Finding Potential Support Vectors in Separable Classification Problems

Damiano Varagnolo, Simone Del Favero, Francesco Dinuzzo,
Luca Schenato and Gianluigi Pillonetto

***Abstract*—The paper considers the classification problem using Support Vector Machines, and investigates how to maximally reduce the size of the training set without losing information. Under separable dataset assumptions, we derive the exact conditions stating which observations can be discarded without diminishing the overall information content. For this purpose, we introduce the concept of Potential Support Vectors, i.e., those data that can become Support Vectors when future data become available. Complementary, we also characterize the set of Discardable Vectors, i.e., those data that, given the current dataset, can never become Support Vectors. These vectors are thus useless for future training purposes, and can eventually be removed without loss of information. We then provide an efficient algorithm based on linear programming which returns the potential and discardable vectors by constructing a simplex tableau. Finally we compare it with alternative algorithms available in the literature on some synthetic data as well as on datasets from standard repositories.**

***Index Terms*—Separable datasets, Support Vector Machines, Data discardability conditions, Potential Support Vectors, Discardable Vectors, Linear Programming.**

I. Introduction

The high generalization capabilities and interesting mathematical properties of Support Vector Machines (SVMs) triggered their success in pattern recognition. Despite their good qualities [1], SVMs may suffer from computational complexity problems in both training and evaluation phases. Instructing SVMs, in fact, requires solving a Quadratic Program (QP) with dimension equal to the number of data points [2].

A challenging problem arises when training steps cannot be performed exploiting the whole dataset simultaneously because of computational or memory constraints. The aim is then to derive suitable incremental learning techniques that divide the main problem into subtasks and compute a final outcome close to the one obtainable by a batch training [3], [4], [5], [6]. Similar difficulties are also encountered when assuming that additional training data will become available in the future. In both cases it may be beneficial to discard the non informative data, i.e., those data that cannot affect the final outcome.

As recalled in [7], data-discarding techniques can be divided into two main classes depending if they use or do not use the SVM classifier. Among the procedures that do not use previously trained SVMs, the one suggested in [8] computes the centroids of the two classes and then discards an example if it close to the centroid of its class *and* far from the centroid of the other classes. In [9], the following conjecture is instead suggested: if the class of a given example does not coincide with the one estimated by a k-Nearest Neighbors (k-NN) classifier then that example has high probability of becoming a Support Vector (SV). In [10], this probability is related to the variability of the labels of its neighbors, while the approach described in [11] exploits the distance of data from the discriminant function obtained from a Fisher Discriminant Analysis (FDA). In [12], first the dataset is clustered through a k-means algorithm and then the clusters containing the data of the same class are condensed into a single point (namely the cluster centroid). The following are instead dataset-reduction strategies that exploit previously trained SVM decision functions. In [13], the authors train an approximate decision surface choosing a subset of the training basis functions via a greedy algorithm. The works [14], [15] instead propose to create a virtual dataset of a fixed size by defining the decision function which best approximates the one obtained using the true dataset. The approach developed in [16] exploits the fact that, for linearly separable datasets, the Optimal Separating Hyperplane (OSH) is the median hyperplane of the smallest segment joining the convex hulls of the observations. Then, the SVM is trained using approximate descriptions of these points. Other authors condensate linearly dependent SVs [17], [18], based on approximated solutions [19], [20] or based on opportune projection-based operations [7], [21]. In [22], all the data sufficiently far from the separating hyperplane are discarded. In [23], authors propose to opportunely modify the original training set by removing or flipping the labels of misclassified data.

Despite offering several and often effective strategies, the literature reviewed above provides only heuristics to address the following problem:

Damiano Varagnolo (corresponding author) is with the Automatic Control Laboratory, School of Electrical Engineering, KTH, Osquldas väg 10, Stockholm, Sweden. Email: damiano@kth.se. Francesco Dinuzzo is with Max Planck Institute for Intelligent Systems, University of Tübingen, Spemannstrasse 38, 72076 Tübingen Germany. Email: fdinuzzo@tuebingen.mpg.de. Simone Del Favero, Luca Schenato and Gianluigi Pillonetto are with the Department of Information Engineering, University of Padova, via Gradenigo 6/b, 35131 Padova, Italy. Emails: { simone.delfavero | giapi | schenato } @dei.unipd.it.

The research leading to these results has received funding from the European Union Seventh Framework Programme [FP7/2007-2013] under grant agreement n°257462 HYCON2 Network of excellence, by the MIUR FIRB project RBFR12M3AC - Learning meets time: a new computational approach to learning in dynamic systems, by the Swedish Governmental Agency for Innovation Systems (VINNOVA) and the Knut and Alice Wallenberg Foundation.

Question 1 which is the smallest subset of the data that carry all the information useful for future retraining purposes?

To answer this question, it is needed to understand if an observation can be discarded at the present time without affecting the forthcoming generalization capabilities. Incidentally, this issue is especially important also in distributed SVMs scenarios, where communication constraints require to minimize the amount of information to be exchanged among agents [24], [25], [26], [27], [28], [29].

To the best of our knowledge, even under linear separability assumptions, an answer to the question reported above has not been provided, i.e., the necessary and sufficient conditions that establish when an example can carry information in the future are not available in the literature. The aim of this paper is to provide a full and detailed answer under separability assumptions. In particular, we propose a novel discardability concept based on a precise mathematical formulation of the information content of a data set: an observation contains information if and only if it can become a SV, and we refer to these types of examples as Potential Support Vectors (PSVs). Then, we show that the discardability conditions can be verified by a simplex-based algorithm, i.e., by a Linear Program (LP). A peculiarity of the data discarding algorithm presented in this paper is that it can be improved only with respect to its computational time, not on the outcome that it returns. Our findings also show that many standard heuristics, such as those rejecting an example considering only its distance from the OSH, may be misleading and bringing to information losses.

The paper is organized as follows: Sec. II reports some useful notation while Sec. III briefly introduces the linear SVM framework and the concepts of Discardable Vectors (DVs) and PSVs. Sec. IV offers various characterizations of these concepts, while Sec. V translates them into numerical data discardability-checking procedures. Sec. VI then extends the previous findings to nonlinear separability problems, while Sec. VII compares the performances of our strategy with some algorithms available in the literature by means of numerical experiments on standard dataset repositories and synthetic data. Finally Sec. IX contains some concluding remarks and discusses future research directions.

II. Notation

The following notation will be adopted in what follows:

- bold fonts indicate vectorial quantities or functions whose range is vectorial, plain italic fonts indicate scalar quantities or functions whose range is a scalar, capital italic fonts indicate matrix quantities;
- i indexes the elements in the dataset;
- d is the dimension of the domain of the inputs;
- $\boldsymbol{x}_i = [x_{i,1}, \dots x_{i,d}]^T \in \mathbb{R}^d$ is a generic input location;
- $\psi : \mathbb{R}^d \mapsto \mathbb{H}$ is a generic feature map transforming input locations $\boldsymbol{x}_i$ into the corresponding $\psi(\boldsymbol{x}_i)$'s in the feature space $\mathbb{H}$;
- $K : \mathbb{R}^d \times \mathbb{R}^d \mapsto \mathbb{R}$ is the kernel corresponding to the feature map $\psi(\cdot)$, i.e., is s.t. $\langle \psi(\boldsymbol{x}_1), \psi(\boldsymbol{x}_2) \rangle_{\mathbb{H}} = K(\boldsymbol{x}_1, \boldsymbol{x}_2)$ for all $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathbb{R}^d$;
- $y_i \in \{+1, -1\}$ is the generic output;
- $\mathbf{0} := (0, \dots, 0) \in \mathbb{R}^d$;
- $\mathcal{D} := \{(\boldsymbol{x}_i, y_i)\}_{i=1,\dots,n}$ is the dataset. "\" indicates the set-theoretic subtraction;
- n is the total number of data in $\mathcal{D}$;
- $\mathcal{X} := \{\boldsymbol{x}_i\}_{i=1,\dots,n}$ is the set of input locations;
- the sets of input locations corresponding to positive and negative outputs are respectively denoted by

$$\mathcal{X}^+ := \{\boldsymbol{x}_i \in \mathcal{X} \text{ such that } y_i = +1\} \tag{1}$$

and

$$\mathcal{X}^- := \{\boldsymbol{x}_i \in \mathcal{X} \text{ such that } y_i = -1\} ; \tag{2}$$

- ∂A indicates the boundary of the set A (under the classical Euclidean topology);
- $\text{int}\,(A)$ indicates the interior of the set A (under the classical Euclidean topology);
- to a generic vector $(\boldsymbol{w}, b) \in \mathbb{R}^{d+1}$ we associate the hyperplane $\mathcal{H}_{\boldsymbol{w},b} := \left\{(\boldsymbol{x}, y) \,\middle|\, \boldsymbol{x} \in \mathbb{R}^d, y = \boldsymbol{w}^T\boldsymbol{x} + b\right\}$ with elements in $\mathbb{R}^{d+1}$, and the hyperplane $\mathcal{H}^0_{\boldsymbol{w},b} := \left\{\boldsymbol{x} \in \mathbb{R}^d \,\middle|\, \boldsymbol{w}^T\boldsymbol{x} + b = 0\right\}$ on the reduced space $\mathbb{R}^d$.

We also recall some basic definitions and facts on geometry and convex analysis:

- a *cone* $\mathcal{K} \subseteq \mathbb{R}^d$ is a set such that if $\boldsymbol{x} \in \mathcal{K}$ and $\lambda \geq 0$ then $\lambda\boldsymbol{x} \in \mathcal{K}$;
- a *convex cone* $\mathcal{K} \subseteq \mathbb{R}^d$ is a set such that if $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathcal{K}$ and $\lambda_1, \lambda_2 \geq 0$ then $\lambda_1\boldsymbol{x}_1 + \lambda_2\boldsymbol{x}_2 \in \mathcal{K}$;
- the *convex hull* of the set $\{\boldsymbol{x}_1, \dots, \boldsymbol{x}_n\}$ is defined as

$$\text{conv}\,(\boldsymbol{x}_1, \dots, \boldsymbol{x}_n) := \sum_{i=1}^{n} \lambda_i \boldsymbol{x}_i \tag{3}$$

with the additional constraints $\lambda_i \geq 0$ for $i = 1, \dots, n$ and $\sum_{i=1}^n \lambda_i = 1$;

- the *conical hull* of the set $\{\boldsymbol{x}_1, \dots, \boldsymbol{x}_n\}$ is defined as

$$\text{coni}\,(\boldsymbol{x}_1, \dots, \boldsymbol{x}_n) := \sum_{i=1}^{n} \lambda_i \boldsymbol{x}_i \tag{4}$$

with the additional constraint $\lambda_i \geq 0$ for $i = 1, \dots, n$. Notice that every conical hull is a convex cone;

- the *lineality* of a convex cone $\mathcal{K}$ is defined as

$$\text{Lin}\,(\mathcal{K}) := \mathcal{K} \cap -\mathcal{K} \tag{5}$$

and corresponds to the smallest subspace contained in $\mathcal{K}$;

- the *polar* of a cone $\mathcal{K}$ is indicated with $\mathcal{K}^\circ$ and corresponds to the set of vectors forming angles not smaller than 90°with every $x \in \mathcal{K}$, i.e.

$$\mathcal{K}^\circ := \{\boldsymbol{z} \in \mathbb{R}^d \;\big|\; \boldsymbol{z}^T\boldsymbol{x} \leq 0 \;\; \forall \boldsymbol{x} \in \mathcal{K}\} . \tag{6}$$

In addition, if $\mathcal{K}$ is a closed convex cone, one has

$$(\mathcal{K}^\circ)^\perp = \text{Lin}\,(\mathcal{K}) \;. \tag{7}$$

III. Linear Support Vector Classification

Here and in Sections IV, V we analyze the linear classification case. All the results will be then extended to the nonlinear case in Sec. VI.

Given then a dataset $\mathcal{D}$, the goal is to find the $(\boldsymbol{w}, b) \in \mathbb{R}^{d+1}$ whose associated hyperplane $\mathcal{H}^0_{\boldsymbol{w},b}$ is s.t. all the inputs of the same class lie on the same side, and the minimal distance between the various inputs and $\mathcal{H}^0_{\boldsymbol{w},b}$ is maximized. The existence of such an hyperplane is ensured if $\mathcal{D}$ satisfies the following definition [30]:

Definition 2 The dataset $\mathcal{D}$ is said to be *linearly separable* if there exists $(\boldsymbol{w}, b)$ s.t. $y_i(\boldsymbol{w}^T \boldsymbol{x}_i + b) \geq 1 \ \forall i$.

Assumption 3 $\mathcal{D}$ is linearly separable, and all the future data added to $\mathcal{D}$ will lead to a new dataset which is still linearly separable. Moreover $\mathcal{D}$ contains elements of both classes.

A. Classification for linearly separable datasets

If the dataset $\mathcal{D}$ is linearly separable, the SVMs framework considers as the optimal classification rule the one that solves the convex optimization problem

$$\begin{aligned}(\boldsymbol{w}^*, b^*) := \quad & \arg\min_{\boldsymbol{w},b} \|\boldsymbol{w}\|_2 \\ & \text{s.t. } y_i \left(\boldsymbol{w}^T \boldsymbol{x}_i + b\right) \geq 1, \quad i = 1, \ldots, n \ .\end{aligned} \tag{8}$$

We refer to the hyperplane $\mathcal{H}_{\boldsymbol{w}^*,b^*}$ associated to the optimal solution as to the OSH. The minimum distance between the OSH $\mathcal{H}^0_{\boldsymbol{w}^*,b^*}$ and any generic input $\boldsymbol{x}_i$, namely

$$m := \min_{\boldsymbol{x}_i \in \mathcal{X}, \boldsymbol{x} \text{ s.t. } \boldsymbol{w}^{*T}\boldsymbol{x} + b^* = 0} \|\boldsymbol{x} - \boldsymbol{x}_i\|_2 = \frac{1}{\|\boldsymbol{w}^*\|_2} \tag{9}$$

is called the *optimal margin*. The elements of the dataset s.t. the distance between their input $\boldsymbol{x}_i$ and the OSH is exactly m are called SVs. We can thus define the set

$$\mathrm{SV}(\mathcal{D}) := \left\{ (\boldsymbol{x}_i, y_i) \in \mathcal{D} \ \middle| \ \min_{\boldsymbol{x} \text{ s.t. } \boldsymbol{w}^{*T}\boldsymbol{x} + b^* = 0} \|\boldsymbol{x} - \boldsymbol{x}_i\|_2 = m \right\}. \tag{10}$$

The following definition is fundamental for our purposes.

Definition 4 $(\boldsymbol{x}_i, y_i) \in \mathcal{D}$ is a *Potential Support Vector* (PSV) for the dataset $\mathcal{D}$ if there exists a dataset $\mathcal{D}^*$ such that $(\boldsymbol{x}_i, y_i)$ is a SV for the augmented dataset $\mathcal{D} \cup \mathcal{D}^*$. Complementary, $(\boldsymbol{x}_i, y_i)$ is a *Discardable Vector* (DV) if is not a PSV.

Let $\mathrm{PSV}(\mathcal{D})$ indicate the set of all the PSVs contained in $\mathcal{D}$, while $\mathrm{DV}(\mathcal{D})$ indicate the set of all discardable vectors. Note that the definition implies that the set of support vectors is included in the set of potential support vectors and that each element of the dataset is either a potential support vector or a discardable vector, i.e.,

$$\mathrm{SV}(\mathcal{D}) \subseteq \mathrm{PSV}(\mathcal{D}), \qquad \mathrm{PSV}(\mathcal{D}) \cup \mathrm{DV}(\mathcal{D}) = \mathcal{D}, \qquad \mathrm{PSV}(\mathcal{D}) \cap \mathrm{DV}(\mathcal{D}) = \emptyset \ .$$

In the following section we provide some necessary and sufficient conditions characterizing these sets.

IV. Characterization of the Potential Support Vectors

We provide two different but equivalent conditions characterizing $\mathrm{PSV}(\mathcal{D})$ and $\mathrm{DV}(\mathcal{D})$. The first, presented in Sec. IV-A, is inherently geometrical and has the advantage to be intuitive and simple to be stated. The second, presented in Sec. IV-B, is more algebraic and technical, but has the advantage to suggest an algorithm to practically compute the PSVs.

A. Characterization in the space of the hyperplanes parameters

Hereby, it is useful to identify the $(d+1)$-dimensional hyperplane $\mathcal{H}_{\boldsymbol{w},b}$ also with its generating parameters $(\boldsymbol{w}, b)$. Hence, with a little abuse of notation, $(\boldsymbol{w}, b)$ is sometimes referred to as an hyperplane.

The generic example $(\boldsymbol{x}_i, y_i)$ splits the space of all the plausible hyperplanes $\{(\boldsymbol{w}, b)\}$ in two sets: the first is the set satisfying the inequality constraints present in (8), being defined by

$$\mathcal{V}_i := \left\{ (\boldsymbol{w}, b) \ \middle| \ y_i \left(\boldsymbol{w}^T \boldsymbol{x}_i + b\right) \geq +1 \right\} \ . \tag{11}$$

The other set is complementary to $\mathcal{V}_i$. With this definition in mind, it is immediate to recognize that the set of feasible solutions $(\boldsymbol{w}, p)$ for (8), i.e., the set of hyperplanes $(\boldsymbol{w}, b)$ that correctly separate $\mathcal{D}$, is given by

$$\mathcal{C} := \bigcap_i \mathcal{V}_i \ . \tag{12}$$

$\mathcal{C}$ is thus the so-called *version space*, i.e., the subset of all hypotheses that are consistent with the observed training examples. With definition (12) we can thus rewrite (8) as follows:

$$(\boldsymbol{w}^*, b^*) = \arg \min_{(\boldsymbol{w},b) \in \mathcal{C}} \|\boldsymbol{w}\|_2 \ . \tag{13}$$

The set $\mathcal{C}$ is convex, being the intersection of convex sets, and non-empty under Assumption 3. Indeed it is not difficult to see that it is also a cone. We notice that Assumption 3 implies also that $\mathcal{C} \neq \emptyset$ will continue to hold even when future data will be added.

Graphical intuitions can be gathered considering the sets

$$\begin{cases} \mathcal{L}^+ := \{(\boldsymbol{w}, b) \,|\boldsymbol{w} = \boldsymbol{0}, b \geq +1\} & \text{(14a)} \\ \mathcal{L}^- := \{(\boldsymbol{w}, b) \,|\boldsymbol{w} = \boldsymbol{0}, b \leq -1\} & \text{(14b)} \end{cases}$$

for which it holds that if $y_i = +1$ then $\mathcal{L}^+ \subseteq \mathcal{V}_i$, while if $y_i = -1$ then $\mathcal{L}^- \subseteq \mathcal{V}_i$, for every $(\boldsymbol{x}_i, y_i)$. This geometrically corresponds to the fact that if $y_i = +1$ then $\mathcal{V}_i$ must "look upwards", while if $y_i = -1$ then $\mathcal{V}_i$ must "look downwards" (cf. Fig. 1).

Let now $\mathcal{B}_i$ be the boundary of $\mathcal{V}_i$, i.e.,

$$\mathcal{B}_i := \partial \mathcal{V}_i = \left\{ (\boldsymbol{w}, b) \ \middle| \ \boldsymbol{w}^T \boldsymbol{x}_i + b = y_i \right\} \ . \tag{15}$$

We notice that the definition of SV given in (10) implies immediately that $(\boldsymbol{x}_i, y_i)$ is a SV if and only if $(\boldsymbol{w}^*, b^*) \in \mathcal{B}_i$. The following constitutes the exact characterization of the set $\mathrm{DV}(\mathcal{D})$ under assumption 3:

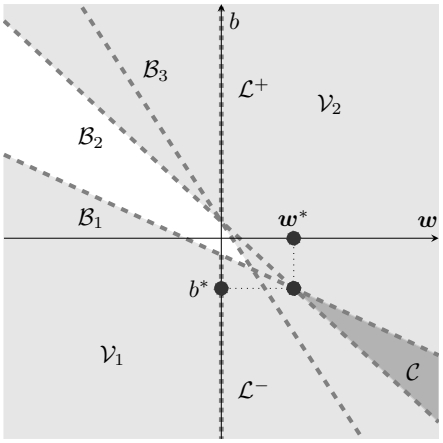

Figure 1. Pictorial representation of how a dataset composed by three examples halves the space of the plausible parameters $(\boldsymbol{w}, b)$ and generates the version space $\mathcal{C}$. To each example $(\boldsymbol{x}_i, y_i)$ corresponds a set $\mathcal{V}_i$ that splits the space of parameters $(\boldsymbol{w}, b)$ into two distinct half-spaces. The version space $\mathcal{C}$ corresponds to the intersection of all these half-spaces, and thus contains all and only the parameters $(\boldsymbol{w}, b)$ that simultaneously satisfy all the inequality constraints (8). Notice that, in this case, $(\boldsymbol{x}_3, y_3)$ does not characterize $\mathcal{C}$ in the sense that it does not contribute to the definition of its boundary.

Proposition 5 Let $\mathcal{D}$ be linearly separable. Then $(\boldsymbol{x}_i, y_i) \in \text{DV}(\mathcal{D})$ if and only if $\mathcal{C} \subset \text{int}(\mathcal{V}_i)$. Equivalently, $(\boldsymbol{x}_i, y_i) \in \text{PSV}(\mathcal{D})$ if and only if $\mathcal{B}_i \cap \mathcal{C} \neq \emptyset$.

From the proof of the previous proposition, it follows that if an example is a PSV, then it is always possible to add just a single additional example to turn it into a SV, as formally stated in the following corollary.

Corollary 6 Let $\mathcal{D}$ be linearly separable. If $(\boldsymbol{x}_i, y_i) \in \text{PSV}(\mathcal{D})$, then there exists $(\boldsymbol{x}_F, y_F)$ such that $\overline{\mathcal{D}} := \mathcal{D} \cup (\boldsymbol{x}_F, y_F)$ is still linearly separable and $(\boldsymbol{x}_i, y_i) \in \text{SV}\left(\overline{\mathcal{D}}\right)$.

Recall also, from the definition of discardability that, if an example is a DV, then under assumption 3 it will always remain a DV. In other words, if $\mathcal{D}$ and $\mathcal{D} \cup \overline{\mathcal{D}}$ are linearly separable and $(\boldsymbol{x}_i, y_i) \in \text{DV}(\mathcal{D})$ then $(\boldsymbol{x}_i, y_i) \in \text{DV}\left(\mathcal{D} \cup \overline{\mathcal{D}}\right)$.

We remark that the propositions proposed in this section provide a geometrically intuitive interpretation of the PSVs and the DVs, but do not lead to algorithms to practically compute these sets. The next subsection instead provides an alternative characterization that directly leads to an algorithm for the computation of $\text{PSV}(\mathcal{D})$ and $\text{DV}(\mathcal{D})$.

B. Characterization in the inputs space

We derive a characterization of the PSVs in the inputs space that is alternative but equivalent to Prop. 5. Before continuing we need the following:

Definition 7 The hyperplane $(\boldsymbol{w}, b) \in \mathbb{R}^{d+1}$ is said to *quasi linearly separate* a dataset $\mathcal{D}$ if $y_i(\boldsymbol{w}^T \boldsymbol{x}_i + b) \geq 0$ for all i.

The following constitutes a characterization of the set of all Potential Support Vectors of a given linearly separable dataset $\mathcal{D}$:

Proposition 8 Let $\mathcal{D}$ be linearly separable. Then $(\boldsymbol{x}_i, y_i) \in \text{PSV}(\mathcal{D})$ if and only if there exists a quasi linearly separating hyperplane $(\boldsymbol{w}, b) \in \mathbb{R}^{d+1}$ s.t. $\boldsymbol{w}^T \boldsymbol{x}_i + b = 0$ and s.t. $\boldsymbol{w}^T \boldsymbol{x}_j + b \neq 0$ for all j s.t. $y_j \neq y_i$.

We remark that the quasi-separating hyperplane cited in the above proposition must pass through $(\boldsymbol{x}_i, 0)$, must not pass through any point $(\boldsymbol{x}_j, 0)$ of the opposite class, and can pass through points $(\boldsymbol{x}_j, 0)$ of the same class. Fig. 2 offers a graphical intuition of Prop. 8 for the case $d = 2$ (i.e., $\boldsymbol{x} \in \mathbb{R}^2$).

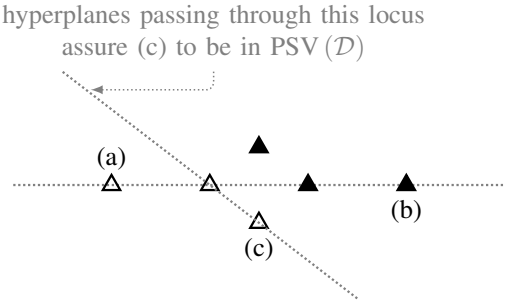

Figure 2. Graphical intuition behind Prop. 8 for $d = 2$. Notice that the points in the picture are of the form $(\boldsymbol{x}, 0)$, and that for $d = 2$ hyperplanes are made of points $(\boldsymbol{x}, \boldsymbol{w}^T \boldsymbol{x} + b)$ in $\mathbb{R}^3$. Prop. 8 then states that an example $(\boldsymbol{x}_i, y_i)$ is in $\text{PSV}(\mathcal{D})$ if and only if at least one of all the hyperplanes passing through $(\boldsymbol{x}_i, 0)$ quasi separates $\mathcal{D}$ while not not passing through any point $(\boldsymbol{x}_j, 0)$, with $(\boldsymbol{x}_j, y_j)$ of the opposite class. The figure shows the limit case where the examples (a) and (b) are not PSVs. In fact, all the hyperplanes that pass through them and that quasi separate $\mathcal{D}$ pass through at least an example of the opposite class, and thus violate the proposition. The example (c), instead, is a PSV.

To transform Prop. 8 into a numerically evaluable condition, consider a generic $(\boldsymbol{x}_i, y_i) \in \mathcal{D}$, and let (cf. Fig. 3)

$$\Delta_{ij} := y_i y_j (\boldsymbol{x}_i - \boldsymbol{x}_j), \quad j = 1, \ldots, n, \quad j \neq i \tag{16}$$

The following proposition provides another full characterization of the PSVs under separability assumptions:

Proposition 9 Let $\mathcal{D}$ be linearly separable. Then the following assertions are equivalent:

1) $(\boldsymbol{x}_i, y_i) \in \text{PSV}(\mathcal{D})$;
2) there exists $\boldsymbol{w} \in \mathbb{R}^d$ s.t.

$$\begin{cases} \Delta_{ij}^T \boldsymbol{w} \leq 0, & \forall j \in \{j \mid y_i = y_j,\ j \neq i\} \quad (17a) \\ \Delta_{ij}^T \boldsymbol{w} < 0, & \forall j \in \{j \mid y_i \neq y_j\}\ . \quad (17b) \end{cases}$$

From Prop. 9 it is possible to obtain the following well known sufficient condition for data discardability [31]:

Corollary 10 Let $\mathcal{D}$ be linearly separable. If $\boldsymbol{x}_i \in \text{int}\left(\text{conv}\left(\{\boldsymbol{x}_j\}_{j \mid y_i = y_j}\right)\right)$ then $(\boldsymbol{x}_i, y_i) \in \text{DV}(\mathcal{D})$.

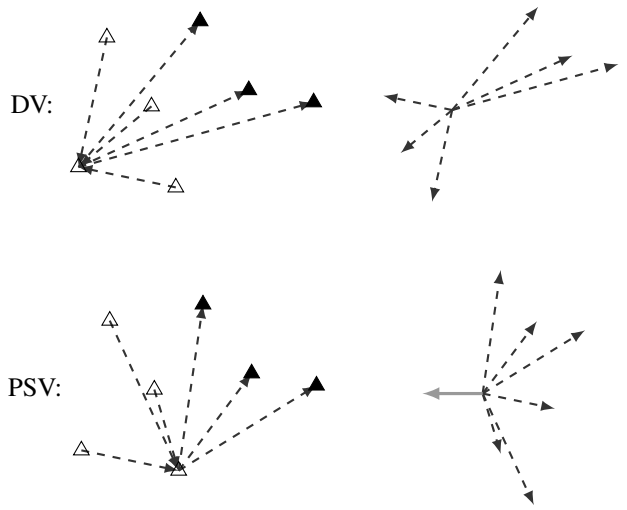

Figure 3. Computation of the various Δ_{ij}'s defined in (16) for a linearly separable dataset. The top panel considers the Δ_{ij}'s constructed starting from a DV, while the bottom panel refers to the case of a PSV. The gray arrow in the bottom panel indicates a $\boldsymbol{w}$ satisfying the conditions stated in Prop. 9. Importantly, the set of Δ_{ij}'s is invariant w.r.t. the class assignment.

The intuition behind Cor. 10 is the following: if the input of $(\boldsymbol{x}_i, y_i)$ is inside the convex hull of the inputs of the data of its class then there is no vector $\boldsymbol{w}$ that can satisfy (17a).

We also notice that if an example is a DV, then this property does not depend on the presence of other DVs, as formally stated in the following:

Proposition 11 Let $\mathcal{D}$ be linearly separable. If $(\boldsymbol{x}_i, y_i), (\boldsymbol{x}_j, y_j) \in \mathrm{DV}(\mathcal{D})$ then $(\boldsymbol{x}_j, y_j) \in \mathrm{DV}(\mathcal{D} \setminus (\boldsymbol{x}_i, y_i))$.

This proposition implies that it is not required to sort the dataset before running data-removal steps, since the discardability of a vector is not affected when removing other discardable vectors. This property will be useful for implementation purposes as shown in the next section.

We also notice that Prop. (11) implicitly describes a scalability property of the consequent numerical procedures. In fact it enables incremental analyses, where the original dataset is split into parts that are then treated consequently.

V. Numerical computation of the Potential Support Vectors under linearity assumptions

We now provide a numerical procedure to compute the sets $\mathrm{PSV}(\mathcal{D})$ and $\mathrm{DV}(\mathcal{D})$ that is based just on *checking whether a suitable LP is unbounded or not*. Consider in fact the following:

Lemma 12 Let $\mathcal{D}$ be linearly separable and $(\boldsymbol{x}_i, y_i) \in \mathcal{D}$. Consider the following (feasible) LP

$$\max_{\omega \in \mathbb{R},\, \boldsymbol{w} \in \mathbb{R}^d} \quad \omega$$
$$\text{s.t.} \begin{cases} \omega & \geq 0 \\ \Delta_{ij}^T \boldsymbol{w} & \leq 0 \quad \text{if } y_j = y_i,\ j \neq i \\ \Delta_{ij}^T \boldsymbol{w} + \omega & \leq 0 \quad \text{otherwise.} \end{cases} \tag{18}$$

If $(\boldsymbol{x}_i, y_i) \in \mathrm{PSV}(\mathcal{D})$ then (18) is unbounded. If instead $(\boldsymbol{x}_i, y_i) \in \mathrm{DV}(\mathcal{D})$ then (18) has $\omega^* = 0$ as optimum.

Lemma 12 can thus be translated into the following Alg. 1, that just checks whether it is possible to move away from the feasible solution $(\boldsymbol{w}, \omega) = (\boldsymbol{0}, 0)$ or not.

Algorithm 1 (SVM-PSV) Computation of $\mathrm{PSV}(\mathcal{D})$ for linearly separable datasets

set $\mathrm{PSV}_{\text{list}} = \mathcal{D}$
for $i = 1, \ldots, n$ **do**
  for $(\boldsymbol{x}_j, y_j) \in \mathrm{PSV}_{\text{list}}$, $i \neq j$ **do** $\Delta_{ij} = (\boldsymbol{x}_i - \boldsymbol{x}_j) y_i y_j$
  end for
  check whether the LP (18) admits a non-null feasible solution or not (case that would imply $\omega^* = 0$)
  if $\omega^* = 0$ **then** $\mathrm{PSV}_{\text{list}} = \mathrm{PSV}_{\text{list}} \setminus (\boldsymbol{x}_i, y_i)$
  end if
end for

In the following we will refer to Alg. 1 to as SVM-PSV. Such procedure correctly returns the list of all the potential support vectors in view of Prop. 11 and Lemma 12.

A. Analysis of the computational complexity of Algorithm 1

Since the algorithm requires to check just whether the LP (18) is unbounded or not, we can construct a simplex pivot table starting from the feasible solution $(\boldsymbol{w}, \omega) = (\boldsymbol{0}, 0)$ and then check if it is unbounded or not exploiting, e.g., Step 3 in [32, p. 47] (the procedure is reported in Appendix for sake of completeness).

The complexity of the proposed approach[1] is thus the following: the construction of the LP (18) requires the construction of a simplex pivot table with $O(d \times n)$ elements. Checking its unboundedness corresponds to perform $O(d \times n)$ sums and to check $O(n)$ inequalities. If no recursive strategy is implemented, the dimensionality of the pivot table deriving from LP (18) decreases as the iteration counter increases when data are discarded. The worst-case scenario is thus when all the points are not discarded and all the simplex tables derive from LPs with $|\mathcal{D}|$ constraints.

We also notice that the algorithm is suitable for parallel computations.

[1]We notice that solving the *entire* simplex algorithm in its original form has an exponential worst-case complexity. The existence of variations of the simplex algorithm with polynomial or sub-exponential worst-case complexities is instead, at the best of our knowledge, still an open problem. We also notice that our approach practically corresponds to perform just one simplex step.

B. Pre-discardability through convex hulls

When the dimensionality d of the inputs $\boldsymbol{x}_i$ is small it may be meaningful to perform an early data-discarding step based on the sufficient condition given in Cor. 10. In these situations it may then be preferable to run the following Alg. 2 before applying SVM-PSV.

Algorithm 2 (SVM-CH) Data-discarding by means of convex hulls

1: compute $\text{conv}\left(\mathcal{X}^+\right)$ and $\text{conv}\left(\mathcal{X}^-\right)$;
2: discard from $\mathcal{D}$ all the data $(\boldsymbol{x}_i, y_i)$ s.t. $\boldsymbol{x}_i \in \text{int}\left(\text{conv}\left(\mathcal{X}^+\right)\right)$ or $\boldsymbol{x}_i \in \text{int}\left(\text{conv}\left(\mathcal{X}^-\right)\right)$.

Alg. 2 has already been suggested in literature, see, e.g., [31]. Its computational complexity strongly depends on how steps (1) and (2) are performed, see, e.g., [33, chap. 33.3]. Nonetheless, all the algorithms that solve step (1) have in general prohibitive computational costs when d is large, e.g., when the inputs are mapped into high-dimensional feature spaces through basis expansions. For instance, the *quick-hull algorithm* [34] has worst-case computational complexity $O\left(n\frac{r^{\lfloor d/2\rfloor-1}}{\lfloor d/2\rfloor!}\right)$ where n is the number of points for which the convex hull has to be computed, r is the number of vertexes of the hull, and d is the geometric space dimensionality. Other algorithms instead, e.g., the *gift wrapping algorithm* [35], [36], may be simpler to implement but have worse computational complexity.

Step (2) (generally referred to as the *redundancy removal problem*) can be implemented for general d's by means of LPs, which can be solved in (weakly) polynomial time. We eventually notice that checking whether $\boldsymbol{x}_i \notin \text{conv}\left(\{\boldsymbol{x}_j | y_i = y_j\}\right)$ is extremely simple for $d = 2$. Embedding the various points in the set of complex numbers $\mathbb{C}$ and assuming that the $\boldsymbol{x}_j$'s with $y_j = y_i$ are sorted s.t. $\arg\left(\boldsymbol{x}_j - \boldsymbol{x}_i\right) < \arg\left(\boldsymbol{x}_{j+1} - \boldsymbol{x}_i\right)$, then

$$\boldsymbol{x}_i \notin \text{conv}\left(\{\boldsymbol{x}_j | y_i = y_j\}\right) \;\Leftrightarrow\; \sum_j \arg\left(\frac{\boldsymbol{x}_{j+1} - \boldsymbol{x}_i}{\boldsymbol{x}_j - \boldsymbol{x}_i}\right) \neq 2\pi \;. \tag{19}$$

C. Analysis of the output of Algorithm 1

If a dataset $\mathcal{D}$ is linearly separable then the list of PSVs is non-empty, since containing also the SVs. Thus if $\mathcal{D}$ is linearly separable then Alg. 1 returns a non-empty list of PSVs. On the contrary:

Proposition 13 Consider implementations of Alg. 1 where DVs are not removed from the dataset. If $\mathcal{D}$ is not linearly separable then the list of returned PSVs is empty.

Prop. 13 might suggest to test the linear separability of a given dataset $\mathcal{D}$ by repeatedly applying Alg. 1 to the various training points. We then notice that this choice would be inefficient, since[2]:

[2]Other separability tests can be found in [37], [38, Chap. 5].

Lemma 14 Consider the following LP

$$\max_{\omega\in\mathbb{R},\ \boldsymbol{w}\in\mathbb{R}^d,\ b\in\mathbb{R}} \omega \quad \text{s.t.} \begin{cases} \omega & \geq 0 \\ y_i\left(\boldsymbol{w}^T\boldsymbol{x}_i + b\right) - \omega & \geq 0 \qquad i = 1,\ldots,n. \end{cases} \tag{20}$$

The dataset $\mathcal{D}$ is linearly separable if and only if (20) is unbounded. On the contrary, the dataset $\mathcal{D}$ is not linearly separable if and only if the optimum in (20) is $\omega^* = 0$.

It is faster to check the separability through (20) rather than through Prop. 13, since the former requires less and smaller simplex pivot tables. We recall that testing the unboundedness of an LP like (20) does not require to compute its solution, but rather to just construct a simplex pivot table from $\omega = 0, \boldsymbol{w} = \boldsymbol{0}, b = 0$ and perform Step 3 in [32, p. 47], consisting in testing a set of scalar inequalities.

VI. Extension to the nonlinear case

Non linearly separable datasets are often approached by mapping the original data into alternative Hilbert spaces, then applying the linear classification approach described before [39, Sec. 12.3] [40, Sec. 10.5].

Formally, this is performed using opportune *kernels* $K : \mathbb{R}^d \times \mathbb{R}^d \mapsto \mathbb{R}$, i.e., continuous, symmetric and positive definite functions that, when restricted to compact domains, satisfy the following [39, Sec. 5.8]:

Assumption 15 K admits the eigen-decomposition

$$K(\boldsymbol{x}_1, \boldsymbol{x}_2) = \sum_{e=1}^{+\infty} \lambda_e \phi_e(\boldsymbol{x}_1)\phi_e(\boldsymbol{x}_2), \quad \lambda_e \geq 0, \quad \sum_{e=1}^{+\infty} \lambda_e^2 < +\infty \tag{21}$$

where the eigenfunctions $\phi_e : \mathbb{R}^d \mapsto \mathbb{R}$ are continuous.

Kernels as in (21) define nonlinear functions $\psi : \mathbb{R}^d \mapsto \mathbb{H}$, called *feature maps*, transforming the generic vector $\boldsymbol{x}_i \in \mathbb{R}^d$ into elements $\psi(\boldsymbol{x}_i)$ of a separable Hilbert space $\mathbb{H}$, e.g.,

$$\begin{aligned} \psi(\boldsymbol{x}) &= \left[\sqrt{\lambda_1}\phi_1(\boldsymbol{x}), \sqrt{\lambda_2}\phi_2(\boldsymbol{x}), \ldots\right]^T, \\ \langle \psi(\boldsymbol{x}_1), \psi(\boldsymbol{x}_2)\rangle_{\mathbb{H}} &= \psi(\boldsymbol{x}_1)^T\psi(\boldsymbol{x}_2). \end{aligned} \tag{22}$$

Through (22), the original points $\boldsymbol{x}_i \in \mathbb{R}^d$ are mapped into another Hilbert space $\mathbb{H}$, equal either to $\mathbb{R}^f$ (if the set of non-null λ_e's is finite and has cardinality f) or to ℓ_2, the set of square summable series.

After mapping the original data, linear classification is performed on the novel set of points. It comes that all the previous concepts (Propositions 5, 8, 9, 11 13, Corollaries 6, 10, Def. 7, Lemmas 12, 14, Alg. 1) apply to the nonlinear case as soon as the original $\boldsymbol{x}_i \in \mathbb{R}^d$, $(\boldsymbol{w}, b) \in \mathbb{R}^{d+1}$ and $(\boldsymbol{x}, y) \in \mathbb{R}^{d+1}$ are substituted with their counterparts in the feature space $\boldsymbol{\chi}_i := \psi(\boldsymbol{x}_i) \in \mathbb{H}$, $(\boldsymbol{v}, b) \in \mathbb{H}\times\mathbb{R}$ and $(\boldsymbol{\chi}, y) \in \mathbb{H}\times\{+1, -1\}$. For example, problem (8) and system (18) in Lemma 12

become

$$(\boldsymbol{v}^*, b^*) := \begin{array}{l} \arg\min_{\boldsymbol{v},b} \|\boldsymbol{v}\|_2 \\ \text{s.t. } y_i\left(\boldsymbol{v}^T\boldsymbol{\chi}_i + b\right) \geq 1, \quad i = 1, \ldots, n\,, \end{array} \tag{23}$$

$$\max_{\omega\in\mathbb{R},\boldsymbol{v}\in\mathbb{H}} \quad \omega$$
$$\text{s.t.} \begin{cases} \omega & \geq 0 \\ y_i y_j\left(\boldsymbol{v}^T\boldsymbol{\chi}_i - \boldsymbol{v}^T\boldsymbol{\chi}_j\right) & \leq 0 \quad \text{if } y_j = y_i,\ j \neq i \\ y_i y_j\left(\boldsymbol{v}^T\boldsymbol{\chi}_i - \boldsymbol{v}^T\boldsymbol{\chi}_j\right) + \omega & \leq 0 \quad \text{otherwise.} \end{cases} \tag{24}$$

In certain cases, especially when the dimensionality of $\mathbb{H}$ is high or infinite, it might be beneficial to rephrase (24) exploiting the fact that, by construction, $K(\boldsymbol{x}_1, \boldsymbol{x}_2) = \langle\psi(\boldsymbol{x}_1), \psi(\boldsymbol{x}_2)\rangle_{\mathbb{H}}$ for all $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathbb{R}^d$. In particular, first notice that $\boldsymbol{v}$ can be always decomposed as $\boldsymbol{v} = \boldsymbol{v}^\perp + \sum_{i=1}^n c_i\boldsymbol{\chi}_i$, where $\boldsymbol{v}^\perp$ is orthogonal to $\operatorname{span}\langle\boldsymbol{\chi}_1, \ldots, \boldsymbol{\chi}_n\rangle$, and this eventually implies

$$\boldsymbol{v}^T\boldsymbol{\chi}_\ell = \left(\sum_{i=1}^n c_i\boldsymbol{\chi}_i\right)^T\boldsymbol{\chi}_\ell = \sum_{i=1}^n c_i\left\langle\boldsymbol{\chi}_i, \boldsymbol{\chi}_\ell\right\rangle_{\mathbb{H}} = \sum_{i=1}^n c_i K(\boldsymbol{x}_i, \boldsymbol{x}_\ell)$$

Letting $\boldsymbol{k}_i := [K(\boldsymbol{x}_i, \boldsymbol{x}_1), \ldots, K(\boldsymbol{x}_i, \boldsymbol{x}_n)]^T$ and $\boldsymbol{c} := [c_1, \ldots, c_n]^T$, (24) becomes

$$\max_{\omega\in\mathbb{R},\boldsymbol{c}\in\mathbb{R}^n} \quad \omega$$
$$\text{s.t.} \begin{cases} \omega & \geq 0 \\ y_i y_j\left(\boldsymbol{k}_i^T\boldsymbol{c} - \boldsymbol{k}_j^T\boldsymbol{c}\right) & \leq 0 \quad \text{if } y_j = y_i,\ j \neq i \\ y_i y_j\left(\boldsymbol{k}_i^T\boldsymbol{c} - \boldsymbol{k}_j^T\boldsymbol{c}\right) + \omega & \leq 0 \quad \text{otherwise.} \end{cases} \tag{25}$$

The LP (25) can then be used to check the discardability of $(\boldsymbol{x}_i, y_i)$ by simply constructing the corresponding simplex pivot table and performing Step 3 in [32, p. 47] on it.

Finally, notice that whether it is better to use formulation (24) or (25) depends on the dimensionality of the input and feature spaces as well as the dataset size.

VII. Numerical Experiments

We consider illustrative linearly separable datasets reduction problems, and numerically compare SVM-PSV with the heuristics offered in [41] and in [12], and with Alg. 2 as proposed in [31].

In general, in dataset reduction problems the aim is to reduce as much as possible the size of the current dataset without reducing the generalization capabilities, considering that new data may become available in the future. As discussed previously, under separability assumptions only PSVs should to be retained: keeping discardable data only increases computational complexity of possible successive trainings.

Description of the datasets

We consider the Iris dataset [42], available at http://archive.ics.uci.edu/ml/datasets/Iris, and other 5 synthetic datasets.

The Iris dataset collects measured characteristics of the flowers of three particular Iris species (*setosa*, *virginica* and *versicolor*). Input space has dimension $d = 4$, since for each flower four inputs are measured. The dataset consists of 50 samples for each class: the data belonging to the first class (*setosa*) are separable with respect to the data belonging to the other two classes. We thus consider the two datasets **Iris12** and **Iris13**, the first containing samples of *Iris setosa* and *Iris virginica*, and the second containing samples of *Iris setosa* and *Iris versicolor*.

The synthetic datasets we use are denoted with the prefix **synt**. **synt#1** has $n = 50$ data points and the inputs belong to $\mathbb{R}^2$, i.e., $d = 2$. Inputs of the positive outputs are sampled from $\mathcal{N}([1,\ 1], \left[\begin{smallmatrix} 0.1 & 0.025 \\ 0.025 & 0.1 \end{smallmatrix}\right])$, i.i.d., while the ones of the negative outputs are sampled from $\mathcal{N}([0,\ 0], \left[\begin{smallmatrix} 0.1 & -0.02 \\ -0.02 & 0.05 \end{smallmatrix}\right])$, i.i.d. Separability of **synt#1** is verified a-posteriori, eventually discarding examples leading to non-separable datasets and resampling.

We also consider other 4 synthetic datasets, denoted with **synt#2.d**, with $d = 2, 3, 4, 5$ indicating also their input location dimensionality. These datasets are generated as follows: consider the hyperplane $(\boldsymbol{w}, b)$, $\boldsymbol{w} = [1, \ldots, 1]^T \in \mathbb{R}^d$, $b = -1$, then generate inputs with the uniform probability on $[0,\ 1]^d$, i.i.d., and classify them as positive or negative depending on their signed distance from the hyperplane $(\boldsymbol{w}, b)$.

Three dataset reduction heuristics

Katagiri and Abe proposed in [41] the following geometric intuition: if a point is either surrounded by other points of the same class or very far from the points of the opposite class then it is not likely to become a SV. Hence, the procedure in [41] first trains the SV classifier on the current dataset. Then, it discards the data that lie inside two suitably defined regions, one per each class, and each corresponding, in the separable case, to the union of:

- an hypersphere that has the same center and ρ times the radius of the minimum-volume hypersphere enclosing all the data of the given class ($\rho \in [0,\ 1]$);
- an hypercone having its vertex at the center of the previous hypersphere, its axis orthogonal to the OSH, opening towards the data of the corresponding class and with its surface forming an angle of $\theta \in [0,\ 180]$ degrees with the OSH (cf. Fig. 4).

Informally, larger ρ's and smaller θ's imply larger dataset reductions *and* larger risk to discard PSVs.

Fig. 4 shows an example of these regions obtained with $\rho = 0.5$ and $\theta = 10°$ on an instance of **synt#2.2**, with $n = 50$. In Fig. 4, points inside the shaded regions are discarded. In the following we will refer to this heuristic as to SVM-KA, KA being the initials of authors Katagiri and Abe.

In [12] instead authors perform dataset reduction without requiring preceding training steps. This is beneficial in applications with large amount of data, where training might be computationally demanding or even infeasible. In particular, the algorithm proposed in [12] starts clustering the training set using k-means, a low computational-complexity unsupervised clustering technique [39, Sec. 13.2.1]. Then it substitutes every cluster that contains data of an unique class with an example, positioned in the centroid of the cluster. Fig. 5 shows the outcome of this algorithm for the same experiment of Fig. 4. For instance, in Fig. 5 all the inputs of cluster 5 are associated

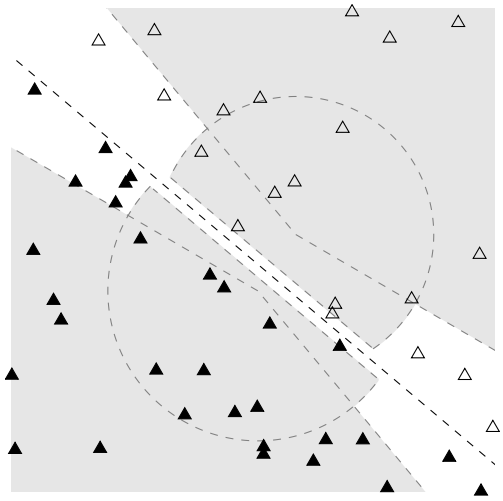

Figure 4. Illustration of the heuristic SVM-KA proposed in [41] with $\rho = 0.5$ and $\theta = 10$. The gray zone indicates the region where data are discarded. The dataset $\mathcal{D}$ is a particular instance of **synt#2.2**, with $n = 50$.

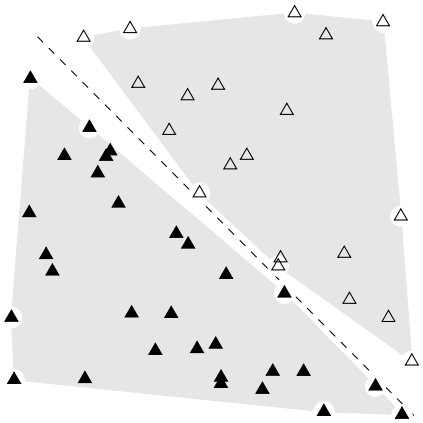

Figure 6. SVM-CH heuristic described in Alg. 2. The gray zone indicates the region where data are discarded. The dataset $\mathcal{D}$ is the same of Fig. 4.

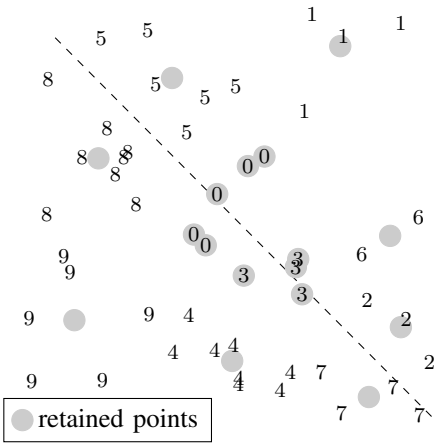

Figure 5. Illustration of the heuristic SVM-KM proposed in [12] with $k = 10$. Numbers indicate the labels assigned by the k-means classification step. The dataset $\mathcal{D}$ is the same of Fig. 4.

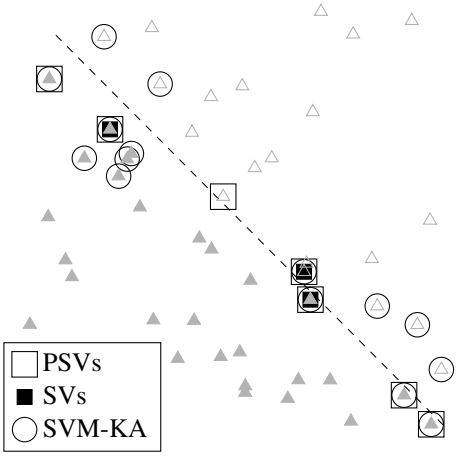

Figure 7. Comparison of the outcome of SVM-PSV with the one of the SVM-KA heuristic in [41] with $\rho = 0.5$, $\theta = 10°$, and the same dataset of Fig. 4.

to the output -1. These data are then discarded and substituted with the centroid of cluster 5. As done in [12], we will refer to this heuristic as to SVM-KM, KM recalling k-means.

We also consider the convex-hulls based data-discarding Alg. 2. Fig. 6 shows the two convex hulls for the same experiment of Fig. 4. In the following we will refer to this heuristic as to SVM-CH, CH standing for Convex Hull.

Figures 7, 8 and 9 compare the outcomes of SVM-PSV with the SVM-KA, SVM-KM and SVM-CH heuristics respectively.

In the example of Fig. 7 SVM-KA erroneously discards a PSV but does not discard any SV, and this because its starting point is an already trained SVM.

Fig. 8 shows instead that, for the same example, SVM-KM erroneously discards a PSV and a SV. We notice that there exist no intuitive relations between the number of clusters k, the dataset reduction properties and the risks to discard PSVs and SVs.

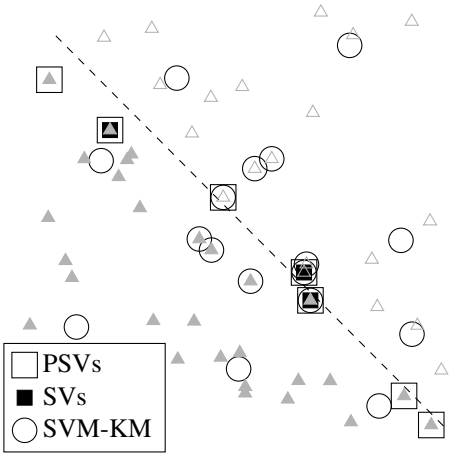

Figure 8. Comparison of the outcome of SVM-PSV with the one of the SVM-KM heuristic in [12] with $k = 10$ and the same dataset of Fig. 4.

	dataset properties			SVM-PSV	SVM-KA [41]			SVM-KM [12]			SVM-CH		
	n	number of SVs	number of PSVs	reduction	reduction	discarded SVs	discarded PSVs	reduction	discarded SVs	discarded PSVs	reduction	discarded SVs	discarded PSVs
synt# 1	50	2.75	5.64	88.7%	72.0%	0	0.39 (6.1%)	77.3%	1.90 (69.0%)	4.55 (77.3%)	71.8%	0	0
synt# 2.2	50	3.00	4.74	90.5%	81.4%	0	0.46 (8.6%)	56.5%	0.65 (21.7%)	1.06 (22.7%)	70.0%	0	0
synt# 2.3	100	3.99	8.46	91.5%	66.3%	0	0.72 (8.3%)	58.6%	0.55 (13.8%)	1.35 (16.1%)	59.1%	0	0
synt# 2.4	200	4.91	12.27	93.9%	57.9%	0	1.15 (9.0%)	72.2%	0.58 (11.7%)	2.33 (18.6%)	55.8%	0	0
synt# 2.5	400	5.91	20.07	95.0%	53.9%	0	1.93 (9.6%)	81.6%	0.70 (11.9%)	5.36 (25.5%)	51.3%	0	0
Iris 12	100	3.00	37.00	63.0%	88.0%	0	26.00 (70.3%)	90.0%	3.00 (100.0%)	37.00 (100.0%)	50.0%	0	0
Iris 13	100	3.00	36.00	64.0%	89.0%	0	28.00 (77.8%)	90.0%	3.00 (100.0%)	36.00 (100.0%)	51.0%	0	0

Table I
SUMMARY OF THE NUMERICAL COMPARISONS OF THE DATASET REDUCTION PERFORMANCES OF ALG. 1, OF THE HEURISTIC PROPOSED IN [41], AND OF THE HEURISTIC PROPOSED IN [12]. FOR EACH SYNTHETIC DATASET, THE REPORTED DATA CORRESPOND TO THE AVERAGE PERFORMANCE OBTAINED CONSIDERING 100 INDEPENDENT EXPERIMENTS.

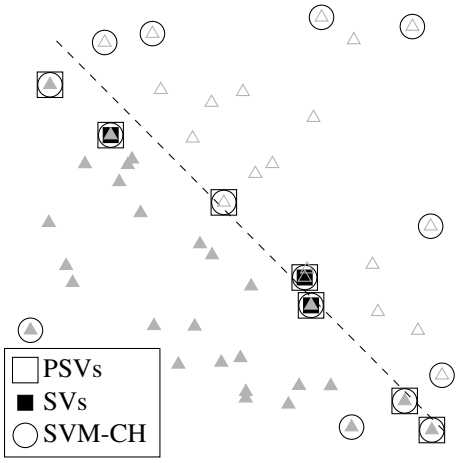

Figure 9. Comparison of the outcome of SVM-PSV with the one of the SVM-CH heuristic described in Alg. 2 on the same dataset of Fig. 4.

Finally, Fig. 9 shows that SVM-CH never discards any PSV but does retain discardable points.

Analysis of the results

We report in Tab. I some experimental results on the dataset reduction capabilities of SVM-PSV, of the SVM-KA heuristic ($\rho = 0.5$, $\theta = 0$, as suggested by the authors), of the SVM-KM heuristic ($k = 10$) and of the SVM-CH heuristic.

We recall that the dataset reduction capabilities are defined as the number of discarded data over the cardinality of the dataset. For the synthetic datasets, results are the averages over 100 independent experiments. For each dataset the table reports: its size n, the actual number of SVs and the number of PSVs, the dataset reduction performance and the number of SVs and PSVs discarded by the various algorithms. In the following

The results presented lead to the following considerations:

- SVM-PSV provides the largest possible dataset reduction performances without information loss, therefore it can be used as a benchmark for comparing different heuristics as the SVM-KA, SVM-KM and SVM-CH.
- SVM-KA heuristic performs satisfactorily, especially for low dimensional dataset. In fact it rarely discards PSVs and it retains a fairly limited number of discardable points. However the performances get worse significantly as the inputs dimensionality increases. For datasets **Iris12** and **Iris13** it achieves a larger reduction rate than the one of SVM-PSV, obviously at the price of a larger discard of of PSVs.
- SVM-KM heuristic seems less effective than the other two methods, and discards more often PSVs and SVs. Nonetheless, and oppositely to SVM-KA, its performances tend to increase with the dimensionality of the inputs. Its data reduction performances on the real datasets **Iris12** and **Iris13** are similar to the ones of SVM-KA.
- In agreement with its theoretical characterization, the SVM-CH heuristic retains all the PSVs, leading to no information losses. It nonetheless retains several discardable points and therefore exhibits a reduce compression capability which further degrades as the dimension of the input space increases. We also notice that SVM-CH compression capabilities are always poorer as compared to SVM-KA.
- The datasets **Iris12**, **Iris13** retain a much larger percentage of PSVs as compared to the synthetic dataset **synt#2.4**, which has the same input dimension $d = 4$. This means that the geometry of **Iris12**, **Iris13** is more complicated than the one of **synt#2.4** therefore, to avoid information losses, they require to retain a larger amount of information.
- Finally, we recall that SVM-KA requires a preceding SVM training step while SVM-KM, SVM-PSV and SVM-CH do not.

VIII. Heuristics for Non Separable Datasets

In the previous sections we shown that under separability assumptions all the DVs can be discarded without affecting the generalization capabilities of the SV classifier. In non-separable cases the classification problem considers as the optimal classification rule the one that solves the convex optimization problem [39, Sec. 12.2]

$$\begin{aligned}(\boldsymbol{w}^*, b^*) := \quad & \arg \min_{\boldsymbol{w},b,\{\chi_i\}} \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C\sum_{i=1}^{n}\chi_i \\ & \text{s.t. } \chi_i \geq 0 \\ & \textit{s.t. } y_i\left(\boldsymbol{w}^T\boldsymbol{x}_i + b\right) \geq 1-\chi_i, \quad i=1,\ldots,n\,.\end{aligned} \tag{26}$$

In this case it can be easily shown that any $(\boldsymbol{x}_i, y_i) \in \mathcal{D}$ can become a SV, i.e., $\text{PSV}(\mathcal{D}) = \mathcal{D}$. In other words, without additional assumptions it is not possible to discard samples guaranteeing no information losses.

Nonetheless, Alg. 1 suggests the following dataset reduction heuristic, formalized in Alg. 3: a) train a SV classifier through (26) on the original dataset $\mathcal{D}$, b) a obtain the separable sub-dataset $\mathcal{D}_\text{s} := \mathcal{D}\backslash\text{SV}(\mathcal{D})$, i.e., let $\mathcal{D}_\text{s}$ be $\mathcal{D}$ without its SVs, c) remove from $\mathcal{D}$ the DVs of $\mathcal{D}_\text{s}$.

Algorithm 3 (SVM-PSVh) Potential Support Vector Inspired Heuristic

1: evaluate the dataset separability using Lemma 14
2: **if** $\mathcal{D}$ is separable **then**
3: compute $\text{PSV}(\mathcal{D})$ with Alg. 1 and then discard $\text{DV}(\mathcal{D})$
4: **else**
5: train the SV classifier on $\mathcal{D}$ solving (26), then compute $\text{SV}(\mathcal{D})$
6: set $\mathcal{D}_\text{s} := \mathcal{D}\backslash\text{SV}(\mathcal{D})$
7: compute $\text{PSV}(\mathcal{D}_\text{s})$ with Alg. 1
8: retain $\text{SV}(\mathcal{D}) \cup \text{PSV}(\mathcal{D}_\text{s})$.
9: **end if**

Numerical Experiments

We compare Alg. 3 (SVM-PSVh) and the two previously analyzed algorithms SVM-KM and SVM-KA on three real non-separable datasets:

1) UCI `Skin / NonSkin` (http://archive.ics.uci.edu/ml/datasets/Skin+Segmentation, 3 features), from which we randomly extract $n_\text{train} = 10^4$ features for training and $n_\text{test} = 10^4$ for testing;
2) UCI `Chess Endgame` (http://archive.ics.uci.edu/ml/datasets/Chess+%28King-Rook+vs.+King%29, 6 features), from which we randomly extract $n_\text{train} = 5\cdot 10^3$ features for training and $n_\text{test} = 5\cdot 10^3$ for testing. The dataset contains 18 classes representing the depth-of-win of chess endgame. In our simulations we classify "ten", "eleven" and "twelve" against "fourteen", "fifteen","sixteen" and "draw";
3) `CodRNA` (http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html, 8 features), from which we randomly extract $n_\text{train} = 7\cdot 10^3$ features for training and $n_\text{test} = 5\cdot 10^3$ for testing.

On each dataset we perform 20 Monte Carlo runs of the following experiment:

1) divide the training set in chunks of 10^3 samples;
2) train four SV classifiers using a 2nd-order polynomial kernels and a regularization parameter $C = 1$, choices which lead to satisfactory prediction capabilities. The first is trained on the full dataset. The second, third and fourth are instead iteratively retrained on the reduced datasets that are obtained using respectively the three heuristics SVM-PSVh, SVM-KM and SVM-KA;
3) test the generalization capabilities of the final Support Vector Classifier (SVC) on the test set.

The results are summarized in the following Figures 10 - 15. In particular, Fig. 11 summarizes how efficiently the SVM-PSVh, SVM-KA and SVM-KM heuristics compress the `Skin / NonSkin` dataset: the figure in fact plots how the size of the retained datasets evolved after the various iterations and over the 20 Monte Carlo runs. The results show that SVM-KA achieves a better compression with respect to SVM-KM and SVM-PSVh outperforms both the other heuristics.

Fig. 10 sketches the empirical distribution of the test errors on the `Skin / NonSkin` dataset. In this case, despite retaining the largest amount of data, SVM-KM gives the largest errors. SVM-KA and SVM-PSVh have instead similar generalization-performance degradations.

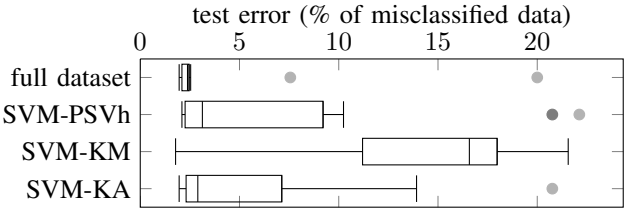

Figure 10. Empirical distribution of the test errors achieved by the SVC strategies full dataset, SVM-PSVh and SVM-KM on 20 Monte Carlo runs on the `Skin / NonSkin` dataset.

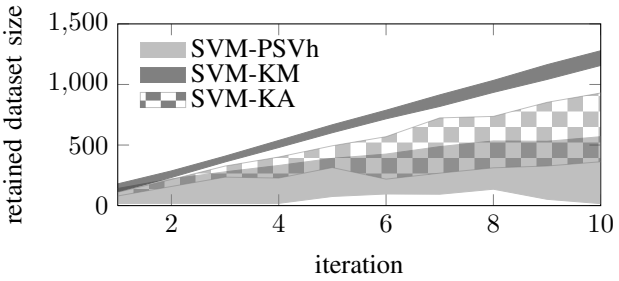

Figure 11. Efficiency in compressing the training set relative to the `Skin / NonSkin` dataset. The solid areas indicate 90% confidence intervals.

Similarly, Fig. 13 and Fig. 12 illustrate data reduction and generalization capabilities of the three proposed heuristics on the `Chess Endgame` dataset. Here SVM-KM has almost no reduction capabilities. Thus, since it retains almost all the examples, its generalization performance are identical to the classifier trained on the whole training set. SVM-PSVh instead

achieves a significant data reduction with almost no performance degradation. The fact that the test set-error distribution is very close to the one achieved using the full dataset indicates that, on this dataset, the SVM-PSVh heuristic successfully retains all the relevant information. Finally, using the SVM-KA heuristic one obtains a larger dataset reduction, but at the cost of a visible generalization performance degradation.

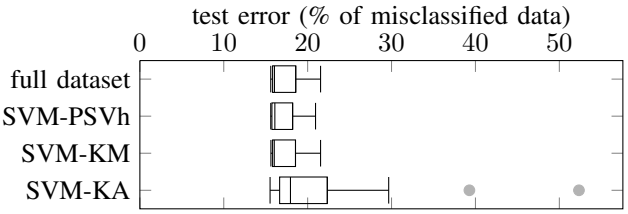

Figure 12. Empirical distribution of the test errors achieved by the SVC strategies full dataset, SVM-PSVh and SVM-Km on 20 Monte Carlo runs on the `Chess Endgame` dataset.

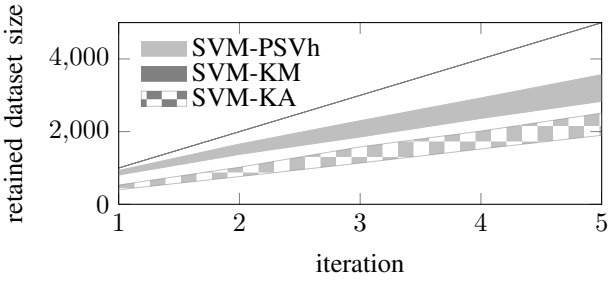

Figure 13. Efficiency in compressing the training set relative to the `Chess Endgame` dataset. The solid areas indicate 90% confidence intervals.

Finally, Fig. 14 and Fig. 15 illustrate data reduction and generalization capabilities of the three considered heuristics on the `CodRNA` dataset. Also in this case SVM-KM retains almost the whole training set. SVM-PSVh and SVM-KA instead achieve similar data reductions, with SVM-PSVh having slightly better generalization performance.

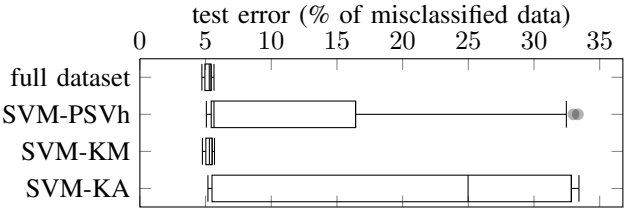

Figure 14. Empirical distribution of the test errors achieved by the SVC strategies full dataset, SVM-PSVh and SVM-Km on 20 Monte Carlo runs on the `CodRNA` dataset.

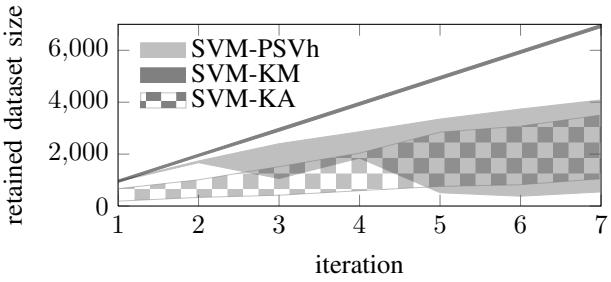

Figure 15. Efficiency in compressing the training set relative to the `CodRNA` dataset. The solid areas indicate 90% confidence intervals.

IX. Conclusions

We have considered the problem of assessing if an element of a training set can become a support vector when new data become available. Under separability assumptions, possibly satisfied also using a suitable feature map, we have fully answered this question formalizing the notions of Potential Support Vectors and Discardable Vectors, and characterizing them via necessary and sufficient conditions.

Geometrical and analytical intuitions underlying the concept of discardability have been provided. In particular, it has been shown that it is possible to check if an example does not bring information in the future just verifying if a certain linear program is unbounded by building a simplex tableau. The algorithm compares favorably with other well known heuristics used to reduce the training set size in synthetic and real-world datasets. In addition, we have also proposed an heuristic based on PSV concepts for classification problems involving nonseparable datasets. Simulations reveal that also in this scenario the approach can be effective for data reduction purposes.

Appendix

Proof (of Prop. 5)

(a) $\mathcal{C} \subset \operatorname{int}(\mathcal{V}_i) \Rightarrow (\boldsymbol{x}_i, y_i) \in \mathrm{DV}(\mathcal{D})$: we start noticing that $\mathcal{C} \subset \operatorname{int}(\mathcal{V}_i) \Rightarrow \mathcal{C} \cap \mathcal{B}_i = \emptyset$. This implies that $(\boldsymbol{x}_i, y_i)$ is not a SV, since $(\boldsymbol{x}_i, y_i)$ is a SV if and only if $(\boldsymbol{w}^*, b^*) \in \mathcal{B}_i$. Since adding new future data cannot lead to expansions of the version space $\mathcal{C}$, this eventually implies $(\boldsymbol{x}_i, y_i) \in \mathrm{DV}(\mathcal{D})$ (cf. Fig. 1).

(b) $(\boldsymbol{x}_i, y_i) \in \mathrm{DV}(\mathcal{D}) \Rightarrow \mathcal{C} \subset \operatorname{int}(\mathcal{V}_i)$: assume *ab absurdo* that $(\boldsymbol{x}_i, y_i) \in \mathrm{DV}(\mathcal{D})$ implies that $\mathcal{C} \subseteq \operatorname{int}(\mathcal{V}_i)$, or equivalently, that $\mathcal{C} \cap \mathcal{B}_i \neq \emptyset$. Consider then any $(\overline{\boldsymbol{w}}, \overline{b}) \in \mathcal{C} \cap \mathcal{B}_i$. $(\overline{\boldsymbol{w}}, \overline{b})$ which must exists because, as just stated, this intersection is not empty. The proof then proceeds showing that since this $(\overline{\boldsymbol{w}}, \overline{b})$ exists then it is possible to add opportune data to $\mathcal{D}$ that make $(\boldsymbol{x}_i, y_i)$ a SV, absurd.

To show this, we consider that the vector $(\overline{\boldsymbol{w}}, \overline{b})$ has two properties:

- $(\overline{\boldsymbol{w}}, \overline{b}) \neq (\boldsymbol{w}^*, b^*)$, where $(\boldsymbol{w}^*, b^*)$ is the OSH for $\mathcal{D}$ since $(\overline{\boldsymbol{w}}, \overline{b}) = (\boldsymbol{w}^*, b^*)$ would imply that $(\boldsymbol{x}_i, y_i)$ is a SV which is in contradiction with the hypothesis that $(\boldsymbol{x}_i, y_i)$ is discardable.

Algorithm 4 (Unboundedness check) (adapted from [32, Ch. 3])

1: write LP (18) in standard form, i.e., as

$$\max_{\boldsymbol{w}'} \ \boldsymbol{c}^T\boldsymbol{w}' \quad \text{s.t.} \ \begin{cases} A\boldsymbol{w}' = \boldsymbol{b} \\ \boldsymbol{w}' \geq \boldsymbol{0} \end{cases} \tag{27}$$

for opportune $A = [\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n], \boldsymbol{b}, \boldsymbol{c}, \boldsymbol{w}'$;

2: consider that particular $\boldsymbol{w}'_0$ corresponding to $(\boldsymbol{w}, \omega) = (\boldsymbol{0}, 0)$ (that, by construction, is a basic feasible solution of 27). For notational simplicity, let it be $\boldsymbol{w}'_0 = (w'_1, \ldots, w'_m, 0, \ldots, 0)$, so that the corresponding simplex pivot tableau corresponding to the constraints $A\boldsymbol{w}' = \boldsymbol{b}$ is

$\boldsymbol{a}_1$	$\ldots$	$\boldsymbol{a}_m$	$\boldsymbol{a}_{m+1}$	$\ldots$	$\boldsymbol{a}_n$	b
1	$\ldots$	0	$\psi_{1,m+1}$	$\ldots$	$\psi_{1,n}$	w'_1
	$\ddots$		$\vdots$		$\vdots$	$\vdots$
0	$\ldots$	1	$\psi_{m,m+1}$	$\ldots$	$\psi_{m,n}$	w'_m

(28)

with the ψ's satisfying

$$\boldsymbol{a}_q = \psi_{1,q}\boldsymbol{a}_1 + \ldots + \psi_{m,q}\boldsymbol{a}_m; \tag{29}$$

3: let $r_j := c_j - \sum_{i=1}^m c_i\psi_{i,j}$;

4: **if** $r_j \geq 0$ for all $j = m+1, \ldots, n$ **then**
$(\boldsymbol{w}, \omega) = (\boldsymbol{0}, 0)$ is the optimal solution, i.e., $\omega^* = 0$;

5: **else**
the problem is unbounded.

6: **end if**

- $\overline{\boldsymbol{w}}^T\boldsymbol{x}_i + \overline{b} = y_i$, because $\left(\overline{\boldsymbol{w}}, \overline{b}\right) \in \mathcal{B}_i$.

Consider now the new vector $(\boldsymbol{x}^F, y^F)$ s.t.

$$\boldsymbol{x}^F = \boldsymbol{x}_i - \left(2\frac{y_i}{\|\overline{\boldsymbol{w}}\|_2^2}\right)\overline{\boldsymbol{w}}, \qquad y^F = -y_i \ . \tag{30}$$

It is then immediate to check that $\overline{\boldsymbol{w}}^T\boldsymbol{x}^F + \overline{b} = y^F$, and thus that $(\overline{\boldsymbol{w}}, \overline{b})$ is the OSH for $\overline{\mathcal{D}} = \left\{(\boldsymbol{x}_i, y_i), (\boldsymbol{x}^F, y^F)\right\}$. Up to this point we thus have two datasets: the original one, $\mathcal{D}$, for which $(\boldsymbol{x}_i, y_i)$ is a DV, and $\overline{\mathcal{D}}$, for which the same $(\boldsymbol{x}_i, y_i)$ is a SV.

Let $\mathcal{C}_{\overline{\mathcal{D}}}$ be the version space, i.e., the set of hyperplanes correctly separating $\overline{\mathcal{D}}$ defined in (12). Then $\mathcal{C}_{\overline{\mathcal{D}}} \cap \mathcal{C}$ is the set of hyperplanes that correctly separate the dataset $\mathcal{D} \cup \overline{\mathcal{D}} = \mathcal{D} \cup \left(\boldsymbol{x}^F, y^F\right)$ ($(\boldsymbol{x}_i, y_i)$ is shared by the two datasets). Notice that, by construction, $\left(\overline{\boldsymbol{w}}, \overline{b}\right) \in \mathcal{C}_{\overline{\mathcal{D}}}$ and also $\left(\overline{\boldsymbol{w}}, \overline{b}\right) \in \mathcal{C}$, and this implies that $\left(\overline{\boldsymbol{w}}, \overline{b}\right) \in \mathcal{C}_{\overline{\mathcal{D}}} \cap \mathcal{C}$. Importantly, this implies that $\mathcal{D} \cup \left(\boldsymbol{x}^F, y^F\right)$ is linearly separated by $\left(\overline{\boldsymbol{w}}, \overline{b}\right)$.

Consider now the following fact: let A be a generic set of elements, and $B \subseteq A$ one of its subsets. Let $a \in A$ be the optimal element of A under a certain metric. Then $a \in B$ implies that a is also the optimal element of B, under the same metric.

Being now $\left(\overline{\boldsymbol{w}}, \overline{b}\right)$ the OSH for $\overline{\mathcal{D}}$ and setting $A = \mathcal{C}_{\overline{\mathcal{D}}}$, $B = \mathcal{C}_{\overline{\mathcal{D}}} \cap \mathcal{C}$, we thus have that

$$\begin{aligned} \left(\overline{\boldsymbol{w}}, \overline{b}\right) = \arg\min_{(\boldsymbol{w},b)\in\mathcal{C}_{\overline{\mathcal{D}}}} \|\boldsymbol{w}\|_2 \quad \Rightarrow \\ \Rightarrow \quad \left(\overline{\boldsymbol{w}}, \overline{b}\right) = \arg\min_{(\boldsymbol{w},b)\in\mathcal{C}_{\overline{\mathcal{D}}}\cap\mathcal{C}} \|\boldsymbol{w}\|_2 \ . \end{aligned} \tag{31}$$

This means that $\left(\overline{\boldsymbol{w}}, \overline{b}\right)$ is the OSH for the augmented dataset $\mathcal{D} \cup \left(\boldsymbol{x}^F, y^F\right)$. This implies that $(\boldsymbol{x}_i, y_i)$ is a SV for $\mathcal{D} \cup \left(\boldsymbol{x}^F, y^F\right)$, i.e., $(\boldsymbol{x}_i, y_i) \in \text{PSV}(\mathcal{D})$, thus leading to a contradiction with the initial hypothesis that $(\boldsymbol{x}_i, y_i)$ is a DV. ■

Proof (of Cor. 6) If $(\boldsymbol{x}_i, y_i) \in \text{PSV}(\mathcal{D})$, then from Prop. 5 it follows that there exists $\left(\overline{\boldsymbol{w}}, \overline{b}\right) \in \mathcal{C} \cap \mathcal{B}_i$. Then the example $(\boldsymbol{x}_F, y_F)$ that satisfies the claim of this corollary can be constructed as shown in the proof of Prop. 5. ■

Proof (of Prop. 8) We assume w.l.o.g. $y_i = +1$. Similar derivations can be performed for the case $y_i = -1$.

(a) $(\boldsymbol{x}_i, y_i) \in \text{PSV}(\mathcal{D}) \Rightarrow \exists$ a quasi-separating $(\boldsymbol{w}, b)$: from Cor. 6 it follows that, if $(\boldsymbol{x}_i, y_i) \in \text{PSV}(\mathcal{D})$ then there exists a future example $\left(\boldsymbol{x}^F, y^F\right)$ s.t. $(\boldsymbol{x}_i, y_i)$ is a SV for $\mathcal{D} \cup \left(\boldsymbol{x}^F, y^F\right)$. If $(\boldsymbol{w}, b)$ is the OSH for $\mathcal{D} \cup \left(\boldsymbol{x}^F, y^F\right)$, then $(\boldsymbol{w}, b)$ is also a separating hyperplane for $\mathcal{D}$. Moreover $(\boldsymbol{x}_i, y_i)$ is a SV for $\mathcal{D} \cup \left(\boldsymbol{x}^F, y^F\right)$, thus

$$\begin{cases} \boldsymbol{w}^T\boldsymbol{x}_i + b = +1 \\ \boldsymbol{w}^T\boldsymbol{x}_j + b \geq +1 \quad \text{if } y_j = +1 \\ \boldsymbol{w}^T\boldsymbol{x}_j + b \leq -1 \quad \text{if } y_j = -1 \ . \end{cases} \tag{32}$$

Consider now the hyperplane $(\boldsymbol{w}, b') = (\boldsymbol{w}, b + y_i)$. It follows that

$$\begin{cases} \boldsymbol{w}^T\boldsymbol{x}_i + b' = 0 \\ \boldsymbol{w}^T\boldsymbol{x}_j + b' \geq 0 \quad \text{if } y_j = +1 \\ \boldsymbol{w}^T\boldsymbol{x}_j + b' \leq -2 \quad \text{if } y_j = -1 \end{cases} \tag{33}$$

i.e., $(\boldsymbol{w}, b')$ quasi-separates $\mathcal{D}$. The proof is then complete considering that $\boldsymbol{w}^T\boldsymbol{x}_j + b' \leq -2 \Rightarrow \boldsymbol{w}^T\boldsymbol{x}_j + b' < 0$, i.e. $\boldsymbol{w}^T\boldsymbol{x}_j + b' \neq 0$.

(b) $\exists$ a quasi-separating $(\boldsymbol{w}, b) \Rightarrow (\boldsymbol{x}_i, y_i) \in \text{PSV}(\mathcal{D})$: for hypothesis it must be $\boldsymbol{w}^T\boldsymbol{x}_j + b < 0$ for all the $\boldsymbol{x}_j \in \mathcal{X}^-$, i.e., for all the data s.t. $y_j = -1$. Letting $c_j := \boldsymbol{w}^T\boldsymbol{x}_j + b$, this hypothesis thus states that $c_j < 0$ for the data with $y_j = -1$.

Consider then

$$\boldsymbol{x}^* := \arg\min_{\boldsymbol{x}_j \in \mathcal{X}^-} |c_j| \ , \tag{34}$$

i.e., the (negative) example $(\boldsymbol{x}^*, -1) \in \mathcal{D}$ that has as its input the one that is the closest to the hyperplane $\boldsymbol{w}^T\boldsymbol{x} + b = 0$. Let then $c^* := \boldsymbol{w}^T\boldsymbol{x}^* + b$, cf. Fig. 16.

The hyperplane $(\boldsymbol{w}, b') = \left(\boldsymbol{w}, b - \frac{c^*}{2}\right)$ is then s.t.

$$\begin{cases} \boldsymbol{w}^T\boldsymbol{x}_j + b' \geq \ -\frac{c^*}{2} \quad \text{if } y_j = +1 \\ \boldsymbol{w}^T\boldsymbol{x}_j + b' \leq c_j - \frac{c^*}{2} \quad \text{if } y_j = -1 \end{cases} \tag{35}$$

and geometrically corresponds to the hyperplane bisecting the space between $(\boldsymbol{w}, b)$ and its translation onto $\left(\boldsymbol{x}^*_j, -1\right)$. Since $-\frac{c^*}{2} > 0$ and $c_j - \frac{c^*}{2} < 0$, $(\boldsymbol{w}, b')$ correctly separates $\mathcal{D}$.

The hyperplane $(\boldsymbol{w}, b'') = (\boldsymbol{w}, b - c^*)$ is instead s.t.

$$\begin{cases} \boldsymbol{w}^T\boldsymbol{x}_j + b'' \geq \ -c^* \quad \text{if } y_j = +1 \\ \boldsymbol{w}^T\boldsymbol{x}_j + b'' \leq c_j - c^* \quad \text{if } y_j = -1 \end{cases} \tag{36}$$

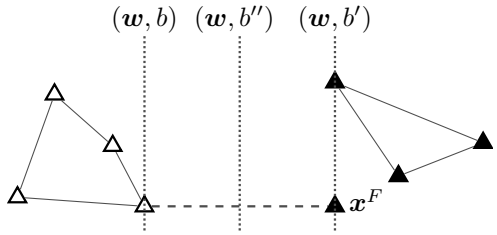

Figure 16. Geometrical interpretation of the quantities involved in the proof of Prop. 8 for the case $d = 2$.

and geometrically corresponds to the translation of $(\boldsymbol{w}, b)$ onto $\left(\boldsymbol{x}_j^*, -1\right)$. Since $-c^* > 0$, $c_j - c^* \leq 0$, and $\boldsymbol{w}^T\boldsymbol{x}^* + b'' = 0$, $(\boldsymbol{w}, b'')$ strictly quasi separates $\mathcal{D}$.

Consider the plausible future example $\left(\boldsymbol{x}^F, -1\right)$, where $\boldsymbol{x}^F := \boldsymbol{x}_i - 2\frac{c_i}{\|\boldsymbol{w}\|_2^2}\boldsymbol{w}$. Geometrically $\boldsymbol{x}^F$ corresponds to the projection of $\boldsymbol{x}_i$ onto $\boldsymbol{w}^T\boldsymbol{x} + b'' = 0$.

Since $(\boldsymbol{w}, b'')$ quasi-separates $\mathcal{D}$, the whole $\text{conv}\left(\left\{\mathcal{X}^- \cup \boldsymbol{x}^F\right\}\right)$ lies in one of the half spaces induced by the hyperplane $\boldsymbol{w}^T\boldsymbol{x} + b'' = 0$. For the same reason, the whole $\text{conv}\left(\{\mathcal{X}^+\}\right)$ lies in one of the half spaces induced by the hyperplane $\boldsymbol{w}^T\boldsymbol{x} + b = 0$.

By construction, the distance between the half space induced by $\boldsymbol{w}^T\boldsymbol{x} + b'' = 0$ that contains $\text{conv}\left(\left\{\mathcal{X}^- \cup \boldsymbol{x}^F\right\}\right)$ and the half space induced by $\boldsymbol{w}^T\boldsymbol{x}+b = 0$ that contains $\text{conv}\left(\{\mathcal{X}^+\}\right)$ is exactly given by the length of the segment connecting $\boldsymbol{x}_i$ and $\boldsymbol{x}^F$. The distance between $\text{conv}\left(\left\{\boldsymbol{x}^F, \boldsymbol{x}_j \text{ s.t. } y_j = -1\right\}\right)$ and $\text{conv}\left(\{\boldsymbol{x}_j \text{ s.t. } y_j = +1\}\right)$ can thus not be smaller than the length of the segment connecting $\boldsymbol{x}_i$ and $\boldsymbol{x}^F$. Since the two points belong respectively to their relative convex hulls, they are thus the extrema of the smallest segment connecting $\text{conv}\left(\left\{\boldsymbol{x}^F, \boldsymbol{x}_j \text{ s.t. } y_j = -1\right\}\right)$ and $\text{conv}\left(\{\boldsymbol{x}_j \text{ s.t. } y_j = +1\}\right)$, i.e. $(\boldsymbol{x}_i, y_i)$ and $(\boldsymbol{x}^F, y^F)$ are the closest points belonging to these two convex hulls.

The convex-hulls based geometrical interpretation of SVCs, see, e.g., [16, Sec. 2], states now that under separability assumptions the OSH bisects the smallest segment connecting the two convex hulls $\text{conv}\left(\{\boldsymbol{x}_j \text{ s.t. } y_j = +1\}\right)$ and $\text{conv}\left(\{\boldsymbol{x}_j \text{ s.t. } y_j = -1\}\right)$. This implies that the OSH for $\mathcal{D} \cup \left(\boldsymbol{x}^F, -1\right)$ is given by a scaled version of $(\boldsymbol{w}, b')$, say $(\alpha\boldsymbol{w}, \alpha b')$ for an opportune not-null $\alpha \in \mathbb{R}$. Being by construction $\boldsymbol{x}^F$ and $\boldsymbol{x}_i$ the closest points to the hyperplane $\alpha\left(\boldsymbol{w}^T\boldsymbol{x} + b'\right) = 0$, $(\boldsymbol{x}^F, y^F)$ and $(\boldsymbol{x}_i, y_i)$ are for sure SVs for the dataset $\mathcal{D} \cup (\boldsymbol{x}^F, y^F)$. This eventually implies $(\boldsymbol{x}_i, y_i) \in \text{PSV}\left(\mathcal{D}\right)$.

■

Proof (of Prop. 9) Prop. 8 states that $(\boldsymbol{x}_i, y_i) \in \text{PSV}\left(\mathcal{D}\right)$ if and only if there exists $(\boldsymbol{w}, b)$ s.t.

$$\begin{cases} y_i\left(\boldsymbol{w}^T\boldsymbol{x}_i + b\right) = 0 \\ y_j\left(\boldsymbol{w}^T\boldsymbol{x}_j + b\right) \geq 0 \quad \text{if } y_i = y_j,\ i \neq j \\ y_j\left(\boldsymbol{w}^T\boldsymbol{x}_j + b\right) > 0 \quad \text{if } y_i \neq y_j. \end{cases} \tag{37}$$

Properly subtracting member to member the various inequalities and the first equality, we obtain

$$\begin{cases} \boldsymbol{w}^T\left(\boldsymbol{x}_i - \boldsymbol{x}_j\right) y_j \leq 0 \quad \text{if } y_i = y_j, i \neq j \\ \boldsymbol{w}^T\left(\boldsymbol{x}_j - \boldsymbol{x}_i\right) y_j < 0 \quad \text{if } y_i \neq y_j\ . \end{cases} \tag{38}$$

In both the cases $y_i = +1$ or $y_i = -1$, given definition (16), (38) can be transformed into system (17). ■

Proof (of Cor. 10) $\boldsymbol{x}_i \in \text{int}\left(\text{conv}\left(\{\boldsymbol{x}_j \mid y_i = y_j\}\right)\right)$ implies that for each direction $\boldsymbol{v} \in \mathbb{R}^d, \|\boldsymbol{v}\|_2 = 1$ there exists an amplitude $\alpha > 0$ s.t. $\boldsymbol{x}_i + \alpha\boldsymbol{v} \in \text{conv}\left(\{\boldsymbol{x}_j \mid y_i = y_j\}\right)$, i.e.,

$$\boldsymbol{x}_i + \alpha\boldsymbol{v} = \sum_{j|y_i=y_j} \lambda_j \boldsymbol{x}_j \tag{39}$$

with $\lambda_j \geq 0$ and $\sum_j \lambda_j = 1$. Since $\boldsymbol{x}_i = \sum_{j|y_i=y_j} \lambda_j \boldsymbol{x}_i$ we have that

$$\alpha\boldsymbol{v} = \sum_{j|y_i=y_j} \lambda_j \left(\boldsymbol{x}_j - \boldsymbol{x}_i\right)\ . \tag{40}$$

Exploiting definition (16) we can write

$$-\boldsymbol{v} = \sum_{j|y_i=y_j} \lambda_j^{'} \Delta_{ij} \tag{41}$$

with $\lambda_j^{'} = \frac{\lambda_j}{\alpha} \geq 0$, thus $-\boldsymbol{v} \in \text{coni}\left(\{\Delta_{ij} | y_i = y_j\}\right)$. Since $\boldsymbol{v}$ is a generic direction in $\mathbb{R}^d$, one has

$$\text{coni}\left(\{\Delta_{ij} | y_i = y_j\}\right) = \mathbb{R}^d \tag{42}$$

and thus that

$$\text{coni}\left(\{\Delta_{ij}\}\right) = \mathbb{R}^d\ . \tag{43}$$

We now prove that this implies $(\boldsymbol{x}_i, y_i) \in \text{DV}\left(\mathcal{D}\right)$. Letting $\mathcal{K} := \text{coni}\left(\{\Delta_{ij}\}\right)$, in fact, it holds that

$$\mathcal{K} = \mathbb{R}^d \Leftrightarrow \text{Lin}\left(\mathcal{K}\right) = \mathbb{R}^d \Leftrightarrow \mathcal{K}^\circ = \{0\}\ . \tag{44}$$

Thus

$$\begin{cases} \Delta_{i1}^T\boldsymbol{w} \leq 0 \\ \vdots \\ \Delta_{in}^T\boldsymbol{w} \leq 0 \end{cases} \tag{45}$$

holds only for $\boldsymbol{w} = \boldsymbol{0}$. This implies that condition 2 in Prop. 9 is not satisfied, thus $(\boldsymbol{x}_i, y_i) \in \text{DV}\left(\mathcal{D}\right)$. ■

Proof (of Prop. 11) Assume $y_i = y_j$. Since $\mathcal{D}$ is linearly separable, $\exists\left(\boldsymbol{w}, b\right)$ s.t.

$$y_\ell\left(\boldsymbol{w}^T\boldsymbol{x}_\ell + b\right) \geq +1 \tag{46}$$

for all $\boldsymbol{x}_\ell \in \mathcal{X}$. Let $c := y_i\left(\boldsymbol{w}^T\boldsymbol{x}_i + b\right)$, $c > 0$ because $(\boldsymbol{w}, b)$ correctly separates $\mathcal{D}$. Assume *ab absurdo* that $(\boldsymbol{x}_j, y_j) \in \text{DV}\left(\mathcal{D}\right)$ but $(\boldsymbol{x}_j, y_j) \in \text{PSV}\left(\mathcal{D} \setminus (\boldsymbol{x}_i, y_i)\right)$. Exploiting Prop. 8, $(\boldsymbol{x}_j, y_j) \in \text{PSV}\left(\mathcal{D} \setminus (\boldsymbol{x}_i, y_i)\right)$ implies that there exists $\left(\widetilde{\boldsymbol{w}}, \widetilde{b}\right)$ s.t.

$$\begin{cases} y_j\left(\widetilde{\boldsymbol{w}}^T\boldsymbol{x}_j + \widetilde{b}\right) = 0 & \text{(47a)} \\ y_\ell\left(\widetilde{\boldsymbol{w}}^T\boldsymbol{x}_\ell + \widetilde{b}\right) \geq 0 \qquad y_\ell = y_j, \ell \neq j & \text{(47b)} \\ y_\ell\left(\widetilde{\boldsymbol{w}}^T\boldsymbol{x}_\ell + \widetilde{b}\right) > 0 \qquad y_\ell \neq y_j\ . & \text{(47c)} \end{cases}$$

Since $(\boldsymbol{x}_j, y_j) \in \text{DV}\left(\mathcal{D}\right)$, $\left(\widetilde{\boldsymbol{w}}, \widetilde{b}\right)$ must wrongly classify $\boldsymbol{x}_i$, i.e., it must be $\widetilde{c} := y_i\left(\widetilde{\boldsymbol{w}}^T\boldsymbol{x}_i + \widetilde{b}\right) < 0$. Exploiting the

previous relation, the fact that $y_i = y_j$, (46) and the definitions of c and $\widetilde{c}$, we can write

$$\begin{cases} y_i \left(\dfrac{\boldsymbol{w}^T}{c} \boldsymbol{x}_i + \dfrac{b}{c} \right) = 1 & \text{(48a)} \\ y_i \left(\dfrac{\widetilde{\boldsymbol{w}}^T}{\widetilde{c}} \boldsymbol{x}_i + \dfrac{\widetilde{b}}{\widetilde{c}} \right) = 1 \, . & \text{(48b)} \end{cases}$$

Subtracting (48a) and (48b) term by term we obtain

$$y_i \left(\underbrace{\left(\frac{\boldsymbol{w}}{c} - \frac{\widetilde{\boldsymbol{w}}}{\widetilde{c}} \right)^T}_{=: \, \overline{\boldsymbol{w}}} \boldsymbol{x}_i + \underbrace{\left(\frac{b}{c} - \frac{\widetilde{b}}{\widetilde{c}} \right)}_{=: \, \overline{b}} \right) = 0 \qquad (49)$$

i.e., an hyperplane $\left(\overline{\boldsymbol{w}}, \overline{b}\right)$ passing through $(\boldsymbol{x}_i, 0)$. Dividing now each term of (46) by c, each term of (47a) $-$ (47c) by $-\widetilde{c}$ and then opportunely summing term by term the various equations of the obtained systems we get

$$\begin{cases} y_i \left(\overline{\boldsymbol{w}}^T \boldsymbol{x}_i + \overline{b} \right) = 0 & \\ y_\ell \left(\overline{\boldsymbol{w}}^T \boldsymbol{x}_\ell + \overline{b} \right) \geq c^{-1} & y_\ell = y_j, \ell \neq j \\ y_\ell \left(\overline{\boldsymbol{w}}^T \boldsymbol{x}_\ell + \overline{b} \right) > c^{-1} & y_\ell \neq y_j \, . \end{cases} \qquad (50)$$

This implies $\left(\overline{\boldsymbol{w}}, \overline{b}\right)$ to be an hyperplane quasi separating $\mathcal{D}$, i.e., $(\boldsymbol{x}_i, y_i) \in \mathrm{PSV}(\mathcal{D})$ which is a contradiction. Hence, it must be $(\boldsymbol{x}_j, y_j) \notin \mathrm{DV}(\mathcal{D} \setminus (\boldsymbol{x}_i, y_i))$. The case $y_i \neq y_j$ can be handled using analogous arguments. ■

Proof (of Lemma 12)

(a) $(\boldsymbol{x}_i, y_i) \in \mathrm{PSV}(\mathcal{D}) \Rightarrow$ (18) unbounded: if $(\boldsymbol{x}_i, y_i) \in \mathrm{PSV}(\mathcal{D})$ then $\exists \boldsymbol{w}' \neq \mathbf{0}$ satisfying (17). Considering $\boldsymbol{w}'$ as fixed, problem (18) attains its maximum for some of the Δ_{ij}'s relative to a datum of the other class, i.e., it must be $\omega' = -\max_{j | y_j \neq y_i} \Delta_{ij}^T \boldsymbol{w}'$. But then every $\alpha \boldsymbol{w}'$ with $\alpha > 0$ satisfies (17), and is s.t. (18) attains its maximum at $\alpha\omega'$, thus (18) is in this case unbounded.

(b) $(\boldsymbol{x}_i, y_i) \in \mathrm{DV}(\mathcal{D}) \Rightarrow$ (18) maximized for $\omega = 0$: if $(\boldsymbol{x}_i, y_i) \in \mathrm{DV}(\mathcal{D})$ then system (17) admits no solution. Since feasible solutions of (18) must satisfy constraints $\Delta_{ij}^T \boldsymbol{w} \leq 0$ for $y_i = y_j$, this implies that $\Delta_{ij}^T \boldsymbol{w} \geq 0$ for some $y_i \neq y_j$. It then immediately follows that for problem (18) an optimal situation corresponds to $\Delta_{ij}^T \boldsymbol{w} = 0$ for all j's and $\omega = 0$. ■

Proof (of Prop. 13) Consider the assumption that the DVs are not removed from the dataset. Then the propositions equivalently states that if $\mathcal{D}$ is not linearly separable then every generic point in the dataset is DV. But this follows immediately since in non-linearly separable datasets do not admit hyperplanes satisfying Prop. 8. ■

Proof (of Lemma 14) Consider the constraints in (8). Assuming $\omega \geq 0$, they can be rewritten as

$$y_i \left(\left(\omega \boldsymbol{w} \right)^T \boldsymbol{x}_i + \left(\omega b \right) \right) - \omega \geq 0, \qquad i = 1, \ldots, n. \qquad (51)$$

It follows immediately that if (20) is unbounded then there exist finite $\omega \neq 0$, $\boldsymbol{w} \neq \mathbf{0}$ satisfying (51). I.e., (8) admits at least one feasible solution, meaning that the dataset is linearly separable.

On the contrary, if the dataset is linearly separable then there exists at least one feasible solution of (8). This leads to a couple $\omega \neq 0$, $\boldsymbol{w} \neq \mathbf{0}$ satisfying (51), i.e., to an unbounded (20) since this couple can be multiplied by arbitrary positive constants.

The other co-implication then follows since the considered concepts are dichotomies. ■

REFERENCES

[1] V. N. Vapnik, *The Nature of Statistical Learning Theory*. New York: Springer-Verlag, 1995.

[2] ——, *Estimation of Dependences Based on Empirical Data*. Berlin: Springer-Verlag, 1982.

[3] T. Joachims, "Making large-scale support vector machine learning practical," in *Advances in Kernel Methods*, B. Schölkopf, C. J. C. Burges, and A. Smola, Eds. The MIT Press, 1998.

[4] G. Cauwenberghs and T. Poggio, *Incremental and Decremental Support Vector Machine Learning*. MIT Press, 2000, vol. 13, pp. 409–415.

[5] A. Shilton, M. Palaniswami, D. Ralph, and A. C. Tsoi, "Incremental training of support vector machines," *IEEE Transactions on Neural Networks*, vol. 16, no. 1, pp. 114–131, january 2005.

[6] F. Orabona, C. Castellini, B. Caputo, L. Jie, and G. Sandini, "On-line independent support vector machines," *Pattern Recognition*, vol. 43, no. 4, pp. 1402 – 1412, April 2010.

[7] X. Liang, "An effective method of pruning support vector machine classifiers," *IEEE Transactions on Neural Networks*, vol. 21, no. 1, pp. 26–38, January 2010.

[8] S. Abe and T. Inoue, "Fast training of support vector machines by extracting boundary data," in *Proceedings of the International Conference on Artificial Neural Networks*, vol. 2130, 2001, pp. 308–313.

[9] B. Li, "Distance-based selection of potential support vectors by kernel matrix," in *Advances in Neural Networks - ISNN 2004*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2004, pp. 468–473.

[10] H. Shin and S. Cho, "Fast pattern selection for support vector classifiers," in *Advances in Knowledge Discovery and Data Mining*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2003, vol. 2637.

[11] H. Lei and Q. Long, "Locate potential support vectors for faster sequential minimal optimization," in *IEEE International Conference on Natural Computation*, July 2011, pp. 367–372.

[12] M. Barros de Almeida, A. de Padua Braga, and J. Braga, "SVM-KM: speeding SVMs learning with a priori cluster selection and k-means," in *6th Brazilian symposium on Neural Networks*, november 2000.

[13] S. S. Keerthi, O. Chapelle, and D. DeCoste, "Building support vector machines with reduced classifier complexity," *Journal of Machine Learning Research*, vol. 7, pp. 1493–1515, 2006.

[14] C. J. C. Burges, "Simplified support vector decision rules," in *Proceedings of the 13th International Conference on Machine Learning*, 1996, pp. 71–77.

[15] C. J. C. Burges and B. Schölkopf, "Improving the accuracy and speed of support vector learning machines," in *Proceedings of the 9th NIPS Conference*, 1997, pp. 375–381.

[16] A. Bordes and L. Bottou, "The huller: A simple and efficient online svm," in *Machine Learning: ECML 2005*. Springer Berlin / Heidelberg, 2005, pp. 505–512.

[17] T. Downs, K. E. Gates, and A. Masters, "Exact simplification of support vector solutions," *Journal of Machine Learning Research*, vol. 2, pp. 293–297, December 2001.

[18] X. Liang, R.-C. Chen, and X. Guo, "Pruning support vector machines without altering performances," *IEEE Transactions on Neural Networks*, vol. 19, no. 10, pp. 1792–1803, October 2008.

[19] Y. Engel, S. Mannor, and R. Meir, "Sparse online greedy support vector regression," in *Machine Learning: ECML 2002*. Springer Berlin / Heidelberg, 2002, vol. 2430, pp. 1–3.

[20] T. Kobayashi and N. Otsu, "Efficient reduction of support vectors in kernel-based methods," in *16th IEEE International Conference on Image Processing*, November 2009, pp. 2077–2080.

[21] F. Orabona, J. Keshet, and B. Caputo, "Bounded kernel-based online learning," *Journal of Machine Learning Research*, vol. 10, pp. 2643–2666, 2009.

[22] S. Katagiri and S. Abe, "Incremental training of support vector machines using hyperspheres," *Pattern Recognition Letters*, vol. 27, no. 13, pp. 1495–1507, October 2006.

[23] D. Geebelen, J. A. K. Suykens, and J. Vandewalle, "Reducing the number of support vectors of svm classifiers using the smoothed separable case approximation," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 4, pp. 682–688, 2012.

[24] R. U. Pedersen, "Using support vector machines for distributed machine learning," Ph.D. dissertation, University of Copenhagen, August 2004.

[25] D. A. Tran and T. Nguyen, "Localization in wireless sensor networks based on support vector machines," *IEEE Transactions on Parallel and Distributed Systems*, vol. 19, no. 7, pp. 981–994, July 2008.

[26] W. Kim, J. Park, and H. Kim, "Target localization using ensemble support vector regression in wireless sensor networks," in *IEEE Wireless Communications and Networking Conference*, Sydney, Australia, April 2010.

[27] A. Navia-Vazquez, D. Gutierrez-Gonzalez, E. Parrado-Hernandez, and J. J. Navarro-Abellan, "Distributed support vector machines," *IEEE Transactions on Neural Networks*, vol. 17, no. 4, pp. 1091–1097, July 2006.

[28] D. Wang, J. Li, and Y. Zhou, "Support vector machine for distributed classification: A dynamic consensus approach," in *IEEE Workshop on Statistical Signal Processing*, August 2009.

[29] D. Wang, J. Zheng, Y. Zhou, and J. Li, "A scalable support vector machine for distributed classification in ad hoc sensor networks," *Neurocomputing*, vol. 74, no. 1-3, pp. 394–400, December 2010.

[30] M. Pontil and A. Verri, "Properties of support vector machines," *Neural Computation*, vol. 10, no. 4, pp. 955–974, may 1998.

[31] E. Osuna and O. De Castro, "Convex hull in feature space for support vector machines," in *8th Ibero-American conference on Artificial Intelligence*, 2002, pp. 411–419.

[32] D. G. Luenberger and Y. Ye, *Linear and Nonlinear Programming*, 2nd ed. Springer, 2008.

[33] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 2nd ed. MIT Press and McGraw-Hill, 2001.

[34] B. C. Barber, D. P. Dobkin, and H. Huhdanpaa, "The quickhull algorithm for convex hulls," *ACM Transactions on Mathematical Software*, vol. 22, no. 4, pp. 469–483, December 1996.

[35] D. R. Chand and S. S. Kapur, "An algorithm for convex polytopes," *Journal of the Association for Computing Machinery*, vol. 17, no. 1, pp. 78–86, 1970.

[36] R. A. Jarvis, "On the identification of the convex hull of a finite set of points in the plane," *Information Processing Letters*, vol. 2, no. 1, pp. 18–21, 1973.

[37] D. Elizondo, "The linear separability problem: Some testing methods," *IEEE Transactions on Neural Networks*, vol. 17, no. 2, pp. 330–344, March 2006.

[38] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. Wiley Interscience, 2000.

[39] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, 2nd ed. Springer, 2008.

[40] V. N. Vapnik, *Statistical Learning Theory*. New York: Springer-Verlag, 1998.

[41] S. Katagiri and S. Abe, "Incremental training of support vector machines using truncated hypercones," in *Artificial Neural Networks in Pattern Recognition*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2006, vol. 4087, pp. 153–164.

[42] R. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of Eugenics*, vol. 7, no. 2, pp. 179–188, 1936.