fulfill the constraint: without loss of generality a similar picture can be obtained with a different orientation of such vectors. Observe that the projection of the feasible set (polyhedron), on the plane identified by the axes β_0 and β_j, is represented by the shaded polyhedron $\mathcal{P}$ in the picture. Finally, since the task in (8) is that of minimizing the Euclidean norm of β, the segment of length $|\beta_j^*|$ in the picture represents indeed the solution.

Observation 5.1 *Let be given the optimal hyperplane π^* from (8)*

$$\pi^* \equiv \left\{ x \in \mathbb{R}^p \;:\; (\beta^*)^T x + \beta_0^* = 0 \right\}.$$

Then, in case the additional pair (x_{N+1}, y_{N+1}) were included, such that the distance $d[x_{N+1}, \pi^*]$ were strictly smaller than the margin (i.e. $1/\|\beta^*\|$), then in Figure 4 the area of the polyhedron $\mathcal{P}$ would be reduced.

[2]Observe that given the hyperplane π in (6), the distance $d(\bar{x}, \pi)$ of the point $\bar{x} \in \mathbb{R}^p$ from the hyperplane π is simply given by

$$d(\bar{x}, \pi) = \frac{|\beta^T \bar{x} + \beta_0|}{\sqrt{\|\beta\|^2}}.$$

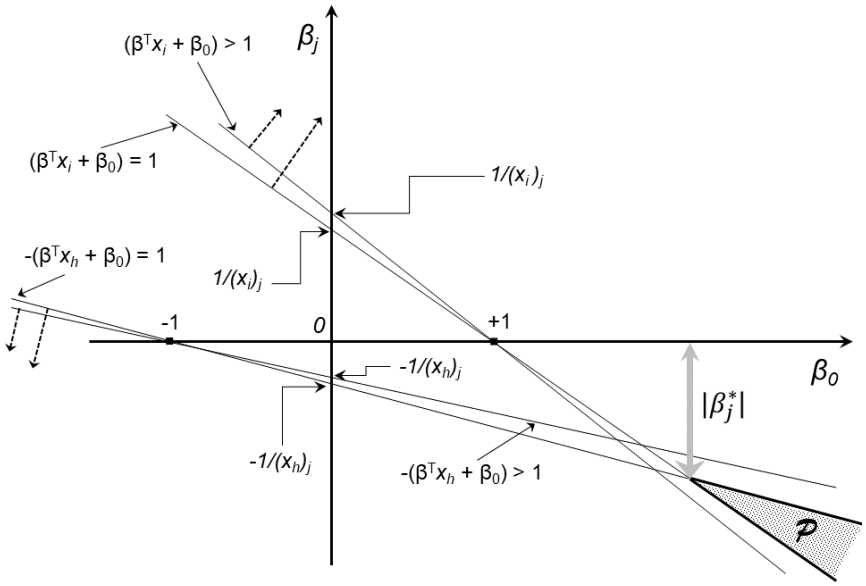

Figure 4: The solution of the convex problem (8): $\mathcal{P}$ represents the projection of the feasible polyhedron in (8) over the plane $\beta_0 - \beta_j$.

Now assume that the convex hulls of $\mathcal{C}^-$ and $\mathcal{C}^+$ have a non–empty intersection (i.e. we are in the non separable case of Figure 3–*right*). In the light of the previous analysis, we can similarly attempt to maximize yet a *generalized* margin $\mathcal{M}$ by allowing some data to be in the wrong half–space, i.e. allowing misclassification. On this purpose, following a standard mathematical programming approach from the literature (see [13]) we introduce the vector of slack variables $\xi = (\xi_1, \ldots, \xi_N)$, $\xi_i \geq 0$, $i = 1, \ldots, N$, and modify accordingly the constraints in the optimization problem (8) as

$$(\beta^T x_i + \beta_0) y_i \geq \mathcal{M}(1 - \xi_i), \qquad i = 1, \ldots, N, \tag{9}$$

being $\mathcal{M}\xi_i$ the *portion* of the margin we allow for the misclassification of the pair (x_i, y_i). In other words, the quantity ξ_i in the i-th constraint $(\beta^T x_i + \beta_0) y_i \geq \mathcal{M}(1 - \xi_i)$ yields the amount of margin by which the point x_i is in the wrong half–space, so that $(\beta^T x_i + \beta_0) y_i \not\geq \mathcal{M}$. This also suggests that, setting ξ in order to bound the sum

$$\sum_{i=1}^{N} \xi_i,$$

we will compute the hyperplane π^* in Observation 5.1 which reduces the overall misclassification as much as possible. Moreover, by (9) the pair (x_i, y_i) will be misclassified if $\xi_i > 1$, so that we can

equivalently bound the total misclassification by solving the optimization problem

$$\begin{array}{ll} \min\limits_{\beta,\,\beta_0,\,\xi} & \|\beta\| \\ \text{s.t.} & (\beta^T x_i+\beta_0)y_i \geq 1-\xi_i, \qquad\qquad i=1,\ldots,N, \\ & \sum\limits_{i=1}^{N}\xi_i \leq C, \\ & \xi_i \geq 0, \qquad\qquad i=1,\ldots,N, \end{array} \tag{10}$$

being $C>0$ a given constant. We strongly remark that for to assessment of the optimal separating hyperplane π^* in (6), when solving (10), those points $\{x_i\}$ which are not misclassified do not play any relevant role. The last comment will turn to be of great importance in the sequel, and differentiates our perspective with respect to the Linear Discriminant Analysis (LDA) approach (see also [25]).

We also highlight that solving the formulation (10) yet preserves the level of complexity of the optimization problem (8), since in both cases we have a convex linearly constrained problem. In the light of a possible simplification of (10) by using the Wolfe dual approach (see also [26]), we equivalently reformulate (10) as

$$\begin{array}{ll} \min\limits_{\beta,\,\beta_0,\,\xi} & \|\beta\|+C\sum\limits_{i=1}^{N}\xi_i \\ \text{s.t.} & (\beta^T x_i+\beta_0)y_i \geq 1-\xi_i, \qquad i=1,\ldots,N, \\ & \xi_i \geq 0, \qquad\qquad i=1,\ldots,N. \end{array} \tag{11}$$

Note that now C represents in (11) a *penalty coefficient*, so that the larger C the closer (11) to a formulation for the linearly separable case. Indeed, intuitively speaking, when $C\to+\infty$ and (11) admits solution, then it can be easily proved that $\mathcal{C}^-$ and $\mathcal{C}^+$ are linearly separable.

Now, associating the Lagrange multipliers $\{\alpha_i\}$ to the constraints $(\beta^T x_i+\beta_0)y_i \geq 1-\xi_i,\ i=1,\ldots,N$, and the multipliers $\{\mu_i\}$ to the constraints $\xi_i\geq 0,\ i=1,\ldots,N$, respectively, we can set the Lagrange (primal) function $\mathcal{L}_P(\beta,\beta_0,\xi,\alpha,\mu)$ associated with the problem (11) as[3]

$$\mathcal{L}(\beta,\beta_0,\xi,\alpha,\mu)=\frac{1}{2}\|\beta\|^2+C\sum_{i=1}^{N}\xi_i+\sum_{i=1}^{N}\alpha_i\left[1-\xi_i-(\beta^T x_i+\beta_0)y_i\right]-\sum_{i=1}^{N}\mu_i\xi_i, \tag{12}$$

so that applying the Karush–Kuhn–Tucker optimality conditions to (11) we finally get (after some computations) the set of equalities/inequalities

$$\begin{aligned} 1-\xi_i^* &\leq \left[(\beta^*)^T x_i+\beta_0^*\right]y_i, \qquad i=1,\ldots,N, \\ \alpha_i^*\left[1-\xi_i^*-\left((\beta^*)^T x_i+\beta_0^*\right)y_i\right] &= 0, \qquad i=1,\ldots,N, \\ \mu_i^*\xi_i^* &= 0, \qquad i=1,\ldots,N, \end{aligned} \tag{13}$$

[3]Observe that in (11) we can equivalently replace the term $\|\beta\|$ in the objective function by the strictly convex one $1/2\|\beta\|^2$.

and the relations

$$\begin{aligned} &\alpha_i^* = C - \mu_i^*, && i = 1,\ldots,N, \\ &\sum_{i=1}^{N} \alpha_i^* y_i = 0, \\ &\beta^* = \sum_{i=1}^{N} \alpha_i^* x_i^* y_i, \\ &\xi_i^*,\ \alpha_i^*,\ \mu_i^* \geq 0, && i = 1,\ldots,N. \end{aligned} \tag{14}$$

Using the equalities (14) in (12), we obtain the so called (quadratic) Lagrange *Wolfe dual* objective function

$$\mathcal{L}_D(\alpha^*) = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} x_i^T x_j y_i y_j \alpha_i^* \alpha_j^* + \sum_{i=1}^{N} \alpha_i^*$$

and the relative (convex) dual maximization problem

$$\begin{aligned} &\max_{\alpha} \ -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} x_i^T x_j y_i y_j \alpha_i \alpha_j + \sum_{i=1}^{N} \alpha_i \\ &\text{s.t.} \ \sum_{i=1}^{N} y_i \alpha_i = 0, \\ &\qquad 0 \leq \alpha_i \leq C, \quad i = 1,\ldots,N, \end{aligned} \tag{15}$$

whose possible solution is given in Figure 5. We urge to remark that the introduction of the dual problem (15) simply represents an analytical rearrangement of (the primal) problem (11). Indeed, the dual variables α do not have any relationship with the unknowns β, β_0 and ξ of the primal problem. Nevertheless, the duality theory allows to prove that there exists a precise correspondence between the solution sets of primal and dual problems [26]. This fact can be fruitfully exploited, inasmuch as the user can decide to solve the easiest formulation (the dual one in our case), according with the available computational means/resources, and then retrieve the solution of the others.

Observe that in our case the solution of the convex quadratic problem (15) is definitely simpler than the solution of the convex quadratic problem (11), since just simple bound constraints are included in (15) (apart from an additional simplex constraint). Furthermore, once the solution α^* is available for (15), then (14) immediately yields the optimal vector β^* for (11) as

$$\beta^* = \sum_{i=1}^{N} \alpha_i^* x_i y_i.$$

We also observe that the non–zero entries of the vector α^* will uniquely correspond to pairs $\{(x_i, y_i)\}$ associated with the so called *support vectors* (see Figure 3–*right*). In particular, the following results hold:

- some support vectors will correspond to a positive value of ξ_i, i.e. $\xi_i^* > 0$ with $\alpha_i^* = C$, so that they will also correspond to those misclassified points which do not lie on the dashed lines of

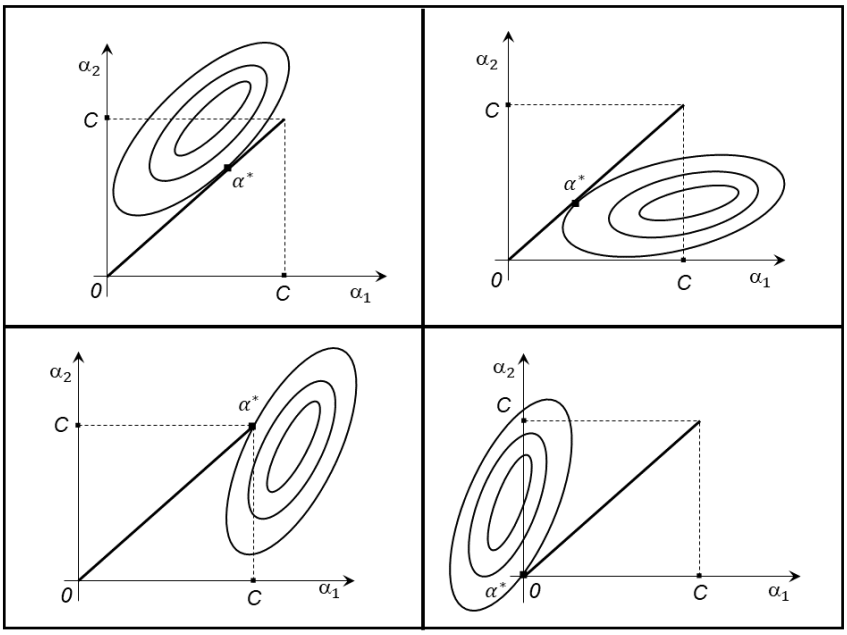

Figure 5: The possible solution α^* of the convex problem (15), whose feasible set (the thicker segment in the four quadrants) has definitely a simpler structure with respect to the feasible polyhedron in (8). Ellipses represent level curves of the quadratic objective function in (15).

Figure 3–*right*. Any of the conditions $\alpha_i^* = C$, associated with such misclassified points, can be equivalently used to compute the value of β_0 by (13)-(14);

- the remaining support vectors will correspond to a zero value of ξ_i, i.e. $\xi_i^* = 0$ with $0 < \alpha_i^* < C$, so that they will also correspond to those support vectors which exactly lie on the dashed lines of Figure 3–*right*.

A typical SVM-based linear separation problem solves the formulation (15), for a given value of C, due to its simplicity. We might be induced to set a very large value of C, which attempts to capture the case in which the two sets in our separation problem are linearly separable. However, though in principle the last consideration holds, from a computational point of view using too large values for C can require a disproportionate CPU time for computation, when large values of N are considered. This last conclusion is mainly suggested by the state-of-the-art numerical procedures which are commonly used to solve the formulation (15).

We complete this section by observing that the dual formulation (15) is subject to a very interesting generalization, in case the *linear separation* framework described in this section is replaced by a *nonlinear separation* mechanism. Indeed, it can be proved [16] that the separating hyperplane (6), such that (4) holds, can possibly be replaced by a different smooth nonlinear hypersurface, in order to (non)linearly separate the sets A and B. Broadly speaking, it means that the A and B might not fulfill Proposition 5.1 but they can be separated by a nonlinear hypersurface. This generalization may have a terrific impact in practice, since there is plenty of real applications where the sets

A and B are separable but they are not linearly separable. From a more analytical standpoint, using a nonlinear hypersurface corresponds to introduce in the dual formulation (15) a so called *kernel* $K(x_i,x_j)$, obtaining the novel optimization problem

$$\begin{aligned} \max_{\alpha} \quad & -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} K(x_i,x_j)y_iy_j\alpha_i\alpha_j + \sum_{i=1}^{N}\alpha_i \\ \text{s.t.} \quad & \sum_{i=1}^{N} y_i\alpha_i = 0, \\ & 0 \le \alpha_i \le C, \quad i = 1,\dots,N, \end{aligned} \tag{16}$$

where $K(x_i,x_j)$ is a Gram matrix. In particular, given the N points $x_1,\dots,x_N \in \mathbb{R}^p$ and the smooth function $\phi : \mathbb{R}^p \to \mathbb{R}$, the relation $K(x_i,x_j) =< \phi(x_i),\phi(x_j) >$ holds, being $< \cdot,\cdot >$ a consistent inner product. Hence, in case the function $\phi(x)$ is the identity, we simply have $K(x,x) = X^TX$, being $X = (x_1 \cdots x_N)$.

5.1 Our SVM-based classification method

In this section we describe an iterative classification process for the points $\{x_i\}$, $x_i \in \mathbb{R}^p$, $i = 1,\dots,N$, following at any step the guidelines outlined in Section 5. As a distinctive task of our proposal, we intend to assess a semi-supervised methodology (see Section 3) of ML, which possibly exploits the structure of the given set of points $\{x_i\}$. The iterative process we propose consists, at step k, of two distinct phases:

- first, we assume that each of the points $\{x_1,\dots,x_k\}$ is coupled with one of the labels $\{y_1,\dots,y_k\}$, being $y_j \in \{-1,+1\}$, $j = 1,\dots,k$, so that the sets

$$A_k = \{x_j \,:\, y_j = +1,\ j \in \{1,\dots,k\}\}$$

$$B_k = \{x_j \,:\, y_j = -1,\ j \in \{1,\dots,k\}\}$$

  are possibly linearly separable, as by Definition 5.1, using the supervised SVM classification method in Section 5;

- second, using the hyperplane π_k^* which linearly separates the sets A_k and B_k (solving an SVM problem as in Section 5), the point x_k is labelled with $y_k \in \{-1,+1\}$. Then, in case $y_k = +1$ we update $A_{k+1} = A_k \cup \{x_k\}$ and $B_{k+1} = B_k$, otherwise we set $A_{k+1} = A_k$ and $B_{k+1} = B_k \cup \{x_k)\}$.

Our proposal is summarized in the Algorithm SEP of Table 3, whose steps are briefly commented as follows, with reference to the flow-chart in Figure 6. After a preliminary initialization, where we set $\chi = \{x_i\}$ and define A_0 and B_0, we compute the *best* separating hyperplane π_0^* between A_0 and B_0, whose parameters are given by $\beta^{(0)}$ and $\beta_0^{(0)}$, using the SVM-based procedure in Section 5. In particular, in Algorithm SEP we indicate with $SVM(A_k,B_k)$ the solution of the problem (11) (or the Wolfe dual formulation (15)) in order to provide the best separating hyperplane π_k^* between A_k and B_k. Note that (15) has a bounded feasible set, so that it always admits solution.

Algorithm SEP

Data: $\chi \equiv \{x_i\} \subset \mathbb{R}^p$
Initialization: Set $k \leftarrow 0,\ choice \in \{TRUE, FALSE\},\ A_0 \leftarrow L_{\max},\ B_0 \leftarrow L_{\min}$
Step k: While $(|\chi \setminus \{A_k \cup B_k\}| \neq 0)$
Compute the parameters $[\beta^{(k)}, \beta_0^{(k)}] = SVM(A_k, B_k)$
Set $d_{\max}^{(k)} \leftarrow 0$
For $i = 1 : |\chi \setminus \{A_k \cup B_k\}|$
Extract x_i from $\chi \setminus \{A_k \cup B_k\}$
Compute the distance $d_i = d[x_i, H(\beta^{(k)}, \beta_0^{(k)}; x)]$
If $\left(d_i \geq d_{\max}^{(k)}\right)$ **Then**
Set $d_{\max}^{(k)} \leftarrow d_i$
Set $x_{\max}^{(k)} \leftarrow x_i$
End If
End For
If $\left(d_{\max}^{(k)} > 0\right)$ **Then**
If $\left(H(\beta^{(k)}, \beta_0^{(k)}; x_{\max}^{(k)}) > 0\right)$ **Then**
Set $A_{k+1} \leftarrow A_k \cup \{x_{\max}^{(k)}\}$
Set $B_{k+1} \leftarrow B_k$
Else (i.e. $H(\beta^{(k)}, \beta_0^{(k)}; x_{\max}^{(k)}) < 0$)
Set $A_{k+1} \leftarrow A_k$
Set $B_{k+1} \leftarrow B_k \cup \{x_{\max}^{(k)}\}$
End If
Else (i.e. $d_{\max}^{(k)} = 0$)
If $(choice = FALSE)$ **Then**
Set $A_{k+1} \leftarrow A_k \cup \{x_{\max}^{(k)}\}$
Set $B_{k+1} \leftarrow B_k$
Else
Set $A_{k+1} \leftarrow A_k$
Set $B_{k+1} \leftarrow B_k \cup \{x_{\max}^{(k)}\}$
End If
End If
$k \leftarrow k + 1$
End While

Table 3: Description of our SVM-based procedure applied to the points $\{x_i\}$ in the set χ.

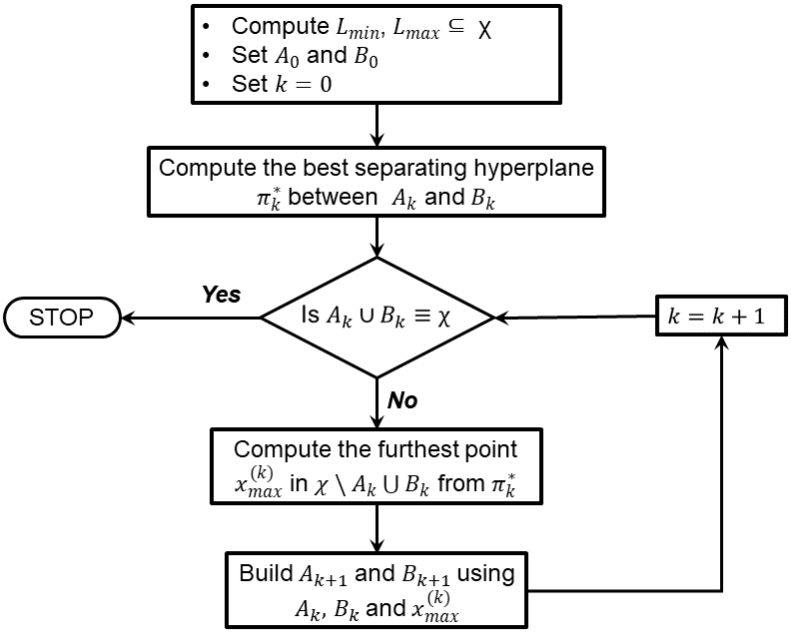

Figure 6: The flowchart of the Algorithm SEP in Table 3.

Therefore, at any step k we first pick the point $x_{\max}^{(k)}$ in $\chi \setminus \{A_k \cup B_k\}$ with the largest distance from π_k^*. Moreover, depending on the half space where $x_{\max}^{(k)}$ is located with respect to the hyperplane π_k^*, we update the novel pair A_{k+1}, B_{k+1} starting from the pair A_k, B_k. In the end, we increase the step and iterate the procedure.

Lemma 5.1 *Consider the set $\chi \subset \mathbb{R}^p$, with $|\chi| < +\infty$. Let $L_{\max}, L_{\min} \subseteq \chi$. Then, the Algorithm SEP in Table 3 provides the pair of sets A_m, B_m after m steps, with*

$$m = |\chi \setminus \{L_{\max} \cup L_{\min}\}|,$$

such that

$$\begin{cases} A_m \cup B_m = \chi \\ \\ A_m \cap B_m = \emptyset. \end{cases}$$

Proof: By Table 3, recalling that $|\chi|$ is finite, the index k ranges from 0 to $|\chi \setminus \{A_0 \cup B_0\}| \leq |\chi| < +\infty$. Moreover, since by construction (see also (1)–(iv)) $|A_k \cup B_k| = |A_{k-1} \cup B_{k-1}| + 1$, then m is exactly given by the number of points in χ which are neither present in $L_{\max}$ nor in $L_{\min}$. □

Proposition 5.2 *Let be given the nonempty sets $L_{\max}, L_{\min} \subseteq \chi$, and consider the Algorithm SEP in Table 3. If $L_{\max}$ and $L_{\min}$ are linearly separable, as by Definition 5.1, then the sets A_k and B_k are linearly separable, for any $k \geq 0$.*

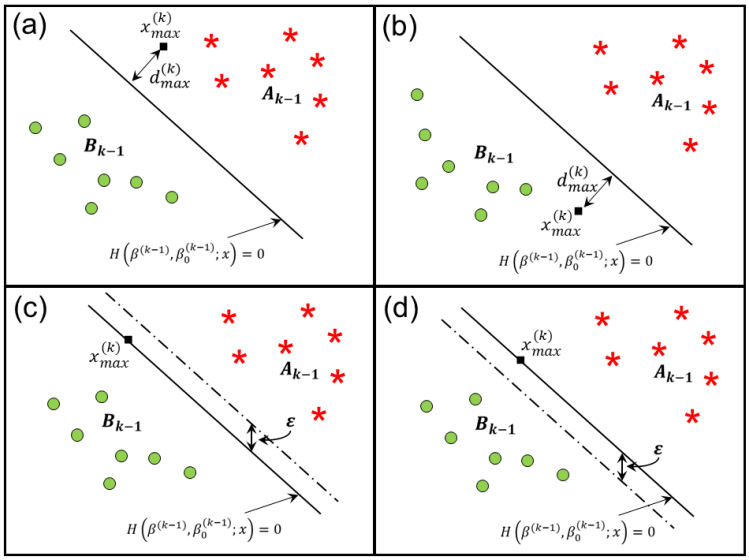

Figure 7: The possible cases for the computation of the sets A_k and B_k in Algorithm SEP.

Proof: We prove the result by induction. When $k=0$ then $A_0 \equiv L_{\max}$ and $B_0 \equiv L_{\min}$, so that by the hypotheses they are linearly separable. We assume that A_{k-1} and B_{k-1} are linearly separable and prove the result for A_k and B_k. By the instructions of Algorithm SEP, the sets A_k, B_k are generated in only three possible ways:

1. $d_{\max}^{(k)} > 0$ and $H(\beta^{(k-1)},\beta_0^{(k-1)};x_{\max}^{(k)}) > 0$: then (see Figure 7-(a)) we have $A_k = A_{k-1} \cup \{x_{\max}^{(k)}\}$ and $B_k = B_{k-1}$. Since A_{k-1} and B_{k-1} are linearly separable by the hyperplane
$$H(\beta^{(k-1)},\beta_0^{(k-1)};x) = 0$$
and we have $d_{\max}^{(k)} > 0$, then the hyperplane $H(\beta^{(k-1)},\beta_0^{(k-1)};x) = 0$ makes A_k and B_k linearly separable, too;

2. $d_{\max}^{(k)} > 0$ and $H(\beta^{(k-1)},\beta_0^{(k-1)};x_{\max}^{(k)}) < 0$: then (see Figure 7-(b)) we have similarly $A_k = A_{k-1}$ and $B_k = B_{k-1} \cup \{x_{\max}^{(k)}\}$. Therefore again the hyperplane $H(\beta^{(k-1)},\beta_0^{(k-1)};x) = 0$ makes A_k and B_k linearly separable;

3. $d_{\max}^{(k)} = 0$: then either *choice*$= TRUE$, implying $A_k = A_{k-1}$ and $B_k = B_{k-1} \cup \{x_{\max}^{(k)}\}$ (see Figure 7-(c)), or *choice*$= FALSE$, implying $A_k = A_{k-1} \cup \{x_{\max}^{(k)}\}$ and $B_k = B_{k-1}$ (see Figure 7-(d)). By induction A_{k-1} and B_{k-1} are linearly separable through the hyperplane $H(\beta^{(k-1)},\beta_0^{(k-1)};x) = 0$, i.e. the strict inequalities (4) are fulfilled for all the points in $A_{k-1} \cup B_{k-1}$. Thus, by Definition 5.1 there exists a sufficiently small value $\varepsilon > 0$ such that one of the hyperplanes
$$H(\beta^{(k-1)},\beta_0^{(k-1)};x) + \varepsilon = 0, \qquad H(\beta^{(k-1)},\beta_0^{(k-1)};x) - \varepsilon = 0$$
linearly separates A_k and B_k; in addition, one of the following relations holds
$$H(\beta^{(k-1)},\beta_0^{(k-1)};x_{\max}^{(k)}) + \varepsilon > 0, \qquad H(\beta^{(k-1)},\beta_0^{(k-1)};x_{\max}^{(k)}) - \varepsilon < 0.$$

□

Lemma 5.2 *Let be given the nonempty sets $L_{\max}, L_{\min} \subseteq \chi$, and consider the Algorithm SEP in Table 3. If $L_{\max}$ and $L_{\min}$ are linearly separable, then*

- *for any $k \geq 1$ the margin $W^{(k)}$ of the SVM problem $SVM(A_k, B_k)$ satisfies*

$$W^{(k)} = \min\left\{W^{(k-1)}, 2d_{\max}^{(k)}\right\}; \tag{17}$$

- *the sequence $\left\{W^{(k)}\right\}$ is monotonically nonincreasing, with*

$$W^{(k)} \leq 2d_{\max}^{(j)}, \qquad j = 0, \ldots, k. \tag{18}$$

Moreover, assume that at step k of the Algorithm SEP we set $x_{\max}^{(k)} \leftarrow \hat{x}$ along with $d_{\max}^{(k)} \leftarrow \hat{d}$, being $\hat{d} = d[\hat{x}, H(\beta^{(k)}, \beta_0^{(k)}; x)]$ with

$$\hat{d} \notin \arg\max_{x_i \in \chi \setminus \{A_k \cup B_k\}} \left\{d[x_i, H(\beta^{(k)}, \beta_0^{(k)}; x)]\right\}.$$

Then, we have $W^{(k)} \leq 2\hat{d}$.

Proof: Recalling the definition of *margin* for the $SVM(A_k, B_k)$ problem, it suffices to observe that at step k the Algorithm SEP computes $d_{\max}^{(k)}$ so that

$$d_{\max}^{(k)} \in \arg\max_{i \in \chi \setminus \{A_k, B_k\}} \left\{d[x_i, H(\beta^{(k)}, \beta_0^{(k)}; x)]\right\}.$$

Thus, from Proposition 5.2 the sets A_k and B_k are linearly separable so that by Proposition 5.1 we have the next possible cases:

- if $x_{\max}^{(k)} \in int\,(conv(A_k)) \cup int\,(conv(B_k))$ then the solution of $SVM(A_k, B_k)$ is not affected by $x_{\max}^{(k)}$ (since $x_{\max}^{(k)}$ cannot be a support vector), so that

  $$W^{(k)} = W^{(k-1)},$$

  with $W^{(k)} < 2d_{\max}^{(k)}$;

- if $x_{\max}^{(k)} \in \partial\,(conv(A_k)) \cup \partial\,(conv(B_k))$ (i.e. $x_{\max}^{(k)}$ is a support vector and does not belong to $L_{\max} \cup L_{\min}$) or $x_{\max}^{(k)} \notin conv(A_k) \cup conv(B_k)$ then

  $$W^{(k)} = 2d_{\max}^{(k)},$$

  being $W^{(k)} < W^{(k-1)}$.

Furthermore, the monotonicity of $\left\{W^{(k)}\right\}$ follows straightforwardly from formula (17), which also iteratively yields

$$\begin{aligned} W^{(k)} &= \min\left\{W^{(k-1)}, 2d_{\max}^{(k)}\right\} = \min\left\{W^{(k-2)}, 2d_{\max}^{(k-1)}, 2d_{\max}^{(k)}\right\} \\ &= \cdots = \min\left\{W^{(0)}, 2d_{\max}^{(0)}, \ldots, 2d_{\max}^{(k)}\right\}. \end{aligned}$$

This last result reveals that $W^{(k)} \leq 2d_{\max}^{(j)}$, for any $j = 0, \ldots, k$. Finally, relation (18) along with the definition of $\hat{d}$ immediately yield relation $W^{(k)} \leq 2\hat{d}$. □

The Lemma 5.2 reveals the reason for the choice to select, at step k of Algorithm SEP, the point $x_{\max}^{(k)}$. It corresponds to select the point with the largest possible distance from the current separating hyperplane π_k^*

$$H(\beta^{(k)}, \beta_0^{(k)}; x) = 0.$$

Thus, in this way we are also guaranteed that the next separating hyperplane $H(\beta^{(k+1)}, \beta_0^{(k+1)}; x) = 0$ yields the largest possible margin.

Proposition 5.3 *Let be given the nonempty sets $L_{\max}, L_{\min} \subseteq \chi$, and consider the Algorithm SEP in Table 3. Then, for any $k \geq 0$ and setting $C > 0$ in (15) sufficiently large, the misclassified points when solving $SVM(A_{k+1}, B_{k+1})$ are a subset of the misclassified points when solving $SVM(A_k, B_k)$.*

Proof: In case $k = 0$ and A_0, B_0 are linearly separable, then Proposition 5.2 immediately yields the result. Conversely, suppose at step $k \geq 0$ the sets A_k and B_k are not linearly separable when solving $SVM(A_k, B_k)$. Let $x_{\max}^{(k)}$ be the point selected at step k by the Algorithm SEP, and let the subsets $A'_k \subseteq A_k$ and $B'_k \subseteq B_k$ contain only the correctly classified points when solving $SVM(A_k, B_k)$. Then, since the constant $C > 0$ is sufficiently large, following the guidelines of the proof of Proposition 5.2 we obtain that all the points in $A'_k \cup B'_k \cup \{x_{\max}^{(k)}\}$ will be correctly classified when solving $SVM(A_{k+1}, B_{k+1})$. Thus, the misclassified points when solving $SVM(A_{k+1}, B_{k+1})$ are a subsets of those computed by $SVM(A_k, B_k)$. □

5.2 An alternative viewpoint with respect to Algorithm SEP

Following a completely different perspective let us consider again the problem (11). We recall that it represents the primal formulation associated to a supervised learning problem where all the labels $y_1, \ldots, y_N$ are used. Thus, in (11) β, β_0 along with the slack variables $\xi_1, \ldots, \xi_N$ are unknowns to be determined. On the other hand, as detailed in Section 5.1, in the iterative procedure within Algorithm SEP at step k we solve a binary classification problem, where *only* $|A_k \cup B_k| + 1$ points are involved, with $|A_k \cup B_k| + 1 < |\chi|$, when $k = 0, \ldots, |\chi| - |A_0 \cup B_0| - 1$, and $|A_k \cup B_k| = |\chi|$ when $k = |\chi| - |A_0 \cup B_0|$. Thus, we might alternatively consider to replace the whole iterative procedure in Table 3 with a suitable reformulation of (11). In this regard, let's first rewrite (11) as ($k \in \{0, \ldots, |\chi| - |A_0 \cup B_0|\}$)

$$\begin{aligned} \min_{\beta,\, \beta_0,\, \xi} \quad & \frac{1}{2}\|\beta\|^2 + C\sum_{i=1}^{N_k} \xi_i \\ \text{s.t.} \quad & (\beta^T x_i + \beta_0) y_i \geq 1 - \xi_i, \qquad i = 1, \ldots, N_k, \\ & \xi_i \geq 0, \qquad i = 1, \ldots, N_k, \end{aligned} \tag{19}$$

being $N_k = |A_k \cup B_k| + 1$. Then, we can alternatively replace the entire procedure in Table 3 by the mixed-integer quadratic programming problem ($M \gg 1$ large enough)

$$\begin{array}{ll} \min\limits_{\beta,\, \beta_0,\, \xi,\, \gamma} & \dfrac{1}{2}\|\beta\|^2 + C\sum\limits_{i=1}^{N}\xi_i \\ \text{s.t.} & -(\beta^T x_i + \beta_0) + 1 - \xi_i \leq (1-\gamma_i)M, \qquad i = 1,\ldots,N, \\ & (\beta^T x_i + \beta_0) + 1 - \xi_i \leq \gamma_i M, \qquad i = 1,\ldots,N, \\ & \xi_i \geq 0, \qquad i = 1,\ldots,N, \\ & \gamma_i \in \{0,1\}, \qquad i = 1,\ldots,N. \end{array} \tag{20}$$

Note that (20) is a mixed-integer quadratic programming problem such that:

- it encompasses $2N$ linear constraints ((19) only includes N_k linear constraints, with $N_k \leq N$), N nonnegativity constraints and N integrality constraints;
- formally it cannot be reformulated using the Wolfe dual approach (as we did to reformulate (11) into (15)) because the integrality constraints in principle spoil the convexity property of the problem;
- the $2N$ linear constraints represent disjunctive constraints, so that if $(\beta^*, \beta_0^*, \xi^*, \gamma^*)$ is the final solution:
  - in case $\gamma_i^* = 0$ then it means $x_i \in B_N$ (and in (19) we will equivalently have $y_i = -1$),
  - in case $\gamma_i^* = 1$ then it means $x_i \in A_N$ (and in (19) we will equivalently have $y_i = +1$);
- it is now worth investigating the relation between the solution by solving directly (20) and the one by solving the procedure in Algorithm SEP. A very preliminary numerical experience, where we used the solvers `Knitro 12.3` and `BARON` available on NEOS Server [22], showed that the solution of (20) and the one obtained following the procedure in Algorithm SEP coincide (i.e. eventually the sets A_N and B_N in Algorithm SEP will contain the same points predicted by solving (20), with an equal value for the quantities β and β_0).

Anyway, a further investigation is definitely mandatory, before any conclusion is carried out, both from a theoretical and numerical perspective. In particular, the chance to possibly extend and generalize the theory of the Wolfe dual reformulation for (20) would be of great interest, as well as a specific analysis for the linearly separable case (i.e. when $\xi = 0$). Furthermore, (20) is also subject to an additional generalization, in case a subset of the N points is imposed to belong to one of the two sets which result from the final binary classification. Indeed, maintaining the same taxonomy adopted in Table 3, and considering the sets $A_0, B_0 \subseteq \chi$, with $|A_0 \cup B_0| \leq |\chi|$, the formulation (20) yields

$$\begin{array}{ll} \min\limits_{\beta,\, \beta_0,\, \xi,\, \gamma} & \dfrac{1}{2}\|\beta\|^2 + C\sum\limits_{i=1}^{N}\xi_i \\ \text{s.t.} & -(\beta^T x_i + \beta_0) + 1 - \xi_i \leq (1-\gamma_i)M, \qquad i \,:\, x_i \notin A_0 \cup B_0, \\ & (\beta^T x_i + \beta_0) + 1 - \xi_i \leq \gamma_i M, \qquad i \,:\, x_i \notin A_0 \cup B_0, \\ & -(\beta^T x_i + \beta_0) + 1 - \xi_i \leq 0, \qquad i \,:\, x_i \in A_0, \\ & (\beta^T x_i + \beta_0) + 1 - \xi_i \leq 0, \qquad i \,:\, x_i \in B_0, \\ & \xi_i \geq 0, \qquad i = 1,\ldots,N, \\ & \gamma_i \in \{0,1\}, \qquad i \,:\, x_i \notin A_0 \cup B_0, \end{array} \tag{21}$$

where equivalently $\gamma_i = 1$ for $x_i \in A_0$ and $\gamma_i = 0$ for $x_i \in B_0$.

As a preliminary theoretical analysis which allows to compare the contents in Section 5.1 and Section 5.2, we include the next results. Let us introduce the set $\mathcal{N}_k$ such that

$$\begin{cases} \mathcal{N}_k \subseteq \{1,\dots,N\}, \\ |\mathcal{N}_k| = |A_0 \cup B_0| + k, \\ \forall x_j \in A_0 \cup B_0 \implies j \in \mathcal{N}_k, \end{cases}$$

so that $\mathcal{N}_k$ represents any subset of $\{1,\dots,N\}$ containing $|A_0 \cup B_0| + k$ indices, including those indices corresponding to points in A_0 and B_0. Moreover, starting from (21) we define the problem

$$\begin{array}{lll} \min\limits_{\beta,\, \beta_0,\, \xi,\, \gamma} & \dfrac{1}{2}\|\beta\|^2 + C\sum\limits_{i\in\mathcal{N}_k} \xi_i & \\ \text{s.t.} & -(\beta^T x_i + \beta_0) + 1 - \xi_i \le (1-\gamma_i)M, & \forall i \in \mathcal{N}_k,\ x_i \notin A_0 \cup B_0, \\ & (\beta^T x_i + \beta_0) + 1 - \xi_i \le \gamma_i M, & \forall i \in \mathcal{N}_k,\ x_i \notin A_0 \cup B_0, \\ & -(\beta^T x_i + \beta_0) + 1 - \xi_i \le 0, & \forall i \in \mathcal{N}_k,\ x_i \in A_0, \\ & (\beta^T x_i + \beta_0) + 1 - \xi_i \le 0, & \forall i \in \mathcal{N}_k,\ x_i \in B_0, \\ & \xi_i \ge 0, & \forall i \in \mathcal{N}_k, \\ & \gamma_i \in \{0,1\}, & \forall i \in \mathcal{N}_k,\ x_i \notin A_0 \cup B_0. \end{array} \tag{22}$$

Proposition 5.4 *Let be given the nonempty sets $\mathcal{L}_{\max}, \mathcal{L}_{\min} \subseteq \chi$, and consider the Algorithm SEP in Table 3. Let $\mathcal{L}_{\max}$ and $\mathcal{L}_{\min}$ be linearly separable. Then, the solution $(\beta^{(k)}, \beta_0^{(k)})$ of the subproblem $SVM(A_k, B_k)$ in Algorithm SEP is such that $(\beta^{(k)}, \bar{\beta}_0^{(k)}, 0, \bar{\gamma}^{(k)})$ is a solution of (22), for some $\bar{\beta}_0^{(k)} \in \mathbb{R}$ and $\bar{\gamma}^{(k)} \in \{0,1\}^{k-|A_0\cup B_0|}$.*

Proof: First observe that by Proposition 5.2, for any $k \ge 0$ the solution $(\beta^{(k)}, \beta_0^{(k)})$ of $SVM(A_k, B_k)$ yields $\xi_i = 0$ in (22), for any $i \in \mathcal{N}_k$. Moreover, for any $k \ge 0$ we have $1/2\|\beta^{(k)}\|^2 \ge 1/2\|\bar{\beta}^{(k)}\|^2$, being $\bar{\beta}^{(k)}$ a subvector in the solution of (22).

Hereafter, without loss of generality we assume $\chi \ne A_0 \cup B_0$ and the proof is carried on by complete induction. We first prove the result for $k = 0$. Assume by contradiction that the solution $(\beta^{(0)}, \beta_0^{(0)})$ of $SVM(A_0, B_0)$ is such that $(\beta^{(0)}, \bar{\beta}_0^{(0)}, 0, \bar{\gamma}^{(0)})$ is not a solution of (22), for any choice of $\bar{\beta}_0^{(0)} \in \mathbb{R}$ and $\bar{\gamma}^{(0)} \in \{0,1\}$. This implies that since (22) always admits solution, there exist parameters $\bar{\beta}^{(0)} \ne \beta^{(0)}$ and $\bar{\beta}_0^{(0)}$ such that the point $(\bar{\beta}^{(0)}, \bar{\beta}_0^{(0)}, 0, \bar{\gamma}^{(0)})$, for some $\bar{\beta}_0^{(0)} \in \mathbb{R}$ and $\bar{\gamma}^{(0)} \in \{0,1\}$, solves (22), with $1/2\|\bar{\beta}^{(0)}\|^2 < 1/2\|\beta^{(0)}\|^2$. This last inequality follows observing that $\bar{\beta}^{(0)} \ne \beta^{(0)}$ and the feasible set of (22) includes the one of the subproblem $SVM(A_0, B_0)$. However, this is not possible since $\beta^{(0)}$ and $\beta_0^{(0)}$ represent indeed the parameters of the best separation hyperplane between the sets A_0 and B_0, i.e. it should be $1/2\|\beta^{(0)}\|^2 \le 1/2\|\beta\|^2$ for any feasible subvector β.

Now, we assume the result holds for $k-1$ and we prove it for the step k. On this purpose, let $(\beta^{(k)}, \beta_0^{(k)})$ be the solution of $SVM(A_k, B_k)$ at step k of Algorithm SEP, and assume by contradiction that $(\beta^{(k)}, \bar{\beta}_0^{(k)}, 0, \bar{\gamma}^{(k)})$, for any $\bar{\beta}_0^{(k)} \in \mathbb{R}$ and $\bar{\gamma}^{(k)} \in \{0,1\}^{k-|A_0\cup B_0|}$, does not solve (22). Thus, since (22) always admits solution, there exists a point $\hat{x} \in \chi \setminus A_{k-1} \cup B_{k-1}$ and there exist parameters $\bar{\beta}^{(k)}$, $\bar{\beta}_0^{(k)}$ and $\bar{\gamma}_0^{(k)}$ such that:

(i) $(\bar{\beta}^{(k)}, \bar{\beta}_0^{(k)}, 0, \bar{\gamma}^{(k)})$ solves (22), with $\bar{\beta}^{(k)} \neq \beta^{(k)}$ and $\bar{\gamma}^{(k)} \in \{0,1\}^{k-|A_0 \cup B_0|}$;

(ii) the hyperplane $\bar{\pi}_k = \left\{ x \in \mathbb{R}^p \ : \ [\bar{\beta}^{(k)}]^T x + \bar{\beta}_0^{(k)} = 0 \right\}$ linearly separates either the sets

$$\begin{cases} A_{k-1} \cup \{\bar{x}\} \\ B_{k-1} \end{cases} \tag{23}$$

or the sets

$$\begin{cases} A_{k-1} \\ B_{k-1} \cup \{\bar{x}\}; \end{cases} \tag{24}$$

(iii) we have $1/2\|\bar{\beta}^{(k)}\|^2 < 1/2\|\beta^{(k)}\|^2$;

(iv) $\hat{x} \neq x_{\max}^{(k)}$ because otherwise we would have $\bar{\beta}^{(k)} = \beta^{(k)}$ (see Algorithm SEP).

By (iv) only the four geometric scenarios in Figure 8 are allowed, depending on case (23) (Figure 8–*left*) or case (24) (Figure 8–*right*). In all the four cases, recalling the definition of $x_{\max}^{(k)}$ we must

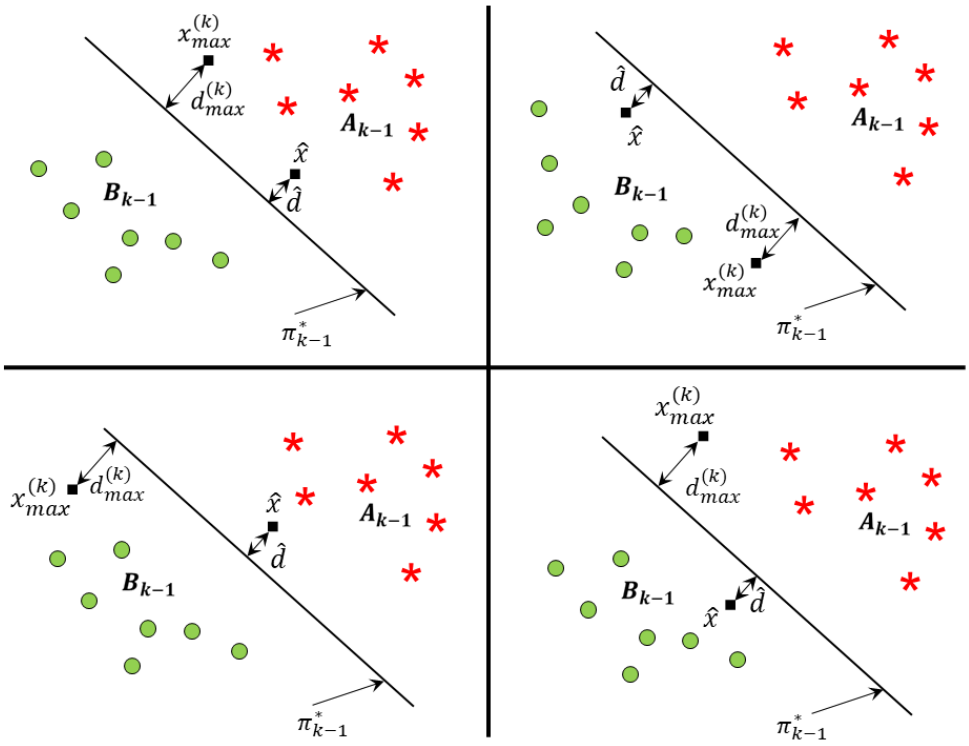

Figure 8: Four scenarios comparing the solution of the subproblem $SVM(A_{k-1}, B_{k-1})$ in Algorithm SEP and the solution either yielded by (23) or (24).

have $d^k_{\max} > \hat{d}$. Thus, by Lemma 5.2 the margin $W^{(k)}$ obtained solving $SVM(A_k, B_k)$ and the margin $\bar{W}^{(k)}$ obtained when solving (22) satisfy $W^{(k)} \leq \bar{W}^{(k)}$, which contradicts (iii). □

6 Applications: our classification and measurement problems

In this section we analyze three application problems where our ML approach was successfully adopted. We deliberately decided to include in our numerical experience test problems from three completely different contexts, in order to give evidence about the versatility of our ML perspective, even in those research areas where (frequently) other perspectives are preferred. We urge to remark that, in all the following three case studies, our proposal is essentially stepped applying the next phases:

- we reformulate the problem as a sequence of mathematical programming models, where in each model two linearly separable sets of points are clearly identified;
- we use a ML technique based on SVMs, for the solution of each mathematical programming model, adopting the guidelines detailed in Section 5;
- we tailor our procedure on the specific features of each case study.

Moreover, for the reader's convenience we maintain the paradigm *fellows/subjects – unit* adopted in Section 4, in order to describe with a similar taxonomy each of the next three case studies. The taxonomy refers to the sorting, ranking and classification of the subjects' performances. We recall the three notions of sorting, ranking and classifications are defined as follows:

Sorting is the rearrangement of numbers (or other orderable objects) in a list into their correct lexicographic order. Sorting is any process of arranging items in some sequence and/or in different sets, ordering: arranging items of the same kind, level, degree in some ordered sequence.

Ranking is the relationship between two mathematical values where each value can be less than, greater than, or equal to the second value. A ranking is a relationship between a set of items such that, for any two items, the first is either 'ranked higher than', 'ranked lower than' or 'ranked equal to' the second. In mathematics, this is known as a weak order or total preorder of objects. It is not necessarily a total order of objects because two different objects can have the same ranking. The rankings themselves may be totally ordered.

Classification is the process of dividing the data space or data points into a number of groups, such that data points in the same groups are more similar to other data points in the same group, and dissimilar to the data points in other groups. The major objective of classification is to find patterns (i.e. similarities within data points) in a labelled dataset.

Each of them can be exemplified referring to the procedure of performance appraisal system in human resources practices. Sorting means to define a mere classification of the performances and of the performers. Ranking is an ordered sequence from the lower to the higher level, and in human resources management practices we observe different rankings from high to low performers. Classification means using patterns to group the performers. Often the performance management system follows a linear sequence: sorting, ranking and classification.

As regards the computational resources used for the elaborations in the current section, we adopted the following tools:

- as regards the hardware, an `Intel`© Core(TM) i7-6700HQ CPU 2.60GHz desktop with 32GB RAM, 1Tb SSD, under `Microsoft`© Windows 10 Pro OS was used;
- as coding environment, `Matlab`© release 2014b [18] was used, for elaborations in all the three case studies, without any specific Toolbox for parallel computing. As regards the SVM training, the *Statistics and Machine Learning Toolbox* was adopted, and in all the SVM-based sub-problems the Lagrange–Wolfe dual formulation (15) was solved. No misclassification was experienced. In addition, the Matlab classifier `fitcsvm()` was adopted, specifying no kernel matrix (i.e. we always solved linearly separable subproblems) and `SMO` (Sequential Minimal Optimization) which implements the algorithm in [17] to solve the dual problem (15). All the other default parameters in `fitcsvm()` were used, apart from `BoxConstraint` which was set to `Inf`;
- the mathematical programming language `AMPL`© [19], coupled with the solvers for Mixed Integer Linear Programming available on NEOS Server [22], including `CPLEX` and `feaspump`, was used for the case study -3-;
- `Excel` from the suite `Microsoft Office 365`© was used for merely polishing and preparing (no pre-processing) the database associated with the case study -1-.

6.1 Case study -1-

We consider an application problem where the unit is represented by a group of operators (fellows), working in public currency exchange booths. These fellows process a number of transactions (currencies trades) where operators' personal discretion with customers is allowed, when charging the amount of the commission. Basically, the discretion is allowed for any of the fellows, and depends on a number of attributes associated with the transaction: total amount of the transaction, system of payment (cash or card), time during the day, currency to be negotiated, etc. The main data associated with this problem is summarized in Table 4. Our ultimate goal is that of proposing

Number of operators	91
Overall number of transactions	257109
Time window for transactions	July 1st, 2017 – June 30th, 2019
Range for the amount of each transaction ($)	0.40 – 2999.60
Range for the amount of the commission ($)	0.00 – 538.13

Table 4: Data associated with the Case study -1-.

a measurement system for the subjects' performance, along with a rewarding system based on the finalized transactions. In particular, fellows who can boost larger transactions and possibly propose larger commissions are welcome, in the eyes of the company. Conversely, fellows who finalize small entity transactions with reduced commissions are possibly less eligible for rewarding. The current system adopted by the company for rewarding basically relies on first computing some *reference average performance*; then, the achievements of each fellow are compared with the last reference performance. Unfortunately, this approach possibly suffers for the following biases:

- each fellow is associated with a pair of values (average transaction amount and average commission amount), that inherently reveal the multicriterial nature of the problem. Hence, there is the difficulty to provide only one performance indicator which adequately considers the contribution of both the values;
- even in case a unique indicator were provided, in order to measure the performance of a fellow on a single transaction, it remains the necessity to aggregate the measures of each fellow's transactions;
- a comparison among all the fellows' performances is required, that is not a trivial task considering the multicriterial nature of the problem;
- time dependent scenarios are sought, when the *Time window for transactions* in Table 4 is suitably adapted (e.g. monthly or weekly reports might be considered);
- it is unnecessary (and it is not required) to address a rewarding strategy which is a continuous function of the performance of fellows. Indeed, the company suggested that *classes of rewarding* would be preferable, in the light of a simplification for the rewarding process.

All that said, we applied the iterative procedure in Table 3 to the Case study -1-, where we limited our analysis to the step $k = 0$ of Algorithm SEP. In particular, Figure 9 displays all the transactions (crosses and bullets) performed by the fellows, along with the elaborations obtained by using Algorithm SEP. From Figure 9 we immediately realize that:

- all the pairs of values associated with the transactions are confined within a cone with vertex in the origin;
- the pairs tend to follow patterns (half-lines) from the origin.

Moreover, the front $L_{\max}$ (see Table 3) is given by the large red bullets, while $L_{\min}$ is represented by the large cyan bullets (whose geometry plays a relevant role for the solution we propose)[4]. Finally, all the other transactions are identified by a blue cross, and circled bullets identify support vectors. After applying the step $k = 0$ of Algorithm SEP, the sets $A_0 = L_{\max}$ and $B_0 = L_{\min}$ prove to be linearly separable (see Section 5), so that the three lines in Figure 9 correspond to: the best separating hyperplane (central line) and those parallel hyperplanes through the support vectors. The overall area of the figure can be partitioned into four different regions of the plane, namely A, B, C, D, with the following meaning:

- **A**: region corresponding to *over performing transactions*;
- **B**: region corresponding to *UP performing transactions*, i.e. transactions which are rather appealing for the company, but are also less relevant with respect to the transactions in region **A**;
- **C**: region corresponding to *DOWN performing transactions*, i.e. those transactions which are identified by points immediately below the best separating hyperplane;

[4]For the sake of completeness observe that we obtained in this case study $|L_{\max}| = 9$ and $|L_{\min}| = 1675$, being the points in $L_{\min}$ evidently clustered nearby the origin.

- **D**: region corresponding to *under performing transactions*, i.e. all the remaining transactions which are not in the regions **A**, **B** and **C**.

As a final remark, the *time* may play an additional relevant role in our analysis. Indeed, consider replacing the picture in Figure 9 by a series of similar pictures, each of which is referred to transactions performed in a given time window. Then, we would be able to assess the possible invariance of each fellow performance with the time, which might be an additional information to steer the rewarding process. The overall scheme of performances associated with the operators (fellows) is

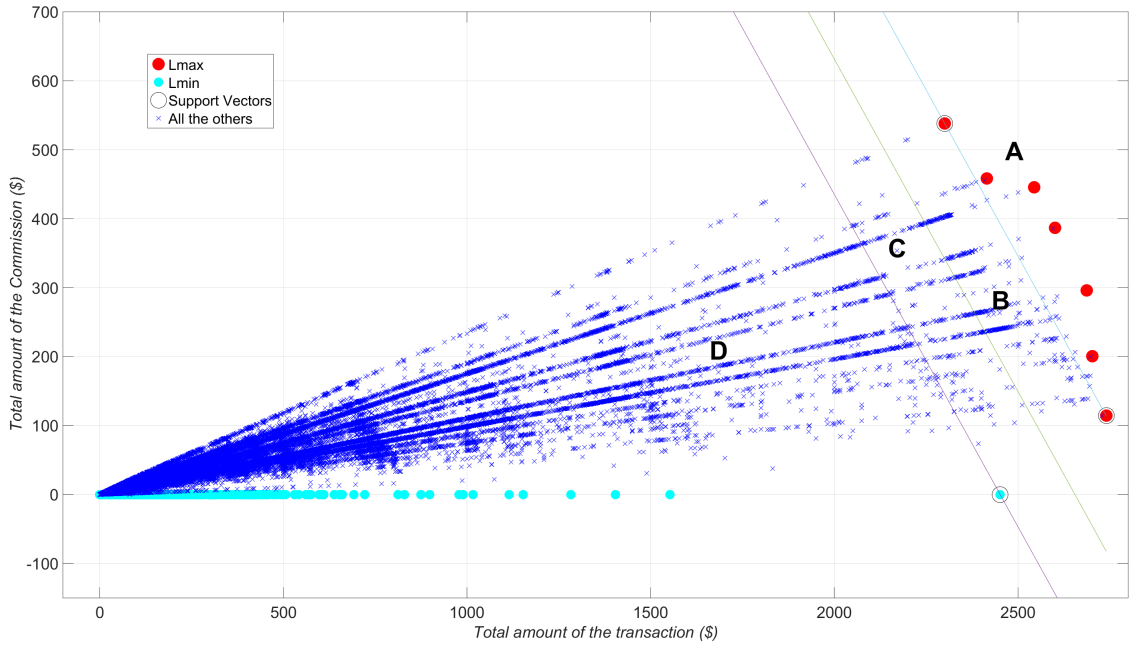

Figure 9: Our SVM-based procedure applied to the Case study -1-. Only one step (namely step $k = 0$) of Algorithm SEP is applied to compute the best separating hyperplane (central line separating the regions **B** and **C**).

reported in Table 5, where the number of transactions of each operator, in regions A, B, C and D respectively, is detailed. Not all the fellows are included in the last table, inasmuch as some of them were a priori not considered eligible for rewarding by the company. It might be interesting to remark that the best separating hyperplane in Figure 9 seems to well take into account the geometry of the points (transactions). This confirms that our SVM-based ML approach has been able to exploit the structure of the whole dataset (i.e. the transactions) used for training, as well as giving indications as regards

- *classification*: i.e. subdividing the performances into four distinct regions that all the fellows contributed to identify;

- *ranking*: i.e. the company can now determine priorities among fellows, based on the associated 4-entries vectors in Table 5. This result goes beyond the mere computation of the fronts $L_{\max}$ and $L_{\min}$, that simply encompass a reduced subset of the fellows;
- *sorting*: i.e. associating a 4-entries vector of performance to each fellow. This follows a classic paradigm of sorting for medalists in a championship, considering that the figures associated with the four regions A, B, C and D, respectively correspond to the number of gold, silver, bronze and 'wood' medals.

We urge to remark that the outcomes we obtained in Table 5 evidently recall the results from a pool of medalist nations, in a sport tournament (where gold, silver, bronze and 'wood' medals are awarded). The purpose and the conclusions of our analysis follow indeed a similar guideline: the regions A, B, C and D in Figure 9 are identified after considering *all* the performances of the fellows (as in a tournament), through a ML process. Moreover, these regions represent sufficiently general reference areas that company stakeholders can decide to consider, in view of their own soft/hard rewarding policies. Finally, the analysis suggested by Figure 9 can be further refined, observing that some points in $L_{\min}$ can be considered as *outliers* when solving the problem $SVM(A_0, B_0)$. Indeed, those cyan points, which refer to large transactions and very small commissions, can be considered of poor relevance within our procedure and possibly skipped by the company.

6.2 Case study -2-

We describe in this section an evaluation process for members of a given community. In order to maintain the privacy and a certain level of anonymity of the members, but also preserving the availability of an essential information, we follow the guidelines in Section 5 and address the community as the *unit*, being again its members the *fellows*. The unit includes 63 fellows who are subject to a joint evaluation process, in order to possibly introduce a reference methodology to spur the performance of both each fellow and the overall unit. Each fellow is subject to an evaluation with respect to a couple of orthogonal criteria, i.e. we associate the pair $x_i \in \mathbb{R}^2$, $i = 1, \ldots, 63$, to the i-th fellow, assuming that the entries $(x_i)_1$, $(x_i)_2$ are uncorrelated. In particular, two independent oracles O_1, O_2 exist, such that the values $\{(x_i)_1\}$ are obtained by the oracle O_1, while the values $\{(x_i)_2\}$ are obtained by the oracle O_2.

Figure 1 reports all the pairs $\{x_i\}$ associated with all the unit fellows, with respect to the two criteria C_1 and C_2. In particular we have from the two oracles the following range of values

$$\begin{cases} 2.5 \leq (x_i)_1 \leq 3.7 & i = 1, \ldots, 63 \\ \\ 10 \leq (x_i)_2 \leq 100 & i = 1, \ldots, 63. \end{cases}$$

To better describe our procedure for classification, ranking and sorting fellows in this unit, we make reference to the scheme in Table 3. In this case study it is $|L_{\max}| = 3$ and $|L_{\min}| = 3$, being $A_0 = L_{\max}$ (the large North-West red bullets in Figure 10) and $B_0 = L_{\min}$ (the South-West large cyan bullets in Figure 10). The three lines in the same figure represent the best separating hyperplane $\pi_0^* : H(\beta^*, \beta_0^*; x) = 0$ (center line) and the lines through the support vectors (circled large bullets), being the problem $SVM(A_0, B_0)$ evidently linearly separable. Observe that at step $k = 21$ the SVM-

Operator Number	(**A**)	(**B**)	(**C**)	(**D**)	Operator Number	(**A**)	(**B**)	(**C**)	(**D**)
1	0	0	0	147	2	0	1	3	1304
3	0	9	2	3465	4	0	0	0	348
5	2	4	15	13336	7	0	7	9	4308
9	0	26	9	5856	10	2	23	20	7804
12	0	14	6	6424	14	0	11	5	5661
15	0	3	2	2467	17	1	6	4	5426
18	0	29	10	7726	19	0	1	3	1361
20	1	0	0	35	21	0	0	1	377
22	0	0	0	605	23	0	0	1	1152
25	0	0	2	390	26	1	3	0	444
27	0	3	1	1967	28	0	4	0	1065
30	1	10	4	2676	31	0	1	1	372
33	0	2	1	679	34	0	0	0	184
35	2	1	4	3191	37	0	0	0	287
38	0	0	1	290	39	0	1	1	322
40	0	0	0	552	41	0	16	6	4470
42	0	0	0	254	43	0	12	1	4345
44	1	5	9	5386	45	2	6	4	10637
46	0	6	5	3820	47	0	9	8	4903
48	0	14	0	1222	49	0	0	0	73
50	0	0	1	89	51	0	0	0	133
52	0	0	1	98	53	1	1	1	674
54	0	12	1	1379	55	3	5	2	821
56	0	0	2	799	57	0	1	3	897
59	0	2	1	3476	61	0	4	1	2734
65	0	0	0	965	66	1	8	8	5687
68	3	19	19	7660	69	0	24	14	5953
70	0	12	13	11689	71	0	4	2	2731
72	0	1	1	2407	74	0	3	2	1644
75	0	1	0	726	76	0	12	5	5790
77	0	2	2	3647	78	0	9	5	4859
79	0	24	15	8345	80	0	27	16	6139
81	2	16	1	6698	83	1	7	8	4177
84	0	2	1	5657	85	0	1	1	337
86	1	40	17	7219	87	1	6	5	3499
88	2	0	1	1835	89	0	2	1	621
90	0	7	5	3672	91	0	2	2	4833
93	0	23	6	4954	94	1	2	0	1186
95	1	6	9	4183	96	0	10	5	3946
97	0	0	0	143	98	1	3	0	892
99	0	0	0	7	100	0	0	0	535
101	0	0	0	7	102	0	0	2	554
103	0	0	3	1499	104	0	0	0	56
105	0	0	0	49	106	0	3	1	1403
107	0	0	1	46	108	0	0	0	12
109	1	1	2	1847					

Table 5: Table of the performances for each operator (fellow) in the Case study -1-. Following the paradigm of a sport tournament, each operator is awarded with a given number of 'gold' medals (A), 'silver' medals (B), 'bronze' medals (C), and 'wood' medals (D).

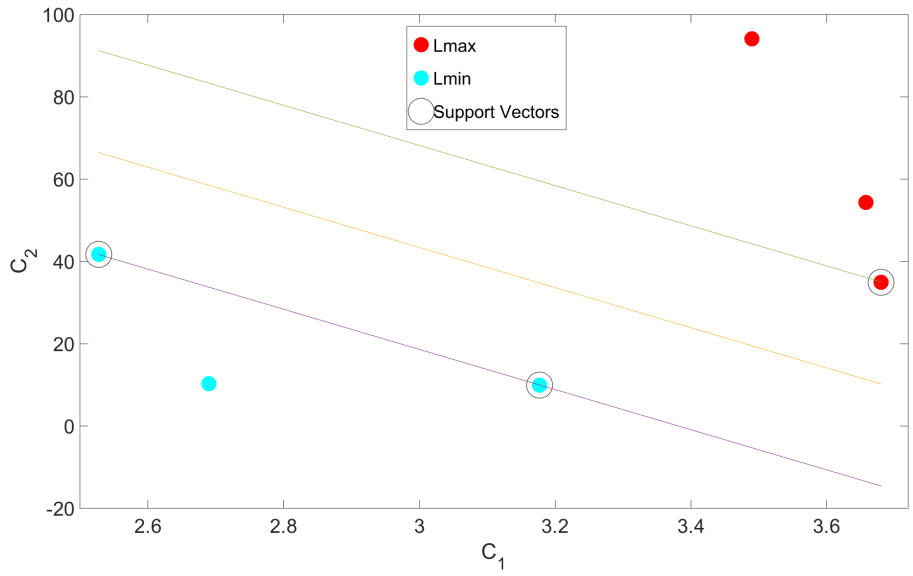

Figure 10: The sets A_0 (large North-East red bullets) and B_0 (large South-West cyan bullets) in Algorithm SEP, for the Case study -2-. The three lines represent the best separating hyperplane (center line) and the lines through the support vectors.

based procedure of Table 3 shows the situation in Figure 11, being A_{21} (respectively B_{21}) given by the small and large red (respectively cyan) points. The points with label $L_{\max} - ex$ (respectively $L_{\min} - ex$) represent those fellows whose performance, at the current step of Algorithm SEP, have been included in the set $L_{\max}$ (respectively $L_{\min}$), yielding *extensions* to $L_{\max}$ and $L_{\min}$. As a further analysis, in accordance with Algorithm SEP we also report in Figures 12 and 13 the outcome at step $k = 50$ and step $k = 63$ (final step, including all the fellows).

For comparative reasons, in Figure 13 a large black bullet appears in the centre, representing indeed a geometric *midpoint* of all the fellows. It equivalently represents an additional '*dummy fellow*', whose performance are computed as the algebraic mean of all the fellows performances. It can be helpful to evaluate the different strategy we adopt, in our ML process, to measure the fellows performance.

In particular, those fellows who dominate the dummy fellow (i.e. such that both their performances are preferable to those of the dummy fellow) must be represented by a point in the rectangle within the dashed lines. This equivalently suggests that, in accordance with a (sound) more standard analysis, the attributes associated with the criterion C_1 and C_2 should be separately considered, in order to assess the comparison among fellows. Conversely, in our framework, the performance of those fellows on a line parallel to the best separating hyperplane π^*, and through the large black bullet, will be considered equivalent to it. Thus, from a more standard viewpoint, which relies on the classic topology associated with $\mathbb{R}^p$, the concept of similarity for the performance of fellows is fulfilled for neighbours whose reciprocal Euclidean distance is relatively small. On the contrary, in our framework we take into account at once the criteria C_1 and C_2, introducing a metrics which

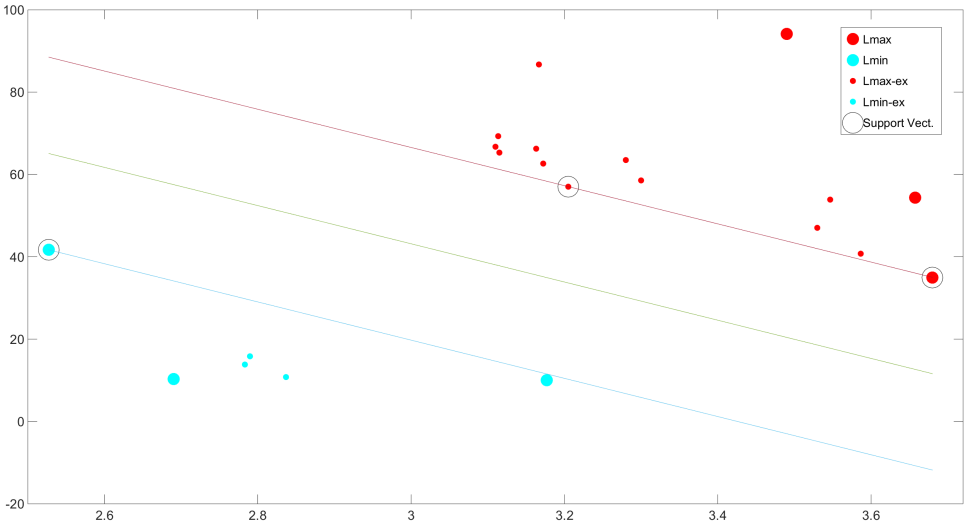

Figure 11: The sets A_{21} (large and small red bullets) and B_{21} (large and small cyan bullets) in Algorithm SEP, for the Case study -2-. The three lines again represent the best separating hyperplane (center line) and the parallel lines through the support vectors.

relies on all the fellows performance, so that the Euclidean distance among points associated with fellows might be irrelevant.

In this regard, our framework also allows to easily perform *simulations*, where a number of fictitious fellows are forced to join the unit. This might yield a relevant tool to evaluate the robustness of our approach, as well as to evaluate the impact of enrolling novel fellows in the unit. On this purpose, we generated 10% additional fellows (say 6 novel fellows) and assigned to them a performance within the best 10% of the best performance of the initial 63 unit fellows, with respect to both the criteria. This is equivalent to enroll an appealing subset of fellows (see the crosses in the upper right corner of Figure 14) in the unit. Figure 14 reports both the novel scenario and the results provided by Algorithm SEP, showing how the additional fellows affect our analysis. It can be seen that the additional fellows might not upset the performance of the overall unit (i.e. the initial 63 fellows plus the fictitious ones): which may possibly suggest further enrollment actions for the unit, in order to improve the performance.

In view of the last considerations, we immediately realize that as regards the capabilities of *classification*, *sorting* and *ranking* for our ML procedure, in this case study -2-, the next specific conclusions hold:

- *classification*: the best separating hyperplane π^* in Figure 13 and the two (support) hyperplanes through the support vectors induce a partition of $\mathbb{R}^2$ in four regions. The two regions above (respectively below) the upper (respectively lower) support hyperplane, and the two regions between the support hyperplanes separated by π^*. All the fellows belong to one (and only one) of these four regions;

- *ranking*: each fellow, whose representative point is above the best separating hyperplane π^*, is performing better with respect to any fellow below π^*. Similarly, any fellow *in-between* the two support hyperplanes has a performance *in-between* the performances of fellows above/below the support hyperplanes;

- *sorting*: the Euclidean distance between the point associated with a fellow and the best separating hyperplane π^* gives an indication, for possibly sorting the fellows. I.e. a partial ordering among the fellows can be induced by this Euclidean distance. Observe that it *does not represent a total ordering*. Indeed, intuitively speaking, it does not allow to define an ordering among those points lying on a line parallel to π^*. This last consideration perfectly matches with the comments at page 4.

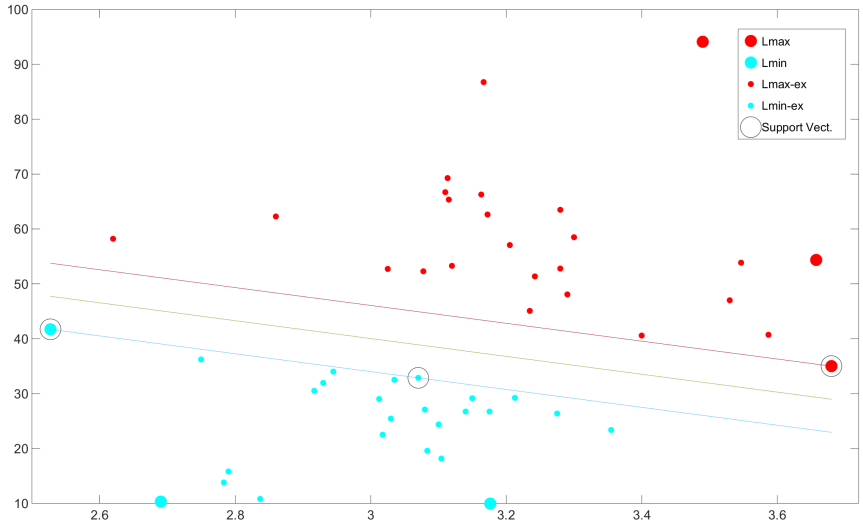

Figure 12: The sets A_{50} (large and small red bullets) and B_{50} (large and small cyan bullets) in Algorithm SEP, for the Case study -2-, including the best separating hyperplane (center line) and the lines corresponding to the support vectors.

On the other hand, for this case study some additional indicators can be defined, exploiting the structure of the problem, in the light of allowing both a thorough comparison among the fellows and the evaluation of the entire unit performance. These indicators can be summarized as follows (the next definitions can be easily extended in case more than two criteria were considered, too):

- *skewness*: for the i-th fellow it is indicated as $skew_i$ and represents the ratio between the performance on the criterion C_1 over the performance on the criterion C_2. Thus, the closer the skewness to one, the more we have a balanced performance for the i-th fellow. The overall skewness for the unit can be similarly defined as $\sum_{i=1}^{N} skew_i/N$, being N the number of fellows;

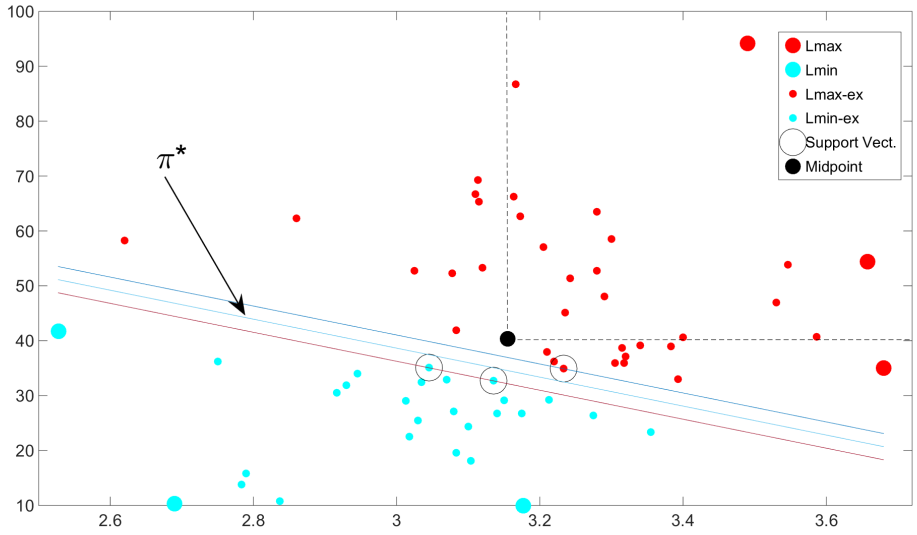

Figure 13: The sets A_{63} (large and small red bullets) and B_{63} (large and small cyan bullets) in Algorithm SEP, for the Case study -2-, including the best separating hyperplane (center line) and the lines corresponding to the support vectors. The large black bullet represents an 'average dummy fellow', whose performance are computed as the algebraic mean point of all the fellows' performances.

- *performance*: it is associated with the entire unit and is given by the distance of the origin from the best separating hyperplane π^*; thus, it is obtained as (see the footnote at page 10)

$$d[0,\pi^*] = d[0,H(\beta^*,\beta_0^*;x)] = \frac{|\beta_0^*|}{\|\beta^*\|}.$$

  This indicator has a twofold purpose. On one hand it allows to monitor the progress of the overall unit, in the case multiple scenarios associated with different time windows were considered (e.g. the performance of the unit in different years). On the other hand, it also allows a comparison among the performances associated with different units. In this regard, considering units with a similar *skewness*, the one with a larger *performance* should be awarded;

- *scenarios simulation*: we can easily generate additional (i.e. fictitious) fellows to the unit (as in Figure 14), in order to evaluate the performance of the enlarged unit and get strategies on modifying its structure, in the light of improving the performance of both the fellows and the overall unit.

6.3 Case study -3-

Here we consider a challenging price forecast problem, associated with a specific asset class, namely the crypto assets. In particular we focus on the most famous crypto asset which is *Bitcoin* [23], inasmuch as it currently corresponds also to the largest market capitalization among the crypto assets.

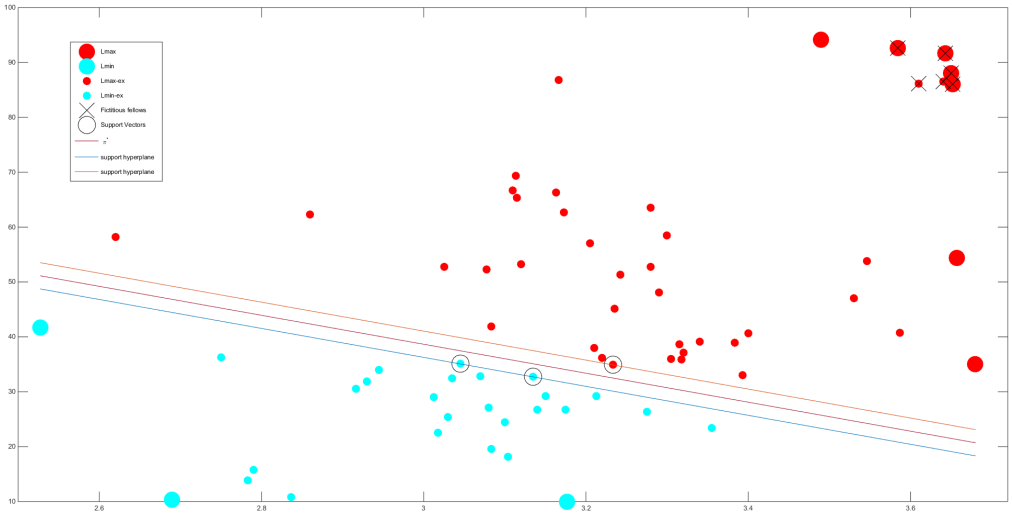

Figure 14: The set of fellows in the unit is integrated with 10% additional (fictitious) fellows, whose performance are within 10% of the best performance (with respect to both the criteria) of the initial fellows. The fictitious fellows are represented by crosses in the upper right corner.

Our main purpose here is showing that we can use our iterative ML approach from Section 5, in order to provide a long term measurement system for estimating Bitcoin price. We will show that a similar result, though less accurate and not completely well posed, is currently achieved through more standard tools which are based on linear regression and need some theoretical assumptions. Our perspective remarkably does not rely on those theoretical assumptions (e.g. the normal distribution of data).

Bitcoin was created in 2008 [23] by an anonymous researcher (or possibly a team of people), under the nickname of Satoshi Nakamoto. It represents a digital asset whose implementation release protocol is open-source. What strongly characterizes Bitcoin with respect to fiat currencies is its decentralized nature, since no private bank or national central bank is neither responsible for managing the overall amount of circulating Bitcoins nor be able to issue new Bitcoins. Bitcoin negotiations need exchanges to finalize transactions; however, peer-to-peer movements on the Bitcoin network can be perfectly completed without the need for intermediaries, too. Transactions among users are validated by network nodes, after solving complex inverse cryptographic problems. Moreover, the transactions cannot be removed from the Bitcoin network, since they are recorded in a public distributed ledger called *blockchain*. Novel Bitcoins are created by special nodes of the network called *miners*, as a reward for solving the above complex inverse cryptographic problems.

In order to foresee the long term price for Bitcoin, a number of different approaches were considered in the literature (the interested reviewer can refer to the recent papers [12, 29] and therein references). However, one of the main difficulties for its correct prediction relies on the high volatility of this asset, whose price can definitely show great oscillations in a short time period. The main

reason of this drawback is that Bitcoin is a relatively recent asset. Thus, considering its market capitalization (which is currently about one tenth of the gold capitalization), it is often the target of speculation with highly leveraged transactions. We strongly remark that the recent paper [12] uses a ML-based approach to provide a quantitative model for Bitcoin price forecast. However, [12] basically relies on using *intrinsic mode functions* (IMFs), coupled with SVMs, which attempt to capture the natural characteristics of the time series associated with Bitcoin prices. On the contrary, the analysis in [29] is based on the optimization method LASSO for ML. Conversely, we combine a preliminary multiobjective programming approach with an SVM, which represents to our knowledge a novel proposal in the literature.

All this said, it is possible to show that to some extent the price of Bitcoin can be plot in terms of its *stock-to-flow* ratio SF, which is defined as the ratio between the overall stock of Bitcoin on the market, and the quantity of Bitcoins minted in a given time period (say one year). Figure 15 represents 1329 price vs. SF pairs, corresponding to the period between March 9th, 2009 and January 21st, 2021[5]. Data has been transformed to allow easy processing, with respect to a logarithmic scale. The (red) bullets represent Bitcoin prices corresponding to SF values, and it is not difficult to realize that data is not normally distributed. Thus, the approximate solution of the linear regression problem to forecast Bitcoin prices, through the solution of a Linear Least Squares problem, may yield a misleading information to large extent.

To better explain the last issue, assume we are given the $1+N$ random variables Y and $\{X_i\}$, being

$$Y = \sum_{i=1}^{N} \alpha_i X_i + u_i, \qquad \beta_i \in \mathbb{R}, \quad i = 1, \dots, N,$$

where Y is the dependent variable and $\{X_i\}$ are the independent variables. Moreover, u_i represents a statistical error, for any i, and satisfies $\mathbb{E}(u_i|X_1, \dots, X_N) = 0$. Then, if $Y, X_1, \dots, X_N$ are *independent and identically distributed* (i.i.d.) and a few mild assumptions are fulfilled, the solution of the Linear Regression problem

$$\min_{a,b} \mathbb{E}\left[Y - \left(b + \sum_{i=1}^{N} a_i X_i \right) \right] \tag{25}$$

can be equivalently obtained by solving the Linear Least Squares problem

$$\min_{a,b} \left[\sum_{i=1}^{p} Y^{(j)} - \left(b + \sum_{i=1}^{N} a_i X_i^{(j)} \right) \right]^2, \tag{26}$$

where p represents the number of available samples for the random variables $(Y, X_1, \dots, X_N)$.
We strongly remark that the solution of the last minimization problem is definitely appealing, since it is an unconstrained convex quadratic one. However, we also highlight that the solutions of (25) and (26) might strongly differ in case the theoretical assumptions on the quantities $Y, X_1, \dots, X_N, u_1, \dots, u_N$ were not fulfilled. A typical example where we experience the last drawback is the case in which the samples do not follow a normal distribution. Indeed, in case the random

[5]Note that data in the early years of Bitcoin history is partly missing, because in 2009–2010 there was not yet a central observer for the accurate data collection.

variables $u_1, \ldots, u_N$ admit the joint normal distribution $N(0, \sigma^2 I)$, with zero expected value and the same variance for all the variables, then the solutions of (25) and (26) coincide.

Typically in applied sciences (26) is often solved assuming the fulfillment of indispensable theoretical assumptions, which are unfortunately often not satisfied. Thus, a test on the reliability of the solutions of (26) is usually sought (e.g. the R^2 test, where $R^2 \approx 1$ implies a fulfillment of the assumptions, while $R^2 \approx 0$ implies that solving (26) is definitely unreliable).

In our Case study -3- we are interested about estimating the price (i.e. Y) of Bitcoin vs. its SF (i.e. X), being $Y = aX + u$, with $a \in \mathbb{R}$, but the error u is not normally distributed, so that the solution of the Linear Least Squares problem (26) might possibly represent a poor estimator (see also the practical analysis on [20, 28]). Thus, we carried on a different analysis, with the aim of possibly providing a more reliable estimation for Bitcoin future price, solving convex subproblems with basically the same computational complexity of (26). In this regard, Figure 15 also reports two continuous lines (*support lines*) which delimit the *narrowest region* (stripe) containing all the red bullets (i.e. the sample pairs price–SF). Finally, the dashed line represents the symmetry axis of the last delimited region (stripe). More formally, the continuous lines were obtained by solving the Linear Programming problem

$$\begin{aligned} &\min_{m,q_2,q_1} \; q_2 - q_1 \\ &(x_i)_2 \geq m(x_i)_1 + q_1, \qquad i = 1, \ldots, 1329, \\ &(x_i)_2 \leq m(x_i)_1 + q_2, \qquad i = 1, \ldots, 1329, \end{aligned}$$

where x_i represents the coordinates of the i-th red bullet, m is the common slope of the continuous lines and q_1, q_2 are the intersections of the continuous lines with the ordinate axis.

Following our perspective from Section 5 we consider the points in Figure 15 as the *fellows*. Then, we compute the sets $L_{\max}$ and $L_{\min}$ (see Figure 16), being now respectively $L_{\max}$ indicated by $L_{East-South}$ and $L_{\min}$ indicated by $L_{West-North}$, with

- $L_{East-South}$: the weak Pareto front associated with both the maximization of the stock-to-flow SF and the minimization of Bitcoin price;
- $L_{West-North}$: the weak Pareto front associated with both the minimization of the stock-to-flow SF and the maximization of Bitcoin price.

Note that the use of different scaling in Figure 15 and Figure 16 suggests that our analysis is quite general, and holds regardless of the law of compression for data (i.e. the bases of the logarithms) we adopt. We remark that the sets $L_{East-South}$ and $L_{West-North}$ are linearly separable and the three parallel lines in Figure 16 are yielded at the end of step $k = 0$ of Algorithm SEP. In particular, the central line is the best separating hyperplane $\pi^* = H(\beta^{(0)}, \beta_0^{(0)}; 0) = 0$, while the others are the support hyperplanes. The thin stripe delimited by the support hyperplanes suggests a range of possible prices for Bitcoin, for each value of the SF. Of course additional iterations in Algorithm SEP could further narrow this stripe; however, considering Bitcoin price volatility, a refined solution where $k \geq 1$ possibly does not provide a more accurate and reliable forecast.
Moreover, as for the analysis where data associated with different assets is used, we might discard

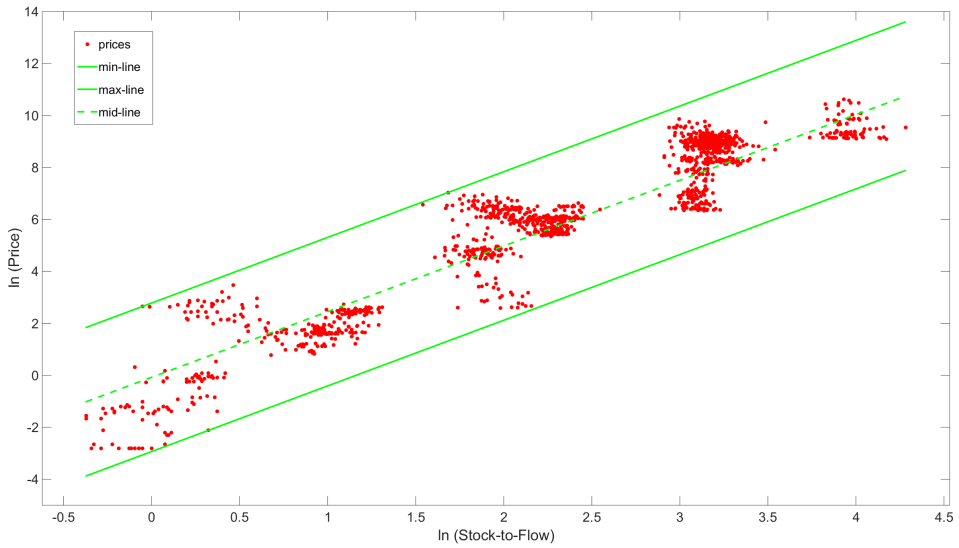

Figure 15: The price of Bitcoin vs. the stock-to-flow ratio. Continuous lines (*support lines*) delimitate the narrowest stripe containing all Bitcoin transactions. The dashed line is expected to give a trend for Bitcoin price, with respect to its stock-to-flow.

some samples price–SF, since they might be considered outliers. This is easily motivated by the fact that small values of SF refer to the early history of Bitcoin, so that the use of only more recent and reliable data can be more convincing.

6.4 Additional comparative analysis of performance based on orthogonal features

In this brief section we clarify to what extent our analysis differs (and is possibly limited) with respect to the three proposed case studies, in the light of some noteworthy reference features. Observe that in the first case study, the performance of all the fellows are the result of a *limited interaction* among them, and can be analyzed with respect to a couple of issues: the *time scenario* at which the performance is evaluated, the possible thresholds for evaluation which can be exogenously introduced by the managers of the unit (compare with Table 5 results). Indeed, a *sequence of possible scenarios* choosing different time windows for data can be compared, that can provide clear indications on the trend of performance for each fellow in the unit. E.g., selecting adjacent time windows allows to generate a sequence of tables similar to Table 5, so that the resulting sequence of four-entries performance associated to each fellow straightforwardly gives indications on her/his progress/downturn.

As regards the second case study, the *high interaction* among fellows is definitely a key aspect, which both steers the analysis towards a measurement of the performance but also a possible *simulation of future scenarios* for the entire unit. In this regard in Section 6.2 we have considered both performance indicators for each fellow and for the entire unit, including a comparison with a simulated scenario where fictitious fellows with guided performance were added to the unit.

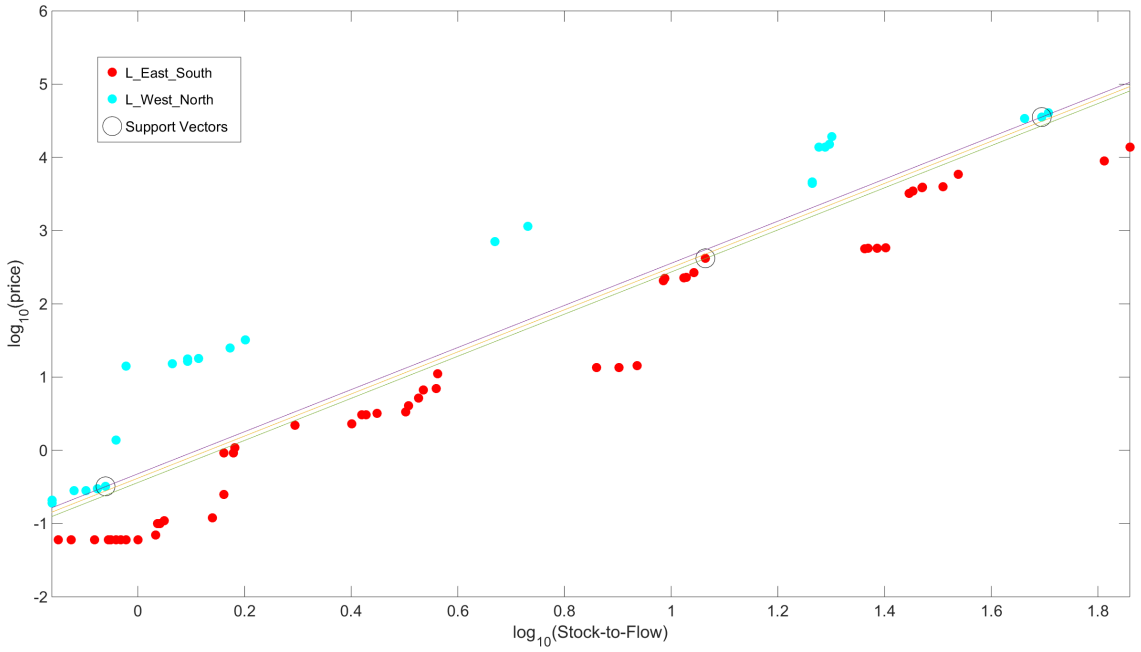

Figure 16: The fronts $L_{East-South}$ and $L_{West-North}$, along with the support vectors and the support hyperplanes, for the problem of forecasting Bitcoin price vs. its stock-to-flow ratio. Circled points represent support vectors.

As regards the third case study, it provides an *intermediate interaction* level among the fellows: indeed, close Bitcoin price/SF values might be weakly correlated, since high volatility of the crypto-asset can be experienced. For this application problem the additional keyword feature in our analysis is undoubtedly represented by *forecasting*. In this regard, while the standard literature for Bitcoin future price estimation is often based on creating reliable implicit/explicit models, we preferred to use a data-driven steering process for price forecast. We strongly remark that apparently no similar forecasting analysis seems to be straightforwardly extended to the first two case studies.

7 Conclusions and future work

The main results of the application of our SVM-based procedure to the performance measurement are:

- the improvement of sorting methods by exploiting the comparison of all the performances at the same time;
- building a plan that expresses the evaluation of the overall performance: the scatter of points re-interpreted and the possibility to simulate different states of performance system;

- a deeper understanding of the performance differentials among fellows, and a deeper understanding of the differences among the fellows and making sense of the distances coupling sorting, ranking and classification analysis;

- generating a model from the data and not preordained, overcoming the measures of centrality in the construction of clusters.

Moreover the applied computational approach supports equally the three logics: longitudinal analysis, simulation and forecasting. And it offers some insights about the possibility to enrich into an integrative perspective sorting, ranking and classification [10].

One of the most frequent questions is: How can executives make better use of the data coming from the performance appraisal? How can organizations perform the integrated performance analysis? Our application of SVMs appears to be a fruitful tool to handle data and extract information from it [11].

Our framework is also subject to a number of possible improvements and enhancements, including more sophisticated implementation issues of solvers, though our three real problems were easily tackled adopting relatively standard SVM-based formulations. In particular, the sets $L_{\max}$ and $L_{\min}$ in Table 3 might not be linearly separable but they can be separable in case a *kernel* [15, 14] is adopted for solving the formulation (15). Thus, a specific kernel can be conceived for the problem in hand, so that a better data exploitation is carried on. As a further possible improvement, the performance of the algorithm we adopted in Table 3 should be greatly increased when larger scale problems were considered. Indeed, at each step the quadratic subproblem (15) or (16) needs to be solved, so that in case the number N of points increases, a more efficient procedure becomes mandatory. In this regard, as a more recent literature on SVMs suggests, the use of Big Data implies severe limitations to the standard implementation of the solver for the quadratic subproblem.

As an additional consideration, in all our three test problems the sets $L_{\max}$ and $L_{\min}$ needed to be separated within a two dimensional subspace (say the features x_1 and x_2), but of course our proposal is much more general, implying the use of any finite number of features. Nevertheless, as well known, the larger the space of features the more difficult in general the linear separation problem (see e.g. [13]).

In the light of a future work, we also have planned to consider other three relevant improvements to our framework:

- in the case the number of features is much small (e.g. *two* as in the three application problems of this paper), there might be the chance to more easily solve the primal formulation (11), with respect to (15). That is because the dual formulation has a number of constraints $(2N+1)$ and a number of variables (N) which are both $O(N)$. Conversely, the primal formulation (11) has only *two* unknowns and N inequality constraints;

- the outcomes of Sections 5.1 and 5.2 definitely need an additional investigation, along with a comparison of their perspectives;

- the p coordinates of the points $x_1, \ldots, x_N$ are subject to an interesting interpretation, which suggests a possible update on some real applications. Indeed, the entry $(x_i)_j$ is the projection of x_i on the unit vector of the j-th axis. Thus, recalling the definition of $L_{\max}$ and $L_{\min}$, if we considered a different basis of $\mathbb{R}^p$ with respect to the canonical one $\{e_1, \ldots, e_p\}$, then

we might recompute the points $x_1, \ldots, x_N$ in the novel basis and then apply our framework. This very standard approach in numerical analysis has an interesting interpretation in some applications. Namely, we might be interested to consider a *perturbation* of the coordinate axes, which corresponds to a possible perturbation of the rules which induce the performance evaluation. We recall indeed that the geometry of the fronts $L_{\max}$ and $L_{\min}$ may considerably change, under small modifications of the directions which score the performances. As a consequence, this fact may dramatically upset the results of Algorithm SEP in Table 3, which is indeed not invariant under linear transformation of initial data.

May be that that the old saying "if you can't measure you can't improve" has a different meaning if we let the analytical process steer the conclusions.

Acknowledgements The authors wish to thank several members of the Department of Management from the Ca' Foscari University of Venice, for their valuable suggestions. Giovanni Fasano also thanks *IN*δ*AM* (Istituto Nazionale di Alta Matematica) for the support he received. Giovanni Fasano also thanks his three lawyer friends Maria, Maristella and Massimo, whose valuable perspective considerably contributed to inspire the contents of this paper.

References

[1] Guest, David E., *Human resource management and performance: a review and research agenda*, The International Journal of Human Resource Management 8(3), pp. 263–276 (1997)

[2] Kuksov, D., Villas-Boas, J. Miguel, *The Performance Measurement Trap*, Marketing Science 38(1), pp. 68–87 (2019)

[3] Haines, Victor Y., St-Onge, S., Marcoux, A., *Performance Management Design and Effectiveness in Quality-Driven Organizations*, Canadian Journal of Administrative Sciences / Revue Canadienne des Sciences de l'Administration 21(2), pp. 146–161 (2009)

[4] Sherman, J. Daniel, Keller, Robert T., *Suboptimal Assessment of Interunit Task Interdependence: Modes of Integration and Information Processing for Coordination Performance*, Organization Science 22(1), pp. 245–261 (2011)

[5] Kiggundu, Moses N., *Task interdependence and job design: Test of a theory*, Organizational Behavior and Human Performance 31(2), pp. 145–172 (1983)

[6] Zhou, Yue M., *Designing for Complexity: Using Divisions and Hierarchy to Manage Complex Tasks*, Organization Science 24(2), pp. 339–355 (2013)

[7] Sherman, J. Daniel, Keller, Robert T. *Suboptimal Assessment of Interunit Task Interdependence: Modes of Integration and Information Processing for Coordination Performance*, Organization Science 22(1), pp. 245–261 (2011)

[8] Moldoveanu, Mihnea C., Bauer, Robert M., *On the Relationship Between Organizational Complexity and Organizational Structuration*, Organization Science 15(1), pp. 98–118 (2004)

[9] de Jong, Simon B., Van der Vegt, Gerben S., Molleman, E., *The relationships among asymmetry in task dependence, perceived helping behavior, and trust*, Journal of Applied Psychology 92(6), pp. 1625–1637 (2007)

[10] Anderson, Shannon W., Kimball, A., *Evidence for the Feedback Role of Performance Measurement Systems*, Management Science 65(9), pp. 4385–4406 (2019)

[11] Abernethy, Margaret A., Dekker, Henri C., Grafton, J., *The Influence of Performance Measurement on the Processual Dynamics of Strategic Change*, Management Science 67(1), pp. 640–659 (2021)

[12] Aggarwal, D., Chandrasekaran, S., Annamalai, B., *A complete empirical ensemble mode decomposition and support vector machine–based approach to predict Bitcoin prices*, Journal of Behavioral and Experimental Finance 27, n. 100335 (2020)

[13] Cristianini, N., Shawe-Taylor, J., *An introduction to support vector machines and other kernel-based learning methods*, Cambridge University Press (2000)

[14] Vapnik, V., *The nature of the statistical learning theory*, Springer Verlag, New York (1995)

[15] Vapnik, V., *Statistical Learning Theory*, Wiley, (1998)

[16] Hastie, T., Tibshirani, R., Friedman, J., *The Elements of Statistical Learning*, Second Edition. NY Springer (2008)

[17] Fan, R.E., Chen, P.H., Lin, C.J., *Working set selection using second order information for training support vector machines*, Journal of Machine Learning Research 6, pp. 1889-1918 (2005)

[18] MATLAB, *version 8.4.0.150421 (R2014b)*, The MathWorks Inc., Natick, Massachusetts, (2014)

[19] Fourer, R., Gay, D.M., Kernighan, B.W., *AMPL: A Modeling Language for Mathematical Programming*, 2nd Ed., Brooks/Cole – Thomson Learning (2003)

[20] Graybill, F.A., Iyer, H.K. *Regression Analysis: Concepts and Applications*, Duxbury Press, Belmont, CA (1994)

[21] Marsland, S., *MACHINE LEARNING: An Algorithmic Perspective*, 2nd Ed., CRC Press Taylor & Francis Group, NW (2015)

[22] Czyzyk, J., Mesnier, M. P., Moré, J. J., *The NEOS Server*, IEEE Journal on Computational Science and Engineering 5(3), pp. 68-75, https://neos-guide.org (1998)

[23] Nakamoto, S., *Bitcoin: A peer-to-peer electronic cash system*, http://www.bitcoin.org/bitcoin.pdf (2009)

[24] Ishizaka, A., Nemery, P., *Multi-criteria Decision Analysis: Methods and Software*, Wiley (2013)

[25] Izenman, A.J., *Linear Discriminant Analysis*. In: Modern Multivariate Statistical Techniques, Springer Texts in Statistics. Springer, New York, NY (2013)

[26] Deng, N., Tian, Y., Zhang, C., *Support Vector Machines - Optimization Based Theory, Algorithms, and Extensions*, Chapman and Hall/CRC (2013)

[27] Chapelle, O., Schölkopf, B., Zien, A. *Semi-Supervised Learning*, The MIT Press, Cambridge, Massachusetts London, England

[28] Seber, G.A.F, Wild, C.F. *Nonlinear Regression*, John Wiley & Sons, NY (1989)

[29] Sreekanth Reddy, L., Sriramya, Dr.P., *A Research On Bitcoin Price Prediction Using Machine Learning Algorithms*, International Journal of Scientific & Technology Research 9(4), pp. 1600–1604 (2020)

[30] Roberts, D.A., Yaida, S., Hanin, B., *The Principles of Deep Learning Theory*, arXiv:2106.10165, cs.LG, Technical Report MIT-CTP/5306 (2021)

[31] Joachims, T., *Transductive inference for text classification using support vector machines*, in Proc. ICML, pp. 200–209 (1999)

[32] Chen, Y., Wang, G., Dong, S., *Learning with progressive transductive support vector machine*, Pattern Recognition Letters 24(12), pp. 1845–1855 (2003)