

# ON $\ell_p$ -SUPPORT VECTOR MACHINES AND MULTIDIMENSIONAL KERNELS

VÍCTOR BLANCO<sup>†</sup>, JUSTO PUERTO<sup>‡</sup>, AND ANTONIO M. RODRÍGUEZ-CHÍA<sup>\*</sup>

<sup>†</sup>DPT. QUANTITATIVE METHODS FOR ECONOMICS & BUSINESS, UNIVERSIDAD DE GRANADA  
*E-mail address:* `vblanco@ugr.es`

<sup>‡</sup>DPT. STATISTICS & OR, UNIVERSIDAD DE SEVILLA  
*E-mail address:* `puerto@us.es`

<sup>\*</sup>DPT. STATISTICS & OR, UNIVERSIDAD DE CÁDIZ  
*E-mail address:* `antonio.rodriguezchia@uca.es`

ABSTRACT. In this paper, we extend the methodology developed for Support Vector Machines (SVM) using  $\ell_2$ -norm ( $\ell_2$ -SVM) to the more general case of  $\ell_p$ -norms with  $p \geq 1$  ( $\ell_p$ -SVM). The resulting primal and dual problems are formulated as mathematical programming problems; namely, in the primal case, as a second order cone optimization problem and in the dual case, as a polynomial optimization problem involving homogeneous polynomials. Scalability of the primal problem is obtained via general transformations based on the expansion of functionals in Schauder spaces. The concept of Kernel function, widely applied in  $\ell_2$ -SVM, is extended to the more general case by defining a new operator called multidimensional Kernel. This object gives rise to reformulations of dual problems, in a transformed space of the original data, which are solved by a moment-sdp based approach. The results of some computational experiments on real-world datasets are presented showing rather good behavior in terms of standard indicators such a *accuracy index* and its ability to classify new data.

## 1. INTRODUCTION

In supervised classification, given a finite set of objects partitioned into classes, the goal is to build a mechanism, based on current available information, for classifying new objects into these classes. Due to their successful applications in the last decades, as for instance in writing recognition [1], insurance companies (to determine whether an applicant is a high insurance risk or not) [19], banks (to decide whether an applicant is a good credit risk or not) [15], medicine (to determine whether a tumor is benigne or maligne) [32, 26], etc; support vector machines (SVMs) have become a popular methodology for supervised classification [4].

Support vector machine (SVM) is a mathematical programming tool, originally developed by Vapnik [35, 36] and Cortes and Vapnik [11], which consists in finding a hyperplane to separate a set of data into two classes, so that the distance from the hyperplane to the nearest point of each class is maximized. In order to do that, the standard SVM solves an optimization problem that accounts for both

---

2010 *Mathematics Subject Classification.* 62H30, 90C26, 53A45, 15A60.

*Key words and phrases.* Support Vector Machines, Kernel functions,  $\ell_p$ -norms, Mathematical Programming.

the training error and the model complexity. Thus, if the separating hyperplane is given as  $\mathcal{H} = \{z \in \mathbb{R}^d : \omega^t z + b = 0\}$ , the function to be minimized is of the form  $\frac{1}{2}\|\omega\|^2 + C \cdot R_{emp}(\mathcal{H})$ , where  $\|\cdot\|$  is a norm and  $R_{emp}$  is an empirical measure of the risk incurred using the hyperplane  $\mathcal{H}$  to classify the training data. The most popular version of SVM is the one using the Euclidean norm to measure the distance. This approach allows for the use of a kernel function as a way of embedding the original data in a higher dimension space where the separation may be easier without increasing the difficulty of solving the problem (the so-called *kernel trick*).

After a fruitful development of the above approach, some years later, it was observed by several authors that the model complexity could be controlled by other norms different from the Euclidean one (see [2, 14, 21, 22, 29]). Among many other facts, it is well-known that using  $\ell_1$  or  $\ell_\infty$  norms (as well as any other polyhedral norms) gives rise to SVM whose induced optimization problems are linear rather than quadratic, making, in principle, possible solving larger size instances. Moreover, it is also agreed that  $\ell_1$ -SVM tends to generate sparse classifiers that can be more easily interpreted and reduce the risk of overfitting. The use of more general norms, as the family of  $\ell_p$ ,  $1 < p < +\infty$ , has been also partially investigated [5, 14, 25]. For this later case, some geometrical intuition on the underlying problems has been given but very few is known about the optimization problems (primal and dual approaches), transformation of data, extensions of the kernel tools (that have been extremely useful in the Euclidean case) and about actual applications to classify databases.

The goal of this paper is to develop a common framework for the analysis of  $\ell_p$ -norm Support Vector Machine ( $\ell_p$ -SVM) with general  $p \in \mathbb{Q}$  and  $p > 1$ . We shall develop the theory to understand primal and dual versions of this problem using these norms. In addition, we also extend the concept of kernel as a way of considering data embedded in a higher dimension space without increasing the difficulty of tackling the problem, that in the general case always appears via homogeneous polynomials and linear functions. In our approach, we reduce all the primal problems to efficiently solvable Second Order Cone Programming (SOCP) problems. The respective dual problems are reduced to solving polynomial optimization problems. A thorough geometrical analysis of those problems allows for an extension of the kernel trick, applicable to the Euclidean case, to the more general  $\ell_p$ -SVM. For that extension, we introduce the concept of multidimensional kernel, a mathematical object that makes the above mentioned job. In addition, we derive the relationship between multidimensional kernel functions and real tensors. In particular, we provide sufficient conditions to test whether a symmetric real tensor, of adequate dimension and order, induces one of the above mentioned multidimensional kernel functions. Also, we develop two different approaches to find the separating hyperplanes. The first one is based on the Theory of Moments and it constructs a sequence of SemiDefinite Programs (SDP) that converges to the optimal solution of the problem. The second one uses limited expansions of functionals representable in Schauder spaces [24], and it allows us to approximate any transformation (whose functional belongs to a Schauder space) in the original space without mapping the data. Both approaches permit to reproduce the *kernel trick* even without specifying any a priori transformation. We report the results of an extensive battery of computational experiments, on some common real-world

instances on the SVM field, which are comparable or superior to those previously known in the literature.

The rest of the paper is organized as follows. In Section 2, we introduce  $\ell_p$ -support vector machines. We derive primal and dual formulations for the problem, as a second order cone programming problem and as polynomial optimization problem involving homogeneous polynomials, respectively. In addition, using the dual formulation, we give explicit expressions of the separating hyperplanes expressed as homogeneous polynomials on the original data. In Section 3, the concept of multidimensional Kernel is defined to extend the Kernel theory for  $\ell_2$ -SVM to a more general case of  $\ell_p$ -SVM with  $p > 1$ . A hierarchy of Semidefinite Programs (SDP) that converges to the actual solution of  $\ell_p$ -SVM is developed in Section 4. Finally, in Section 5, the results of some computational experiments on real-world datasets are reported.

## 2. $\ell_p$ -NORM SUPPORT VECTOR MACHINES

For a given  $p \in \mathbb{Q}$  with  $p > 1$ , the goal of this section is to provide a general framework to deal with  $\ell_p$ -SVM. In  $\ell_p$ -SVM, the problem will be formulated as a mathematical programming problem whose objective function depends on the  $\ell_p$ -norm of some of the decision variables. The input data for this problem is a set of  $d$  quantitative measures about  $n$  individuals. The  $d$  measures about each individual  $i \in \{1, \dots, n\}$  are identified with the vector  $\mathbf{x}_i \in \mathbb{R}^d$ , while for  $j \in \{1, \dots, d\}$ , the  $n$  observations about the  $j$ -th measure are denoted by  $\mathbf{x}_{.j} \in \mathbb{R}^n$ . The  $i$ th individual is also classified into a class  $y_i$ , with  $y_i \in \{-1, 1\}$ , for  $i = 1, \dots, n$ . The classification pattern is defined by  $\mathbf{y} = (y_1, \dots, y_n) \in \{-1, 1\}^n$ .

The goal of SVM is to find a hyperplane  $\mathcal{H} = \{z \in \mathbb{R}^d : \omega^t z + b = 0\}$  in  $\mathbb{R}^d$  that minimizes the misclassification of data to their own class in the sense that is explained below. SVM tries to find a band defined by two parallel hyperplanes,  $\mathcal{H}_+ = \{z \in \mathbb{R}^d : \omega^t z + b = 1\}$  and  $\mathcal{H}_- = \{z \in \mathbb{R}^d : \omega^t z + b = -1\}$  of maximal width without misclassified observations. The ultimate aim is that each class belongs to one of the halfspaces determined by the strip. Note that if the data are linearly separable, this constraint can be written as follows:

$$y_i(\omega^t \mathbf{x}_i + b) \geq 1, \quad i = 1, \dots, n.$$

Since in many cases a linear separator is not possible, misclassification is allowed by adding a variable  $\xi_i$  for each individual which will take value 0 if the observation is adequately classified with respect to this strip, i.e., the above constraints are fulfilled for that individual; and it will take a positive value proportional on how far is the observation from being well-classified. (This misclassifying error is usually called the *hinge-loss* of the  $i$ th individual and represents the amount  $\max\{0, 1 - y_i(\omega^t \mathbf{x}_i + b)\}$ , for all  $i = 1, \dots, n$ .) Then, the constraints to be satisfied are:

$$y_i(\omega^t \mathbf{x}_i + b) \geq 1 - \xi_i, \quad \forall i = 1, \dots, n.$$

Therefore, the goal will be simultaneously to maximize the margin between the two hyperplanes,  $\mathcal{H}_+$  and  $\mathcal{H}_-$  and to minimize the deviation of misclassified observations. To measure the norm-based margin between the hyperplanes  $\mathcal{H}_+$  and  $\mathcal{H}_-$ , one can use the results by Mangasarian [27], to obtain that whenever the distance measure is the  $\ell_q$ -norm (with  $\frac{1}{p} + \frac{1}{q} = 1$ ), the margin for  $\ell_p$ -SVM is exactly  $\frac{2}{\|\omega\|_p}$  (Recall that  $\|\omega\|_q$  is the dual norm of  $\|\omega\|_p$ ). Henceforth, we assume without loss of generality that  $q = \frac{r}{s} > 1$ , with  $r, s \in \mathbb{Z}_+$  and  $\gcd(r, s) = 1$ .

Next, for the deviation of misclassified observations, one can take the summation of the slack variables  $\xi_i$  as a measure for that term in the objective function. Thus, the problem of finding the best hyperplane based on the above two criteria can be equivalently modeled with the aggregated objective function  $\|\omega\|_p^p + C \sum_{i=1}^n \xi_i$ , where  $C$  is a parameter of the model representing the tradeoff between the margin and the deviation of misclassified points (weighting the importance given to the correct classification of the observations in the training dataset or to the ability of the model to classify out-sample data).

Hence, the  $\ell_p$ -SVM problem can be formulated as:

$$\begin{aligned}
 (\ell_p\text{-SVM}) \quad \rho^* = \min \quad & \|\omega\|_p^p + C \sum_{i=1}^n \xi_i \\
 \text{s.t.} \quad & y_i(\omega^t x_i + b) \geq 1 - \xi_i, & \forall i = 1, \dots, n, \\
 & \xi_i \geq 0, & \forall i = 1, \dots, n, \\
 & \omega \in \mathbb{R}^d, b \in \mathbb{R}.
 \end{aligned}$$

Observe that the above problem is a convex nonlinear optimization problem which can be efficiently solved using global optimization tools. Actually, it can be formulated as the following convex optimization problem with a linear objective function, a set of linear constraints and a single nonlinear inequality constraint:

$$\begin{aligned}
 (1) \quad & \min \quad t + C \sum_{i=1}^n \xi_i \\
 (2) \quad & \text{s.t.} \quad y_i(\omega^t x_i + b) \geq 1 - \xi_i, & \forall i = 1, \dots, n, \\
 (3) \quad & t \geq \|\omega\|_p^p, \\
 (4) \quad & \xi_i \geq 0, & \forall i = 1, \dots, n, \\
 (5) \quad & \omega \in \mathbb{R}^d, b, t \in \mathbb{R},
 \end{aligned}$$

where constraint  $t \geq \|\omega\|_p^p$  can be conveniently reformulated by introducing new variables  $v_j$  and  $u_j$  to account for  $|\omega_j|$  and  $|\omega_j|^p$ , respectively (note that  $p = \frac{r}{r-s}$ ), for  $j = 1, \dots, d$ :

$$\begin{aligned}
 & v_j \geq \omega_j & \forall j = 1, \dots, d, \\
 & v_j \geq -\omega_j & \forall j = 1, \dots, d, \\
 & t \geq \sum_{j=1}^d u_j, \\
 (6) \quad & u_j^{r-s} \geq v_j^r, & \forall j = 1, \dots, d.
 \end{aligned}$$

Although the above formulation is still nonlinear, constraints in (6) can be efficiently rewritten as a set of second order cone constraints and then solved via interior point algorithms (see [3]).

At this point, we would like to remark that the cases  $p = 1, +\infty$  also fit (with slight simplifications) within the above framework and obviously they both give rise to linear programs that can be solved via standard linear programming tools. For that reason, we do not follow up with their analysis in this paper that focus on

more general problems that fall in the class of conic linear programming, i.e., we assume without loss generality that  $1 < p < +\infty$ .

A second reformulation is also possible using its Lagrangian dual formulation.

**Proposition 2.1.** *The Lagrangian dual problem of ( $\ell_p$ -SVM) can be formulated as a polynomial optimization problem.*

*Proof.* Observe first that ( $\ell_p$ -SVM) is convex and satisfies Slater's qualification constraint therefore it has zero duality gap with respect to the Lagrangian dual problem. Its Lagrangian function is:

$$(LD) \quad L(\omega, b; \alpha, \beta) = \|\omega\|_p^p + C \sum_{i=1}^n \xi_i - \sum_{i=1}^n \alpha_i (y_i(\omega^t x_i + b) + \xi_i - 1) - \sum_{i=1}^n \beta_i \xi_i,$$

where  $\alpha_i$  is the dual variable associated to constraints  $y_i(\omega^t x_i + b) \geq 1 - \xi_i$  and  $\beta_i$  the one for constraints  $\xi_i \geq 0$ , for  $i = 1, \dots, n$ . The KKT optimality conditions for the problem read as:

$$(7) \quad \frac{\partial L}{\partial \omega_j} = p|\omega_j|^{p-1} \text{sgn}(\omega_j) - \sum_{i=1}^n \alpha_i y_i x_{ij} = 0, \quad \forall j = 1, \dots, d,$$

$$(8) \quad \frac{\partial L}{\partial b} = \sum_{i=1}^n \alpha_i y_i = 0,$$

$$(9) \quad \frac{\partial L}{\partial \xi_i} = C - \alpha_i - \beta_i = 0, \quad \forall i = 1, \dots, n,$$

$$\alpha_i, \beta_i \geq 0, \quad \forall i = 1, \dots, n.$$

where  $\text{sgn}(\cdot)$  stands for the sign function.

Hence, applying conditions (8) and (9), we obtain the following alternative expression of (LD):

$$L(\omega, b; \alpha) = \|\omega\|_p^p - \sum_{i=1}^n \alpha_i y_i \omega^t x_i + \sum_{i=1}^n \alpha_i.$$

In addition, from (7) and taking into account that  $\frac{1}{p-1} = q-1$ , we can reconstruct the optimal value of  $\omega_j$  for any  $j = 1, \dots, d$ , as follows:

$$|\omega_j| = \frac{1}{p^{q-1}} \left( \text{sgn}(\omega_j) \sum_{i=1}^n \alpha_i y_i x_{ij} \right)^{q-1},$$

and then,

$$\omega_j = \frac{1}{p^{q-1}} \text{sgn}(\omega_j) \left( \text{sgn}(\omega_j) \sum_{i=1}^n \alpha_i y_i x_{ij} \right)^{q-1}.$$

Observe that the above two expressions are well-defined for any  $q \geq 1$ , because by (7), we have that  $\text{sgn}(\omega_j) \left( \sum_{i=1}^n \alpha_i y_i x_{ij} \right) \geq 0$ .

Actually, by (7) we have that

$$(10) \quad \text{sgn}(\omega_j) = \text{sgn} \left( \sum_{i=1}^n \alpha_i y_i x_{ij} \right) =: \mathcal{S}_{\alpha, j}.$$

Hence:

$$(11) \quad \omega_j = \frac{1}{p^{q-1}} \mathcal{S}_{\alpha,j} \left( \mathcal{S}_{\alpha,j} \sum_{i=1}^n \alpha_i y_i x_{ij} \right)^{q-1}.$$

Therefore, again the Lagrangian dual function can be rewritten as:

$$\begin{aligned} L(\alpha) &= \left( \frac{1}{p^q} \right) \sum_{j=1}^d \left( \left| \sum_{i=1}^n \alpha_i y_i x_{ij} \right|^{q-1} \right)^p - \left( \frac{1}{p^{q-1}} \right) \sum_{i=1}^n \sum_{j=1}^d \alpha_i y_i x_{ij} \mathcal{S}_{\alpha,j} \left( \mathcal{S}_{\alpha,j} \sum_{k=1}^n \alpha_k y_k x_{kj} \right)^{q-1} + \sum_{i=1}^n \alpha_i \\ &= \left( \frac{1}{p^q} \right) \sum_{j=1}^d \left| \sum_{i=1}^n \alpha_i y_i x_{ij} \right|^q - \left( \frac{1}{p^{q-1}} \right) \sum_{j=1}^d \left| \sum_{i=1}^n \alpha_i y_i x_{ij} \right|^q + \sum_{i=1}^n \alpha_i \\ &= \left( \frac{1}{p^q} - \frac{1}{p^{q-1}} \right) \sum_{j=1}^d \left| \sum_{i=1}^n \alpha_i y_i x_{ij} \right|^q + \sum_{i=1}^n \alpha_i \end{aligned}$$

provided that  $\sum_{i=1}^n \alpha_i y_i = 0$  and  $0 \leq \alpha_i \leq C, \forall i = 1, \dots, n$ .

Thus, the Lagrangian dual problem may be formulated as follows:

$$\begin{aligned} (\text{PLD}) \quad & \max \quad \left( \frac{1}{p^q} - \frac{1}{p^{q-1}} \right) \sum_{j=1}^d \left| \sum_{i=1}^n \alpha_i y_i x_{ij} \right|^q + \sum_{i=1}^n \alpha_i \\ & \text{s.t.} \quad \sum_{i=1}^n \alpha_i y_i = 0, \\ & \quad 0 \leq \alpha_i \leq C, \quad \forall i = 1, \dots, n. \end{aligned}$$

Introducing the variables  $\delta_j$  and  $u_j$  to represent  $\left| \sum_{i=1}^n \alpha_i y_i x_{ij} \right|$  and  $\left| \sum_{i=1}^n \alpha_i y_i x_{ij} \right|^q$ , taking into account that the coefficient  $\left( \frac{1}{p^q} - \frac{1}{p^{q-1}} \right)$  is always negative for  $1 < p, q < +\infty$  and considering  $q = \frac{r}{s}$ , the problem above is equivalent to the following polynomial optimization problem:

$$\begin{aligned} (\text{POP}_{\text{LD}}) \quad & \max \quad \left( \frac{1}{p^q} - \frac{1}{p^{q-1}} \right) \sum_{j=1}^d u_j + \sum_{i=1}^n \alpha_i, \\ & \text{s.t.} \quad \delta_j \geq \sum_{i=1}^n \alpha_i y_i x_{ij}, \quad \forall j = 1, \dots, d, \\ & \quad \delta_j \geq - \sum_{i=1}^n \alpha_i y_i x_{ij}, \quad \forall j = 1, \dots, d, \\ & \quad u_j^s \geq \delta_j^r, \quad \forall j = 1, \dots, d, \\ & \quad \sum_{i=1}^n \alpha_i y_i = 0, \\ & \quad 0 \leq \alpha_i \leq C, \quad \forall i = 1, \dots, n. \end{aligned}$$

□

The reader may observe that the problem (POP<sub>LD</sub>) simplifies further for the cases of integer  $q$  ( $q = r$  and  $s = 1$ ), and especially if  $r$  is even, which results in:

$$\begin{aligned} \max \quad & \left( \frac{1}{p^q} - \frac{1}{p^{q-1}} \right) \sum_{j=1}^d \delta_j + \sum_{i=1}^n \alpha_i \\ \text{s.t.} \quad & \delta_j \geq \left( \sum_{i=1}^n \alpha_i y_i x_{ij} \right)^r, \quad \forall j = 1, \dots, d, \\ & \sum_{i=1}^n \alpha_i y_i = 0, \\ & 0 \leq \alpha_i \leq C, \quad \forall i = 1, \dots, n. \end{aligned}$$

In order to extend the kernel theory developed for  $\ell_2$ -norm to a general  $\ell_p$ -norm, now we study an alternative formulation of the Lagrangian dual problem in terms of linear functions and homogeneous polynomials in  $\alpha$ . For this analysis we consider the case where  $q = \frac{r}{s}$  with  $s = 1$ . In order to simplify the proposed formulations we denote by  $H_y = \{\alpha \in [0, C]^n : \sum_{i=1}^n \alpha_i y_i = 0\}$ , the feasible region of (P<sub>LD</sub>) where the dual variables  $\alpha$  belong to.

**Theorem 2.1.** *There exists an arrangement of hyperplanes of  $\mathbb{R}^n$ , such that, in each of its full dimensional elements:*

- i) (P<sub>LD</sub>) can be formulated as a mathematical programming problem with objective function given by a linear term plus a homogeneous polynomial of degree  $r$  on  $\alpha$  and linear constraints.
- ii) (P<sub>LD</sub>) and the separating hyperplane it induces depend on the original data throughout homogeneous polynomials.

*Proof.* Using (10), the first addend of the objective function of (P<sub>LD</sub>) can be rewritten as:

$$\sum_{j=1}^d \left| \sum_{i=1}^n \alpha_i y_i x_{ij} \right|^r = \sum_{j=1}^d \left( \mathcal{S}_{\alpha,j} \sum_{i=1}^n \alpha_i y_i x_{ij} \right)^r.$$

The above is a piecewise multivariate polynomial in  $\alpha$  (recall that  $y$  and  $x$  are input data) with a finite number of “branches” induced by the different signs of the terms  $\mathcal{S}_{\alpha,j}^r$  for all  $j = 1, \dots, d$ . Each branch is obtained fixing arbitrarily  $\mathcal{S}_{\alpha,j}^r$  to +1 or -1. The domains of these branches are defined by the arrangement induced by the set of homogeneous hyperplanes  $\{\sum_{i=1}^n \alpha_i y_i x_{ij} = 0, j = 1, \dots, d\}$ . It is well-known that this arrangement has  $O(2^d)$  full dimensional subdivision elements that we shall call *cells*, see [13]; all of them pointed, closed, convex cones. Since a generic cell is univocally defined by the signs of the expressions  $\sum_{i=1}^n \alpha_i y_i x_{ij}$  for  $j = 1, \dots, d$ , denote by  $\mathcal{C}(s_1, \dots, s_d) = \{\alpha \in \mathbb{R}^n : \mathcal{S}_{\alpha,j} = s_j, j = 1, \dots, d\}$ , with  $s_j \in \{-1, 1\}$  for all  $j = 1, \dots, d$ . Next, for all  $\alpha \in \mathcal{C}(s_1, \dots, s_d)$  the signs are constant and this allows us to remove the absolute value in the expression of the first addend of the objective function of (P<sub>LD</sub>) and then to rewrite it as sum of monomials of the same degree. Indeed, denoting by  $z^\gamma := z_1^{\gamma_1} \cdots z_n^{\gamma_n}$ , for all  $z = (z_1, \dots, z_n) \in \mathbb{R}^n$  and

$\gamma = (\gamma_1, \dots, \gamma_n) \in \mathbb{N}^n$ , we have the following equalities for any  $\alpha \in \mathcal{C}(s_1, \dots, s_d)$ :

$$\sum_{j=1}^d \left| \sum_{i=1}^n \alpha_i y_i x_{ij} \right|^r = \sum_{j=1}^d \left( s_j \sum_{i=1}^n \alpha_i y_i x_{ij} \right)^r = \sum_{j=1}^d \left( \sum_{\gamma \in \mathbb{N}_p^n} s_j^r c_\gamma \alpha^\gamma y^\gamma x_{\cdot j}^\gamma \right) = \sum_{\gamma \in \mathbb{N}_p^n} c_\gamma \alpha^\gamma y^\gamma \sum_{j=1}^d s_j^r x_{\cdot j}^\gamma$$

where  $c_\gamma = \binom{\sum_{i=1}^n \gamma_i}{\gamma_1, \dots, \gamma_n} = \frac{(\sum_{i=1}^n \gamma_i)!}{\gamma_1! \dots \gamma_n!}$ , and  $\mathbb{N}_a^n := \{\gamma \in \mathbb{N}^n : \sum_{i=1}^n \gamma_i = a\}$ , for any  $a \in \mathbb{N}$ .

The above discussion justifies the validity of the following representation of  $(P_{LD})$  within the cone  $\mathcal{C}(s_1, \dots, s_d)$ :

$$(12) \quad \max \left( \frac{1}{p^r} - \frac{1}{p^{r-1}} \right) \sum_{\gamma \in \mathbb{N}_p^n} c_\gamma \alpha^\gamma y^\gamma \sum_{j=1}^d s_j^r x_{\cdot j}^\gamma + \sum_{i=1}^n \alpha_i$$

$$(13) \quad \text{s.t. } s_j \sum_{i=1}^n \alpha_i y_i x_{ij} \geq 0, \quad \forall j = 1, \dots, d,$$

$$(14) \quad \sum_{i=1}^n \alpha_i y_i = 0,$$

$$(15) \quad 0 \leq \alpha_i \leq C, \quad \forall i = 1, \dots, n.$$

Finally, let us deduce the expression of the separating hyperplane as a function of the optimal solution of  $(P_{LD})$ ,  $\bar{\alpha}$ . For a particular  $z \in \mathbb{R}^d$  the separating hyperplane is  $\mathcal{H} = \{(z_1, \dots, z_d) \in \mathbb{R}^d : \sum_{j=1}^d \omega_j z_j + b = 0\}$ . Using (11), this hyperplane is given by:

$$\sum_{j=1}^d \frac{1}{p^{r-1}} \mathcal{S}_{\bar{\alpha}, j}^r \left( \sum_{i=1}^n \bar{\alpha}_i y_i x_{ij} \right)^{r-1} z_j + b = 0,$$

where the signs are those associated to  $\bar{\alpha}$ . Equivalently,

$$\frac{1}{p^{r-1}} \sum_{\gamma \in \mathbb{N}_{r-1}^n} c_\gamma \bar{\alpha}^\gamma y^\gamma \sum_{j=1}^d \mathcal{S}_{\bar{\alpha}, j}^r x_{\cdot j}^\gamma z_j + b = 0.$$

Finally, to compute  $b$ , for any  $i_0 \in \{1, \dots, n\}$  with  $0 < \bar{\alpha}_{i_0} < C$ , by the complementary slackness conditions we get that we can also reconstruct the intercept of the hyperplane:

$$b = y_{i_0} - \sum_{j=1}^d \bar{\omega}_j x_{i_0 j} = y_{i_0} - \frac{1}{p^{q-1}} \sum_{j=1}^d \mathcal{S}_{\bar{\alpha}, j} \left( \mathcal{S}_{\bar{\alpha}, j} \sum_{i=1}^n \bar{\alpha}_i y_i x_{ij} \right)^{q-1} x_{i_0 j},$$

and the result follows.  $\square$

Observe that the even case can be seen as a particular case of the odd case in which a single arrangement is considered whose signs are all equal to one.

**Corollary 2.1.** *For even  $r$ , the Lagrangian dual problem,  $(P_{LD})$ , is given as:*

$$(16) \quad \max_{\alpha \in \mathcal{H}_y} \left( \frac{1}{p^r} - \frac{1}{p^{r-1}} \right) \sum_{\gamma \in \mathbb{N}_p^n} c_\gamma \alpha^\gamma y^\gamma \sum_{j=1}^d x_{\cdot j}^\gamma + \sum_{i=1}^n \alpha_i.$$



*Proof.* Note that if  $r$  is even one has that:

$$\sum_{j=1}^d \left| \sum_{i=1}^n \alpha_i y_i x_{ij} \right|^r = \sum_{j=1}^d \left( \sum_{i=1}^n \alpha_i y_i x_{ij} \right)^r.$$

Hence, the arrangement of hyperplanes (and signs patterns) are not needed in this case and the result follows.  $\square$

**Remark 2.1.** *Observe that formulation (12)-(15) can be slightly modified to be valid for the case  $q = \frac{r}{s}$  with  $s \neq 1$  as follows:*

$$\begin{aligned} & \max \left( \frac{1}{p^q} - \frac{1}{p^{q-1}} \right) \sum_{j=1}^d \delta_j + \sum_{i=1}^n \alpha_i \\ \text{s.t. } & \sum_{\gamma \in \mathbb{N}_+^n} c_\gamma \alpha^\gamma y^\gamma s_j^T x_j^\gamma - \delta_j^s \leq 0, & \forall j = 1, \dots, d, \\ & s_j \sum_{i=1}^n \alpha_i y_i x_{ij} \geq 0, & \forall j = 1, \dots, d, \\ & \sum_{i=1}^n \alpha_i y_i = 0, \\ & 0 \leq \alpha_i \leq C, & \forall i = 1, \dots, n, \\ & \delta_i \geq 0, & \forall j = 1, \dots, d. \end{aligned}$$

The results in Theorem 2.1, namely that the Lagrangian problem ( $P_{LD}$ ) and the separating hyperplane it induces depend on the original data throughout homogeneous polynomials; is the basis to introduce the concept of multidimensional kernel that extends further the kernel trick already known for the SVM problem with Euclidean distance. This is the aim of the following section.

### 3. MULTIDIMENSIONAL KERNELS

As mentioned in the introduction, when the linear separation between two sets is not clear in the original space, a common technique in supervised classification is to embed the data in a space of higher dimension where this separation may be easier. If we consider  $\Phi : \mathbb{R}^d \rightarrow \mathbb{R}^D$ , a transformation on the original data, the expressions of the Lagrangian dual problem and the separating hyperplane of these transformed data would depend on the function  $\Phi$ . In this sense, the increase of the dimension of the space would be translated in an increase of the difficulty to tackle the resulting problem. However, when the  $\ell_2$ -norm is used, the so-called *kernel trick* provides expressions of the Lagrangian dual problem and the separating hyperplane that just depend on the so-called Kernel function. Basically, the idea behind the kernel trick is to use a Kernel function to handle transformations on the data, and incorporate them to the SVM problem, without the explicit knowledge of the transformation function. Therefore, although implicitly we are solving a problem in a higher dimension, the resulting problem is stated in the dimension of the original data and as a consequence, it has the same difficulty than the original one. Our goal in this section is to extend this idea of the kernel trick to  $\ell_p$ -SVM. In order to do that, consider a data set  $[\mathbf{x}] = (x_1, \dots, x_n)$  together with their classification

pattern  $\mathbf{y} = (y_1, \dots, y_n)$  and  $r \in \mathbb{N}$ . Given  $\Phi : \mathbb{R}^d \rightarrow \mathbb{R}^D$ , the set-valued function  $S_\Phi : 2^{\mathbb{H}_\mathbf{y}} \rightarrow 2^{\{-1,1\}^D}$  ( $2^{\mathbb{H}_\mathbf{y}}$  stands for the power set of  $\mathbb{H}_\mathbf{y}$ ), is defined as:

$$S_\Phi(R) := \left\{ \mathbf{s} \in \{-1, 1\}^D : s_j = \operatorname{sgn} \left( \sum_{i=1}^n \alpha_i y_i \Phi_j(\mathbf{x}_i) \right)^r, \right. \\ \left. \text{for } j = 1, \dots, D, \text{ for some } \alpha \in R \right\}.$$

In what follows, we say that the family of sets  $\{R_k\}_{k \in \mathcal{K}} \subseteq 2^{\mathbb{H}_\mathbf{y}}$  is a *subdivision* of  $\mathbb{H}_\mathbf{y}$  if: (1)  $\mathcal{K}$  is finite; and (2)  $\bigcup_{k \in \mathcal{K}} R_k = \mathbb{H}_\mathbf{y}$  and  $\operatorname{ri}(R_k) \cap \operatorname{ri}(R_{k'}) = \emptyset$  for any  $k, k' (k \neq k') \in \mathcal{K}$  (where  $\operatorname{ri}(R)$  stands for the relative interior of a set  $R$ ).

**Definition 3.1.** *Given a transformation function,  $\Phi : \mathbb{R}^d \rightarrow \mathbb{R}^D$ , a subdivision  $\{R_k\}_{k \in \mathcal{K}}$  is said a suitable  $\Phi$ -subdivision of  $\mathbb{H}_\mathbf{y}$  if*

$$S_\Phi(R_k) = \{s_{R_k}\} \text{ for some } s_{R_k} \in \{-1, 1\}^D \text{ and for all } k \in \mathcal{K}.$$

Observe that the signs of  $\sum_{i=1}^n \alpha_i y_i \Phi_j(\mathbf{x}_i)$ , for  $j = 1, \dots, D$ , are constant within any element  $R_k$  of a suitable  $\Phi$ -subdivision. Hence, any finer subdivision of a suitable  $\Phi$ -subdivision remains suitable. Also, one may construct the maximal subdivision of  $\mathbb{H}_\mathbf{y}$  with such a property by defining:

$$\mathcal{C}(s_1, \dots, s_D) = \left\{ \alpha \in \mathbb{H}_\mathbf{y} : \operatorname{sgn} \left( \sum_{i=1}^n \alpha_i y_i \Phi_j(\mathbf{x}_i) \right)^r = s_j, \text{ for } j = 1, \dots, D \right\}$$

for any  $\mathbf{s} \in \{-1, 1\}^D$ , and choosing  $\{R_k\}_{k \in \mathcal{K}} = \left\{ \mathcal{C}(s_1, \dots, s_D) \right\}_{\mathbf{s} \in \{-1, 1\}^D}$  (observe that each set of this subdivision is defined univocally by a vector  $\mathbf{s} \in \{-1, 1\}^D$ ).

**Definition 3.2.** *Given a suitable  $\Phi$ -subdivision,  $\{R_k\}_{k \in \mathcal{K}} \subseteq 2^{\mathbb{H}_\mathbf{y}}$ , and  $(\gamma, \lambda) \in \mathbb{N}_r^{n+1}$ ,  $\lambda \in \{0, 1\}$ , the operator*

$$(17) \quad \mathbf{K}[\mathbf{x}]_{R_k, \gamma, \lambda}(z) := \sum_{j=1}^D s_{R_k, j}^r \Phi_j(\mathbf{x})^\gamma \Phi_j(z)^\lambda, \forall z \in \mathbb{R}^d, \forall k \in \mathcal{K},$$

*is called a  $r$ -order Kernel function of  $\Phi$ . For  $k \in \mathcal{K}$ ,  $\mathbf{K}[\mathbf{x}]_{R_k, \gamma, \lambda}(z)$  is called the  $k$ -th slice of the kernel function.*

The reader can observe that the objective function of (P<sub>LD</sub>) and the separating hyperplane obtained as a result of solving this problem can be rewritten for the  $\Phi$ -transformed data using the Kernel function (17).

Indeed, using (12) the objective function of the Lagrangian dual problem when using  $\Phi(\mathbf{x})$  instead of  $\mathbf{x}$  is:

$$\left( \frac{1}{p^r} - \frac{1}{p^{r-1}} \right) \sum_{\gamma \in \mathbb{N}_r^n} c_\gamma \alpha^\gamma y^\gamma \mathbf{K}[\mathbf{x}]_{R_k, \gamma, 0}(z) + \sum_{i=1}^n \alpha_i, \quad \forall \alpha \in R_k, \forall k \in \mathcal{K}.$$

Since the separating hyperplane is built for  $\alpha^* \in R_{k^*}$ , the optimal solution of an optimization problem, the expression (16) of this hyperplane is given by:

$$\frac{1}{p^{r-1}} \sum_{\gamma \in \mathbb{N}_{r-1}^n} c_\gamma \alpha^{*\gamma} y^\gamma \mathbf{K}[\mathbf{x}]_{R_{k^*}, \gamma, 1}(z) + b = 0, \quad \text{for } k^* \in \mathcal{K} \text{ such that } \alpha^* \in R_{k^*}.$$

**Remark 3.1.** *The general definition of kernel simplifies whenever  $r$  is even. In such a case, the sign coefficients are no longer needed. Hence,  $\{\mathbf{H}_y\}$  (with  $|\mathcal{K}| = 1$ ) is a suitable  $\Phi$ -subdivision of  $\mathbf{H}_y$  for any transformation  $\Phi : \mathbb{R}^d \rightarrow \mathbb{R}^D$ . Then, the kernel function becomes:*

$$K[\mathbf{x}]_{\mathbf{H}_y, \gamma, \lambda}(z) := \sum_{j=1}^D \Phi_j(\mathbf{x})^\gamma \Phi_j(z)^\lambda, \quad \forall z \in \mathbb{R}^d,$$

for  $(\gamma, \lambda) \in \mathbb{N}_r^n$  and  $\lambda \in \{0, 1\}$ , but being it independent of  $\alpha$  (since  $S_\Phi(\mathbf{H}_y) = \{(1, \dots, 1)\}$ ).

**Remark 3.2.** *For the Euclidean case ( $r = 2$ ), note that usual definition of kernel is  $K(z, z') = \Phi(z)^t \Phi(z')$  which is independent of the observations. Nevertheless, such an expression is only partially exploited in its application to the SVM problem. For solving the dual problem,  $K$  is applied to pairs of observations, i.e., only through  $K(\mathbf{x}_{i_1}, \mathbf{x}_{i_2})$  for  $i_1, i_2 = 1, \dots, n$ , whereas for classifying an arbitrary observation  $z$ , the unique expressions to be evaluated are of the form  $K(\mathbf{x}_i, z)$ .*

*Thus, the kernel for the Euclidean case can be expressed:*

$$K(\mathbf{x}_{i_1}, \mathbf{x}_{i_2}) = \Phi(\mathbf{x}_{i_1})^t \Phi(\mathbf{x}_{i_2}) = K[\mathbf{x}]_{\mathbf{H}_y, \gamma, 0}(z), \quad \forall z \in \mathbb{R}^d$$

for  $(\gamma, \lambda) = \mathbf{e}_{i_1} + \mathbf{e}_{i_2}$ ,  $i_1, i_2 = 1, \dots, n$ , with  $\lambda = 0$ , and

$$K(\mathbf{x}_{i_1}, z) = \Phi(\mathbf{x}_{i_1})^t \Phi(z) = K[\mathbf{x}]_{\mathbf{H}_y, \gamma, 1}(z), \quad \forall z \in \mathbb{R}^d$$

for  $(\gamma, \lambda) = \mathbf{e}_{i_1} + \mathbf{e}_{n+1}$ ,  $i_1 = 1, \dots, n$ , where  $\mathbf{e}_j$  denotes the  $j$ -th canonical  $(n+1)$ -dimensional vector, for  $j = 1, \dots, n$ . The above discussion shows that the standard Euclidean kernel is a particular case of our multidimensional kernel.

The following example illustrates the construction of the kernel operator for a given transformation  $\Phi$ .

**Example 3.1.** *Let us consider six points  $[\mathbf{x}] = ((0, 0), (0, 1), (1, 0), (1, 1), (1, -1), (-1, 1))$  on the plane with patterns  $\mathbf{y} = (1, 1, 1, -1, -1, -1)$ . The points are drawn in Figure 1 where the 1-class points are identified with filled dots while the  $-1$ -class is identified with circles. Clearly, the classes are not linearly separable.*

*Consider the transformation  $\Phi : \mathbb{R}^2 \rightarrow \mathbb{R}^3$ , defined as*

$$\Phi(x_1, x_2) = (x_1^2, \sqrt[3]{2}x_1x_2, x_2^2), \quad \forall (x_1, x_2) \in \mathbb{R}^2.$$

*Mapping the six points using  $\Phi$ , we get that, for any nonnegative integer  $r$ , the signs appearing at the kernel expressions are:  $\text{sgn}(\alpha_3 - \alpha_4 - \alpha_5 - \alpha_6)$ ,  $\text{sgn}(-\sqrt[3]{2}\alpha_4 + \sqrt[3]{2}\alpha_5 + \sqrt[3]{2}\alpha_6)$  and  $\text{sgn}(\alpha_2 - \alpha_4 - \alpha_5 - \alpha_6)$ . Since  $\mathbf{H}_y = \{\alpha \in \mathbb{R}_+^6 : \alpha_1 + \alpha_2 + \alpha_3 - \alpha_4 - \alpha_5 - \alpha_6 = 0\}$ , we get that the signs can be simplified to:*

- $\text{sgn}(\alpha_3 - \alpha_4 - \alpha_5 - \alpha_6) = \text{sgn}(-\alpha_1 - \alpha_2) = -1$ , if  $\alpha_1 + \alpha_2 > 0$  and 1 otherwise,
- $\text{sgn}(-\sqrt[3]{2}\alpha_4 + \sqrt[3]{2}\alpha_5 + \sqrt[3]{2}\alpha_6) = \text{sgn}(\alpha_5 + \alpha_6 - \alpha_4)$ .
- $\text{sgn}(\alpha_2 - \alpha_4 - \alpha_5 - \alpha_6) = \text{sgn}(-\alpha_1 - \alpha_3) = -1$ , if  $\alpha_1 + \alpha_3 > 0$  and 1 otherwise.

*Observe that the cases where the argument within the sign function is zero do not affect the formulations since the corresponding factor is null. Hence, only the*

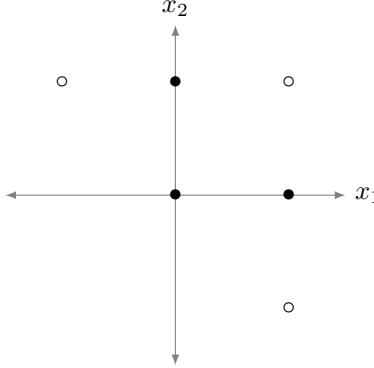


FIGURE 1. Points of Example 3.1 and their classification patterns.

expressions for the signs when  $\alpha_1 + \alpha_2 > 0$  (in the first item) and  $\alpha_1 + \alpha_3 > 0$  (in the third item) are considered.

For odd  $r$  (in which the  $r$ -th power of the signs above coincide with the signs themselves), we define the suitable subdivision  $\{R_1, R_2\}$ , where:

$$R_1 = \{\alpha \in \mathbf{H}_{\mathbf{y}} : \alpha_5 + \alpha_6 \geq \alpha_4\} \text{ and } R_2 = \{\alpha \in \mathbf{H}_{\mathbf{y}} : \alpha_5 + \alpha_6 \leq \alpha_4\}.$$

Note that  $S_{\Phi}(R_1) = \{(-1, 1, -1)\}$  while  $S_{\Phi}(R_2) = \{(-1, -1, -1)\}$ , i.e.,  $\{R_1, R_2\}$  is a suitable  $\Phi$ -subdivision.

Thus,

$$\mathbf{K}[\mathbf{x}]_{R_k, \gamma, \lambda}(z) = \begin{cases} -\Phi_1(\mathbf{x})^\gamma \Phi_1(z)^\lambda + \Phi_2(\mathbf{x})^\gamma \Phi_2(z)^\lambda - \Phi_3(\mathbf{x})^\gamma \Phi_3(z)^\lambda, & \text{if } k = 1, \\ -\Phi_1(\mathbf{x})^\gamma \Phi_1(z)^\lambda - \Phi_2(\mathbf{x})^\gamma \Phi_2(z)^\lambda - \Phi_3(\mathbf{x})^\gamma \Phi_3(z)^\lambda, & \text{if } k = 2, \end{cases}$$

being then:

$$\mathbf{K}[\mathbf{x}]_{R_k, \gamma, \lambda}(z) = \begin{cases} -(\mathbf{x}_1^\gamma z_1^\lambda - \mathbf{x}_2^\gamma z_2^\lambda)^2, & \text{if } k = 1, \\ -(\mathbf{x}_1^\gamma z_1^\lambda + \mathbf{x}_2^\gamma z_2^\lambda)^2, & \text{if } k = 2. \end{cases}$$

For even  $r$ , because the  $r$ -th power of the signs do not affect to the expressions, the  $r$ -order Kernel function of  $\Phi$  is given by

$$\mathbf{K}[\mathbf{x}]_{R_k, \gamma, \lambda}(z) = (\mathbf{x}_1^\gamma z_1^\lambda + \mathbf{x}_2^\gamma z_2^\lambda)^2,$$

for  $k = 1, 2$ ,  $(\gamma, \lambda) \in \mathbb{N}_r^{n+1}$  and  $\lambda \in \{0, 1\}$ . □

**3.1. Multidimensional Kernels and higher-dimensional tensors.** Given a subdivision  $\{R_k\}_{k \in \mathcal{K}}$  of  $\mathbf{H}_{\mathbf{y}}$  and a set of functions  $\{\mathbf{K}[\mathbf{x}]_{R_k, \gamma, \lambda}\}_{k \in \mathcal{K}}$ , for any  $(\gamma, \lambda) \in \mathbb{N}_r^{n+1}$  with  $\lambda \in \{0, 1\}$ , the *critical* question in this section is the existence of  $D \in \mathbb{Z}_+$  and  $\Phi : \mathbb{R}^d \rightarrow \mathbb{R}^D$  such that,  $\{R_k\}_{k \in \mathcal{K}}$  is a suitable  $\Phi$ -subdivision and

$$\mathbf{K}[\mathbf{x}]_{R_k, \gamma, \lambda}(z) := \sum_{j=1}^D s_{R_k, j}^r \Phi_j(\mathbf{x})^\gamma \Phi_j(z)^\lambda, \quad \forall z \in \mathbb{R}^d,$$

where  $S_{\Phi}(R_k) = \{s_{R_k}\}$  with  $s_{R_k} \in \{-1, 1\}^D$ .

For the sake of simplicity in the formulations, each of the elements of the  $\Phi$ -suitable subdivision of  $\mathbf{H}_{\mathbf{y}}$  will be denoted as follows:

$$R_k = \{\alpha \in \mathbb{R}^n : M_j^k \alpha \geq 0, j = 1, \dots, m_k\},$$

where  $M_j^k \in \mathbb{R}^n$ , for  $k \in \mathcal{K}$  and  $j = 1, \dots, m_k$ .

First of all, using that  $K[\mathbf{x}]_{R_k, \gamma, \lambda}(z)$  is a  $r$ -order Kernel function of  $\Phi$ , the problem (12)-(15) for a transformation of the original data via  $\Phi$ , in each element,  $k \in \mathcal{K}$ , of a suitable  $\Phi$ -subdivision can be written as:

$$(18) \quad \max F_k(\alpha) := \left( \frac{1}{p^r} - \frac{1}{p^{r-1}} \right) \sum_{\gamma \in \mathbb{N}_p^r} c_\gamma \alpha^\gamma y^\gamma K[\mathbf{x}]_{R_k, \gamma, 0}(z) + \sum_{i=1}^n \alpha_i$$

$$(19) \quad \text{s.t. } \tilde{g}_j^k(\alpha) := M_j^k \alpha \geq 0, \quad \forall j = 1, \dots, m_k,$$

$$(20) \quad \tilde{\ell}_0(\alpha) := \sum_{i=1}^n \alpha_i y_i = 0,$$

$$(21) \quad \tilde{\ell}_i(\alpha) := C - \alpha_i \geq 0, \quad \forall i = 1, \dots, n,$$

$$(22) \quad \tilde{\ell}_{n+i}(\alpha) := \alpha_i \geq 0, \quad \forall i = 1, \dots, n.$$

Observe that the problem above is a reformulation of the Lagrangian dual problem, (PLD), for the  $\Phi$ -transformed data that only depends on the original data via the  $r$ -order Kernel function of  $\Phi$  and the suitable  $\Phi$ -subdivision, and it can be seen as an extension of the kernel trick to  $\ell_p$ -norms with  $p > 1$ .

In the particular case where  $\Phi$  is the identity transformation, the above formulation becomes (12)-(15) whenever the suitable  $\Phi$ -subdivision consists of the full dimensional elements of the arrangement of hyperplanes  $\{\sum_{i=1}^n \alpha_i y_i x_{ij} = 0, j = 1, \dots, d\}$ . Furthermore, observe that  $F_k$  does not depend on  $z$ , since the degree,  $\lambda$ , of such a value is zero in that function.

We shall connect the above mentioned *critical* question with some interesting mathematical objects, *real symmetric tensors*, that are built upon the given data set  $[\mathbf{x}]$  and  $\mathbf{y}$ . It will become clear, after Theorem 3.1, that existence of a kernel operator is closely related with rank-one decompositions of the above mentioned tensors.

Recall that a real  $r$ -th order  $m$ -dimensional symmetric tensor,  $\mathbb{L}$ , consists of  $m^r$  real entries  $\mathbb{L}_{j_1 \dots j_r} \in \mathbb{R}$  such that  $\mathbb{L}_{j_1 \dots j_r} = \mathbb{L}_{j_{\sigma(1)} \dots j_{\sigma(r)}}$ , for any permutation  $\sigma$  of  $\{1, \dots, r\}$ .

**Lemma 3.1.** *Let  $\Phi : \mathbb{R}^d \rightarrow \mathbb{R}^D$ ,  $\hat{z} \in \mathbb{R}$  and let  $\mathcal{S} = \{R_k\}_{k \in \mathcal{K}}$  be a suitable  $\Phi$ -subdivision of  $\mathbf{H}_\mathbf{y}$ . Then, the  $k$ -th slice of any  $r$ -order kernel function of  $\Phi$  at  $\hat{z}$ , induces a real  $r$ -th order  $(n+1)$ -dimensional symmetric tensor.*

*Proof.* Let us define the following set of  $(n+1)^r$  real numbers:

$$\mathbb{K}_{i_1 \dots i_r}^k = \begin{cases} K[\mathbf{x}]_{R_k, \gamma_0, 0}(\hat{z}), & \text{if } i_j < n+1, \forall j = 1, \dots, r, \\ K[\mathbf{x}]_{R_k, \gamma_1, 1}(\hat{z}), & \text{if there exists } s \in \{1, \dots, r\} \text{ such that } i_s = n+1. \end{cases}$$

being  $(\gamma_0, \lambda) = \sum_{l=1}^r e_{i_l}$  with  $\lambda = 0$  and  $(\gamma_1, \lambda) = \sum_{l=1}^r e_{i_l}$  with  $\lambda = 1$ .

Let us check whether the above tensor is symmetric. Let  $\sigma$  be a permutation of the indices. For  $(i_1, \dots, i_r)$ , which comes from a particular choice of  $(\gamma, \lambda)$ , if  $\sigma$  is applied to  $(i_1, \dots, i_r)$ , the resulting  $(\gamma', \lambda')$  becomes:

$$(\gamma', \lambda') = \begin{cases} \sum_{l=1}^r e_{\sigma(l)}, & \text{if } i_{\sigma(i)} < n+1, \forall i, \\ \sum_{l=1}^r e_{\sigma(l)}, & \text{if } \exists s : i_{\sigma(s)} = n+1 \end{cases} = \begin{cases} \sum_{l=1}^r e_{i_l}, & \text{if } i_i < n+1, \forall i, \\ \sum_{l=1}^r e_{i_l}, & \text{if } \exists s : i_s = n+1 \end{cases} = (\gamma, \lambda)$$

Hence,  $\mathbb{K}_{i_1 \dots i_r} = \mathbb{K}_{i_{\sigma(1)} \dots i_{\sigma(r)}}$ , since the multi-indices constructed from  $(\gamma, \lambda)$  and  $(\gamma', \lambda')$  coincide.  $\square$

Let us now denote by  $\otimes$  the tensor product, i.e.  $v \otimes w = (v_i w_j)_{i,j=1}^m$  for any  $v, w \in \mathbb{R}^m$ .

**Lemma 3.2** ([10]). *Let  $\mathbb{K}$  be a real  $r$ -order  $(n+1)$ -dimensional symmetric tensor. Then, there exists  $\widehat{D} \in \mathbb{N}$ ,  $v_1, \dots, v_{\widehat{D}} \in \mathbb{R}^{n+1}$  and  $\psi_j \in \mathbb{R} \forall j = 1, \dots, \widehat{D}$ , such that  $\mathbb{K}$  can be decomposed as*

$$\mathbb{K} = \sum_{j=1}^{\widehat{D}} \psi_j v_j \otimes \dots \otimes v_j.$$

That is,  $\mathbb{K}_{i_1 \dots i_r} = \sum_{j=1}^{\widehat{D}} \psi_j v_{ji_1} \dots v_{ji_r}$  for any  $i_1, \dots, i_r \in \{1, \dots, n+1\}$ . Such a

decomposition is said a rank-one tensor decomposition of  $\mathbb{K}$ . The minimum  $\widehat{D}$  that assures such a decomposition is the symmetric tensor rank and  $\psi_1, \dots, \psi_{\widehat{D}}$  are its eigenvalues.

The following result extends the classical Mercer's Theorem [28] to  $r$ -order Kernel functions.

**Theorem 3.1.** *Let  $\{R_k\}_{k \in \mathcal{K}}$  be a subdivision of  $\mathbf{H}_y$  and  $\mathbb{K}^k$ , for  $k \in \mathcal{K}$ , be a  $r$ -order  $(n+1)$ -dimensional symmetric tensor such that each  $\mathbb{K}^k$  can be decomposed as:*

$$\mathbb{K}^k = \sum_{j=1}^{\widehat{D}} \psi_{kj} v_j \otimes \dots \otimes v_j, \forall k \in \mathcal{K},$$

and satisfying, either

- (1)  $r$  is even and  $\psi_j := \psi_{kj} \geq 0$ , or
- (2)  $r$  is odd and  $\psi_j := |\psi_{kj}|$  and for all  $k \in \mathcal{K}$ :

$$\text{sgn}(\psi_{kj}) = \text{sgn}\left(\sum_{i=1}^n \alpha_i y_i \sqrt[r]{\psi_j v_{ji}}\right), \text{ for all } \alpha \in \mathbb{R}_k.$$

Then, there exists a transformation  $\Phi$ , such that  $\{R_k\}_{k \in \mathcal{K}}$  is a  $\Phi$ -suitable subdivision of  $\mathbf{H}_y$  and  $\{\mathbb{K}^k\}_{k \in \mathcal{K}}$  induces a  $r$ -order kernel function of  $\Phi$ .

*Proof.* Let  $z \in \mathbb{R}^d$  and define  $\Phi : \mathbb{R}^d \rightarrow \mathbb{R}^{\widehat{D}}$  as:

$$\begin{cases} \Phi_j(x_i) &= \sqrt[r]{\psi_j v_{ji}}, \text{ for } i = 1, \dots, n, \\ \Phi_j(z) &= \sqrt[r]{\psi_j v_{j,n+1}}, \end{cases} \text{ for } j = 1, \dots, \widehat{D},$$

which is well defined because of the nonnegativity of the eigenvalues  $\psi_j$ .

- Let us assume first that  $r$  is even. Note that, since  $r$  is even and  $\{R_k\}_{k \in \mathcal{K}}$  is a suitable subdivision, the latest is also a suitable  $\Phi$ -subdivision (actually, for any  $\Phi$ ), since the signs are always positive (or the sign function is always 1).

Hence, for  $(\gamma, \lambda) = \sum_{i=1}^r e_{i_i}$ ,

$$\mathbb{K}[\mathbf{x}]_{R_k, \gamma, \lambda}(z) = \sum_{j=1}^{\widehat{D}} \Phi_j(\mathbf{x})^\gamma \Phi_j(z)^\lambda = \sum_{j=1}^{\widehat{D}} \psi_j v_{ji_1} \dots v_{ji_r} = \mathbb{K}_{i_1 \dots i_r}^k,$$

is a  $r$ -order kernel function of  $\Phi$ .

- Assume that  $r$  is odd. Observe that  $\text{sgn}\left(\sum_{i=1}^n \alpha_i y_i \Phi_j(x_i)\right)$   
 $= \text{sgn}\left(\sum_{i=1}^n \alpha_i y_i \sqrt[r]{\psi_j} v_{ji}\right) = \text{sgn}(\psi_{kj})$ , being then

$$S_\Phi(R_k) = \{(\text{sgn}(\psi_{k1}), \dots, \text{sgn}(\psi_{k\widehat{D}}))\}.$$

Thus, we get that  $\{R_k\}_{k \in \mathcal{K}}$  is a suitable  $\Phi$ -subdivision of  $H_{\mathbf{y}}$ .

Also, because  $\psi_{kj} = \text{sgn}(\psi_{kj})\psi_j$  and  $\text{sgn}(\psi_{kj}) = \text{sgn}\left(\sum_{i=1}^n \alpha_i y_i \sqrt[r]{\psi_j} v_{ji}\right)$ , for all  $\alpha \in \mathbb{R}_k$ , we get that:

$$\begin{aligned} \mathbb{K}[\mathbf{x}]_{R_k, \gamma, \lambda}(z) &= \sum_{j=1}^{\widehat{D}} \text{sgn}\left(\sum_{i=1}^n \alpha_i y_i \Phi_j(x_i)\right)^r \Phi_j(\mathbf{x})^\gamma \Phi_j(z)^\lambda \\ &= \sum_{j=1}^{\widehat{D}} \text{sgn}\left(\sum_{i=1}^n \alpha_i y_i \sqrt[r]{\psi_j} v_{ji}\right)^r \psi_j v_{ji_1} \cdots v_{ji_r} \\ &= \mathbb{K}_{i_1 \dots i_r}^k \end{aligned}$$

for  $(\gamma, \lambda) = \sum_{l=1}^r e_{i_l}$ . Hence,  $\mathbb{K}^k$  induces a  $k$ th-slice of a  $r$ -order kernel function of  $\Phi$ . □

The decomposition of symmetric 2-order  $n$ -dimensional tensors ( $n \times n$  symmetric matrices) provided in Lemma 3.2, is equivalent to eigenvalue decomposition [12] and the symmetric tensor rank coincides with the usual rank of a matrix. Hence, for  $r = 2$  (Euclidean case), the conditions of Theorem 3.1 reduce to check positive semidefiniteness of the induced kernel matrix (Mercer's Theorem). On the other hand, computing rank-one decompositions of higher-dimensional symmetric tensors is known to be NP-hard, even for symmetric 3-order tensors [17]. Actually, there is no finite algorithm to compute, in general, the rank one decompositions of general symmetric tensors. In spite of that, several algorithms have been proposed to perform such a decomposition. One commonly used strategy finds approximations to the decomposition by sequentially increasing the dimension of the transformed space ( $\widehat{D}$ ). Specifically, one fixes a dimension  $\widehat{D}$  and finds  $v$  and  $\psi$  that minimize  $\|\mathbb{K} - \sum_{j=1}^{\widehat{D}} \psi_{kj} v_j \otimes \cdots \otimes v_j\|_2^2$ . Next, if a zero-objective value is obtained, a tensor decomposition is found; otherwise,  $\widehat{D}$  is increased and the process is repeated. The interested reader referred to [7, 18, 20] for further information about algorithms for decomposing real symmetric tensors.

In some interesting cases, the assumptions of Theorem 3.1 are proved to be verified by some general classes of tensors. In particular, even order  $P$  tensors,  $B$  tensors,  $B_0$  tensors, diagonally dominated tensors, positive Cauchy tensors and sums-of-squares (SOS) tensors are known to have all their eigenvalues nonnegative (the reader is referred to [9, 31] for the definitions and results on these families of tensors). Thus, several classes of multidimensional kernel functions can be easily constructed. For instance, if  $r$  is even and we assume that all  $\mathbf{x}_i \neq \mathbf{0}$ , for all  $i = 1, \dots, n+1$ , it is well-known that the symmetric  $r$ -order  $(n+1)$ -dimensional tensor,  $\mathbb{K}$ , with entries:

$$\mathbb{K}_{i_1 \dots i_r} = \frac{1}{\|\mathbf{x}_{i_1}\| + \cdots + \|\mathbf{x}_{i_r}\|}, \quad i_1, \dots, i_r = 1, \dots, n+1,$$

for some norm  $\|\cdot\|$  in  $\mathbb{R}^d$ , is a Cauchy-shaped tensor. Next,  $\mathbb{K}$  is positive semidefinite [33], since  $\|x_i\| > 0$ , for all  $i = 1, \dots, n$ . Hence, by Theorem 3.1, it induces a  $r$ -order kernel function.

#### 4. SOLVING THE $\ell_p$ -SVM PROBLEM

By Theorem 2.1, (P<sub>LD</sub>) can be rewritten as a polynomial optimization problem, i.e., as a global optimization problem and then, in general, it is NP-hard. In spite of that we can approximately solve moderate size instances using a modern optimization technique based on the Theory of Moments that allows building a sequence of semidefinite programs whose solutions converge, in the limit, to the optimal solution of the original problem [23]. We use it to derive upper bounds for the Lagrangian dual problem (P<sub>LD</sub>).

Let us denote by  $\mathbb{R}[\alpha]$  the ring of real polynomials in the variables  $\alpha = (\alpha_1, \dots, \alpha_n)$ , for  $n \in \mathbb{N}$  ( $n \geq 1$ ), and by  $\mathbb{R}[\alpha]_r \subset \mathbb{R}[\alpha]$  the space of polynomials of degree at most  $r \in \mathbb{N}$ . We also denote by  $\mathcal{B} = \{\alpha^\gamma : \gamma \in \mathbb{N}^n\}$  a canonical basis of monomials for  $\mathbb{R}[\alpha]$ , where  $\alpha^\gamma = \alpha_1^{\gamma_1} \cdots \alpha_n^{\gamma_n}$ , for any  $\gamma \in \mathbb{N}^n$ . Note that  $\mathcal{B}_r = \{\alpha^\gamma \in \mathcal{B} : \sum_{i=1}^n \gamma_i \leq r\}$  is a basis for  $\mathbb{R}[\alpha]_r$ .

For any sequence indexed in the canonical monomial basis  $\mathcal{B}$ ,  $\mathbf{w} = (w_\gamma)_{\gamma \in \mathbb{N}^n} \subset \mathbb{R}$ , let  $\mathbf{L}_\mathbf{w} : \mathbb{R}[\alpha] \rightarrow \mathbb{R}$  be the linear functional defined, for any  $f = \sum_{\gamma \in \mathbb{N}^n} f_\gamma \alpha^\gamma \in \mathbb{R}[\alpha]$ , as  $\mathbf{L}_\mathbf{w}(f) := \sum_{\gamma \in \mathbb{N}^n} f_\gamma w_\gamma$ .

The *moment* matrix  $M_r(\mathbf{w})$  of order  $r$  associated with  $\mathbf{w}$ , has its rows and columns indexed by the elements in the basis  $\mathcal{B}_r$  and for two elements in such a basis,  $b_1 = \alpha^\gamma, b_2 = \alpha^\beta$ ,  $M_r(\mathbf{w})(b_1, b_2) = M_r(\mathbf{w})(\gamma, \beta) := \mathbf{L}_\mathbf{w}(\alpha^{\gamma+\beta}) = w_{\gamma+\beta}$ , for  $|\gamma|, |\beta| \leq r$  (here  $|a|$  stands for the sum of the coordinates of  $a \in \mathbb{N}^n$ ). Note that the moment matrix of order  $r$  has dimension  $\binom{n+r}{n} \times \binom{n+r}{n}$  and that the number of  $w_\gamma$  variables is  $\binom{n+2r}{n}$ .

For  $g = \sum_{\zeta \in \mathbb{N}^n} g_\zeta \alpha^\zeta \in \mathbb{R}[\alpha]$ , the *localizing* matrix  $M_r(g\mathbf{w})$  of order  $r$  associated with  $\mathbf{w}$  and  $g$ , has its rows and columns indexed by the elements in  $\mathcal{B}$  and for  $b_1 = \alpha^\gamma, b_2 = \alpha^\beta$ ,  $M_r(g\mathbf{w})(b_1, b_2) = M_r(g\mathbf{w})(\gamma, \beta) := \mathbf{L}_\mathbf{w}(\alpha^{\gamma+\beta}g(\alpha)) = \sum_{\zeta} g_\zeta w_{\zeta+\gamma+\beta}$ , for  $|\gamma|, |\beta| \leq r$ . Observe that a different choice for the basis of  $\mathbb{R}[\alpha]$ , instead of the standard monomial basis, would give different moment and localizing matrices, although the results would be also valid.

The main assumption to be imposed when one wants to assure convergence of some SDP relaxations for solving polynomial optimization problems is known as the Archimedean property (see for instance [23]) and it is a consequence of Putinar's results [30]. The importance of Archimedean property stems from the link between such a condition with the positive semidefiniteness of the moment and localizing matrices (see [30]).

We built a hierarchy of SDP relaxations ‘*a la Lasserre*’ for solving the dual problem. We observe that some constraints in this problem are already semidefinite (linear) therefore it is not mandatory to create their associated localizing constraints although its inclusion reinforces the relaxation values.

In the following result we state the semidefinite programming relaxations of the Lagrangean dual problem (18)-(22). Obviously, this result can be easily applied to problem (12)-(15) when we consider that  $\Phi$  is the identity transformation.



**Theorem 4.1.** *Let  $r \in \mathbb{Z}_+$  and  $\{R_k\}_{k \in \mathcal{K}}$  be a suitable  $\Phi$ -subdivision of  $H_{\mathbf{y}}$ . Let  $t \geq t_0 = \lceil \frac{r}{2} \rceil$ , and*

$$\begin{aligned} \rho_t^k &= \inf_{\mathbf{w}} L_{\mathbf{w}}(-F_k) \\ \text{s.t. } M_t(\mathbf{w}) &\succeq 0, \\ M_{t-1}(\tilde{g}_j^k \mathbf{w}) &\succeq 0, \quad j = 0, \dots, m_k, \\ M_{t-1}(\tilde{\ell}_i \mathbf{w}) &\succeq 0, \quad i = 0, \dots, n, \\ M_{t-1}(\tilde{\ell}_{n+i} \mathbf{w}) &\succeq 0, \quad i = 0, \dots, n, \\ L_{\mathbf{w}}(\mathbf{w}_0) &= 1. \end{aligned}$$

*Then, the sequence  $\{\rho_t^k\}_{t \geq t_0}$  of optimal values of the hierarchy of problems above satisfies*

$$\lim_{t \rightarrow +\infty} -\rho_t^k \downarrow \max_{\alpha \in \mathbb{R}_+^n} F_k(\alpha).$$

*Proof.* We apply Lasserre's hierarchy of SDP relaxations to approximate the dual global optimization problem. The feasible domain is compact since  $\alpha$  belongs to a closed and bounded set (recall that, in particular,  $\alpha \in H_{\mathbf{y}} \cap R_k$ , so  $\alpha \in [0, C]^n$ ) and therefore it satisfies the Archimedean property [30]. Next, the maximum degree of the polynomials involved in the problem is  $r$ . Therefore we can apply [23, Theorem 5.6] with relaxation orders  $t \geq t_0 := \lceil r/2 \rceil$  to conclude that the sequence,  $\{-\rho_t^k\}_{t \geq t_0}$ , of optimal values of the SDP relaxations, in the statement of the theorem, converges to  $\max_{\alpha \in \mathbb{R}_+^n} F_k(\alpha)$ , the optimal value of the Lagrangian problem.  $\square$

In many cases the convergence ensured by the above theorem is attained in a finite number of steps and it can be certified by a sufficient condition called the *rank condition* [23, Theorem 6.1], implying that the optimal  $\alpha$ -optimal values can be extracted.

**Example 4.1.** *Let us illustrate the proposed moment-SDP methodology to the toy instance of Example 3.1. Recall that we apply, for odd  $r$ , the kernel function  $K[\mathbf{x}]_{R_k, \gamma, \lambda}(z)$  given by:*

$$K[\mathbf{x}]_{R_k, \gamma, \lambda}(z) = \begin{cases} -(\mathbf{x}_1^\gamma z_1^\lambda - \mathbf{x}_2^\gamma z_2^\lambda)^2, & \text{if } k = 1, \\ -(\mathbf{x}_1^\gamma z_1^\lambda + \mathbf{x}_2^\gamma z_2^\lambda)^2, & \text{if } k = 2. \end{cases}$$

where  $R_1 = \{\alpha \in H_{\mathbf{y}} : \alpha_5 + \alpha_6 \geq \alpha_4\}$  and  $R_2 = \{\alpha \in H_{\mathbf{y}} : \alpha_5 + \alpha_6 \leq \alpha_4\}$ .

For  $r = 3, s = 1$ , i.e., when  $p = \frac{3}{2}$  and  $q = 3$ , the following two problems have to be solved:

$$\begin{aligned} \max & \left( \frac{1}{\left(\frac{3}{2}\right)^3} - \frac{1}{\left(\frac{3}{2}\right)^2} \right) \sum_{\gamma \in \mathbb{N}_3^6} c_\gamma \alpha^\gamma y^\gamma K[\mathbf{x}]_{R_k, \gamma, 0}(z) + \sum_{i=1}^6 \alpha_i \\ \text{s.t. } & \alpha \in H_{\mathbf{y}} \cap R_k. \end{aligned}$$

for  $k = 1, 2$ .

For the 1-th slide of the suitable subdivision,  $R_1$ , the problem is explicitly expressed as:

$$\begin{aligned} \max F_1(\alpha) &= \frac{-4}{27} (-\alpha_2^3 + 3\alpha_2^2\alpha_4 + 3\alpha_2^2\alpha_5 + 3\alpha_2^2\alpha_6 - 3\alpha_2\alpha_4^2 - 6\alpha_2\alpha_4\alpha_5 - 6\alpha_2\alpha_4\alpha_6 - \\ &\quad 3\alpha_2\alpha_5^2 - 6\alpha_2\alpha_5\alpha_6 - 3\alpha_2\alpha_6^2 - \alpha_3^3 + 3\alpha_3^2\alpha_4 + 3\alpha_3^2\alpha_5 + 3\alpha_3^2\alpha_6 - 3\alpha_3\alpha_4^2 - 6\alpha_3\alpha_4\alpha_5 - \\ &\quad 6\alpha_3\alpha_4\alpha_6 - 3\alpha_3\alpha_5^2 - 6\alpha_3\alpha_5\alpha_6 - 3\alpha_3\alpha_6^2 + 12\alpha_4^2\alpha_5 + 12\alpha_4^2\alpha_6 + 4\alpha_5^3 + 12\alpha_5^2\alpha_6 + \\ &\quad 12\alpha_5\alpha_6^2 + 4\alpha_6^3) + (\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 + \alpha_5 + \alpha_6) \\ \text{s.t. } \alpha_1 + \alpha_2 + \alpha_3 - \alpha_4 - \alpha_5 - \alpha_6 &= 0, \\ \alpha_5 + \alpha_6 &\geq \alpha_4, \\ 0 \leq \alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6 &\leq 10. \end{aligned}$$

We use *Gloptipoly* 3.8 [16] to translate the above problem into the SDP-relaxed problem and *SDPT3*[34] as the semidefinite programming solver.

Note that since the degree of the multivariate polynomial involved in the objective function is 3, at least a relaxation order of 2 is needed for the moment matrices. Using the basis  $\mathcal{B}_2$ , the moment matrix of order 2 has the following shape:

$$M_2(\mathbf{w}) = \begin{bmatrix} 1 & \alpha_1 & \cdots & \alpha_6 & \alpha_1^2 & \cdots & \alpha_6^2 \\ w_{000000} & w_{100000} & \cdots & w_{000001} & w_{200000} & \cdots & w_{000002} \\ w_{100000} & w_{200000} & \cdots & w_{100001} & w_{300000} & \cdots & w_{100002} \\ & & \ddots & & & & \\ w_{000002} & & & & & & w_{000004} \end{bmatrix} \begin{matrix} 1 \\ \alpha_1 \\ \vdots \\ \alpha_6^2 \end{matrix},$$

that is a  $28 \times 28$  real matrix and in which 210 variables are involved.

Observe that the constraints of the problems are linear, so no need of relaxing the constraints or using localizing matrices is needed. The semidefinite problem to solve is:

$$\begin{aligned} \rho_2^1 &= \min L_{\mathbf{w}}(-F_1) \\ \text{s.t. } M_2(\mathbf{w}) &\succ 0, \\ w_{100000} + w_{010000} + w_{001000} - w_{000100} - w_{000010} - w_{000001} &= 0, \\ w_{000010} + w_{000001} &\geq w_{000100}, \\ 0 \leq w_{100000}, w_{010000}, w_{001000}, w_{000100}, w_{000010}, w_{000001} &\leq 10, \end{aligned}$$

where in  $L_{\mathbf{w}}(-F_1)$  each term  $\alpha^\gamma$  is transformed into  $w_\gamma$ , for  $\gamma \in \mathbb{N}_r^n$ .

Solving the above problem, we get  $\rho_2^1 = -5.6569$  and:

$$w_{100000} = 0, w_{010000} = w_{001000} = w_{000100} = 2.1213, w_{000010} = w_{000001} = 1.0611.$$

Also, *Gloptipoly* checked that the rank condition holds, certifying that  $\alpha^* = (0, 2.1213, 2.1213, 2.1213, 1.0611, 1.0611)$  is optimal for our problem.

The problem for the second subdivision,  $R_2$ , can be analogously stated (by considering  $\alpha_5 + \alpha_6 \leq \alpha_4$  instead of the one defining  $R_1$ ). Also, for relaxation order 2, *Gloptipoly* obtained an optimal value of  $\rho_2^2 = -5.6569$  (the same objective value as for  $R_1$ ), and the solution was certified to be optimal with the same values as in the problem for  $k = 1$  (observe that the obtained optimal solution belongs to both  $R_1$  and  $R_2$  since  $\alpha_5^* + \alpha_6^* = \alpha_4^*$ ).

The optimal separating hyperplane is now constructed from  $\alpha^*$ . First, the intercept,  $b$ , is derived by using an observation  $i_0$  such that  $0 < \alpha_{i_0} < 10$ , for instance taking  $i_0 = 2$  ( $y_2 = 1$  and  $\mathbf{x}_{2\cdot} = (0, 1)$ ), being

$$b = y_2 - \frac{1}{\left(\frac{3}{2}\right)^2} \sum_{\gamma \in \mathbb{N}_2^6} c_\gamma (\alpha^*)^\gamma y^\gamma \mathbf{K}[\mathbf{x}]_{R_1, \gamma, 1}(\mathbf{x}_{2\cdot}) = 1 - \frac{1}{\left(\frac{3}{2}\right)^2} \sum_{\gamma \in \mathbb{N}_2^6} c_\gamma (\alpha^*)^\gamma y^\gamma \mathbf{x}_{\cdot 1}^{2\gamma} = 3$$

Thus, the hyperplane has the following shape:

$$\begin{aligned} \mathcal{H} &= \left\{ z \in \mathbb{R}^2 : \frac{1}{\left(\frac{3}{2}\right)^2} \sum_{\gamma \in \mathbb{N}_2^6} c_\gamma \alpha^{*\gamma} y^\gamma \mathbf{K}[\mathbf{x}]_{R_1, \gamma, 1}(z) + 3 = 0 \right\} \\ &= \left\{ z \in \mathbb{R}^2 : \frac{1}{\left(\frac{3}{2}\right)^2} \sum_{\gamma \in \mathbb{N}_2^6} c_\gamma \alpha^{*\gamma} y^\gamma (\mathbf{x}_{\cdot 1}^\gamma z_1 - \mathbf{x}_{\cdot 2}^\gamma z_2)^2 = -3 \right\}, \end{aligned}$$

and the data can be classified according to the sign of the evaluation of each point on the above hyperplane. Hence, for the six obtained points, we get that  $\mathbf{x}_{1\cdot}$ ,  $\mathbf{x}_{2\cdot}$ , and  $\mathbf{x}_{3\cdot}$  are classified in the  $+1$  side of the separating hyperplane, while  $\mathbf{x}_{4\cdot}$ ,  $\mathbf{x}_{5\cdot}$ , and  $\mathbf{x}_{6\cdot}$  are classified in the  $-1$  side. Thus, all the points are well-classified.

**Remark 4.1.** Theorem 4.1 can be extended to solve the Lagrangian dual problem related to the general case in which  $q = \frac{r}{s}$ , for  $r, s \in \mathbb{Z}_+$  with  $\gcd(r, s) = 1$  and  $q > 1$  (see Remark 2.1). In such a case, if  $\{R_k\}_{k \in \mathcal{K}}$  is a suitable  $\Phi$ -subdivision of  $\mathbf{H}_y$  and  $t \geq t_0 = \lceil \frac{r}{2} \rceil$ , we have to consider the following hierarchy of semidefinite programming problems

$$\begin{aligned} \tilde{\rho}_t^k &= \inf_{\mathbf{w}} L_{\mathbf{w}}(-\tilde{f}_{sR_k}) \\ \text{s.t. } & \mathbf{M}_t(\mathbf{w}) \succeq 0, \\ & \mathbf{M}_{t-1}(\tilde{g}_j^k \mathbf{w}) \succeq 0, \quad j = 1, \dots, m_k \\ & \mathbf{M}_{t-\lceil \frac{r}{2} \rceil}(-\tilde{g}_{m_k+j}^k \mathbf{w}) \succeq 0, \quad j = 1, \dots, D, \\ & \mathbf{M}_{t-1}(\tilde{\ell}_i \mathbf{w}) \succeq 0, \quad i = 0, \dots, n, \\ & \mathbf{M}_{t-1}(\tilde{\ell}_{n+i} \mathbf{w}) \succeq 0, \quad i = 0, \dots, n, \\ & L_{\mathbf{w}}(\mathbf{w}_0) = 1, \end{aligned}$$

where,  $\tilde{g}_{m_k+j}^k(\alpha, \delta) = \sum_{\gamma \in \mathbb{N}_r^n} c_\gamma \alpha^\gamma y^\gamma s_{R_k, j}^r \mathbf{x}_{\cdot j}^\gamma - \delta_j^s$  for  $j = 1, \dots, D$ . Then, it is satisfied that

$$\lim_{t \rightarrow +\infty} -\tilde{\rho}_t^k \downarrow \max_{\alpha, \delta} \tilde{f}_{sR_k}(\alpha, \delta).$$

Observe that the polynomial optimization problems in both cases,  $q$  integer and rational, have a similar shape (and also their semidefinite programming relaxations). However, for fractional  $q$ , there are  $D$  variables  $\delta_1, \dots, \delta_D$ , as well as  $D$  constraints,  $\tilde{g}_{m_k+1}^k(\alpha), \dots, \tilde{g}_{m_k+D}^k(\alpha)$ , defining the feasible region (recall that  $\tilde{g}_1^k(\alpha), \dots, \tilde{g}_{m_k}^k(\alpha)$  describe the cells of a suitable  $\Phi$ -subdivision), so the kernel trick cannot be applied under this setting.

**4.1. Solving the primal  $\ell_p$ -SVM problem.** The above machinery allows us to solve the dual of the SVM problem, resorting to hierarchies of SDP problems. The main drawback of that approach is the increasing size of the SDP objects that have to be handled as the relaxation order of the problem grows. The current development of SDP solvers limits the applicability of that approach to problems with several hundreds of variables which may be an issue to handle big databases.

One way to overcome that inconvenience is to attack directly the primal problem. Our strategy in order to solve the primal problem will be the following. Let  $\mathcal{C}_{\mathbb{R}^D}(T)$  be the Banach space of continuous functions from a compact set  $T \subseteq \mathbb{R}^d$  to  $\mathbb{R}^D$ . It is well-known that  $\mathcal{C}_{\mathbb{R}^D}(T)$  admits a Schauder basis (see [24]). In particular,  $\mathcal{B} = \{z^\gamma : \gamma \in \mathbb{N}^d\}$ , the standard basis of multidimensional monomials is a Schauder basis for this space. Also Bernstein and trigonometric polynomials and some others are Schauder bases of this space. This means that any continuous function defined on  $T$  can be exactly represented as a sum of terms in the basis  $\mathcal{B}$  (sometimes infinitely many). Thus for any continuous function  $\Phi : T \mapsto \mathbb{R}^D$ , there exists an expansion such that  $\Phi(z) = \sum_{j=1}^{\infty} \tau_j z_j$ , with  $\tau_j \in \mathbb{R}$  and  $z_j \in \mathcal{B}$  for any  $j = 1, \dots, \infty$ .

These expansions are function dependent but one may expect that with a sufficient number of terms we can approximate up to a certain degree of accuracy the standard kernel transformations usually applied in SVM. In this regard, our solution strategy transforms the original data by using a truncated Schauder basis (up to a given number of terms) and then solves the transformed problem ( $\ell_p$ -SVM) in this new extended space of original variables. This provides the classification in the extended space and this classification is applied to the original data. By standard arguments based on continuity and compactness given a prespecified accuracy the truncation order can be fixed to ensure the result.

**Example 4.2.** *We illustrate the primal methodology for the same dataset of Example 4.1. If the transformation provided in such an example is used to compute the  $\ell_{\frac{3}{2}}$ -SVM, we get the following primal formulation:*

$$\begin{aligned}
\min \quad & t + 10\xi_1 + 10\xi_2 + 10\xi_3 + 10\xi_4 + 10\xi_5 + 10\xi_6 \\
s.t. \quad & b + \xi_1 \geq 1, \\
& \omega_3 + b + \xi_2 \geq 1, \\
& \omega_1 + b + \xi_3 \geq 1, \\
& -\omega_1 - \sqrt[3]{2}\omega_2 - \omega_3 - b + \xi_4 \geq 1, \\
& -\omega_1 + \sqrt[3]{2}\omega_2 - \omega_3 - b + \xi_5 \geq 1, \\
& -\omega_1 + \sqrt[3]{2}\omega_2 - \omega_3 - b + \xi_6 \geq 1, \\
& t^2 \geq \|\omega\|_{\frac{3}{2}}^3, \\
& \xi_i \geq 0, i = 1, \dots, 6, \\
& b \in \mathbb{R}, \omega_j \in \mathbb{R}, j = 1, 2, 3.
\end{aligned}$$

Note that the constraint  $t^2 \geq \|w\|_{\frac{3}{2}}^3$  can be equivalently rewritten, by introducing the auxiliary variables  $\zeta_1, \zeta_2$  and  $\zeta_3$ , as:

$$\begin{cases} t^2 \geq \sum_{i=1}^d u_i, \\ v_j \geq \omega_j, v_j \geq -\omega_j, j = 1, 2, 3, \\ u_j \zeta_j \geq v_j^2, j = 1, 2, 3, \\ v_j \geq \zeta_j^2, j = 1, 2, 3, \end{cases}$$

since, for each  $j = 1, 2, 3$ ,  $v_j$  represents  $|\omega_j|$ , and because of the above non-linear constraints, we have that:

$$v_j^4 \leq \zeta_j^2 u_j^2 \leq u_j^2 v_j \Rightarrow u_j^2 \geq v_j^3 \quad (u_j \geq v_j^{\frac{3}{2}})$$

Thus, solving the above second order cone programming problem we get  $\omega^* = (2, 0, 2)$  and  $b^* = 3$ . We also obtain that all the misclassifying errors  $\xi$  are equal to zero. The same result was obtained in Example 4.1. In Figure 2, we draw (left picture) the separating curve when projecting the obtained hyperplane onto the original feature space.

Let us consider now the Schauder basis for continuous functions that consists of all monomials in  $\mathbb{R}[z_1, \dots, z_d]$ . One may define the transformation  $\Phi : \mathbb{R}^d \rightarrow \mathbb{R}^{\mathbb{N}^d}$ ,  $\Phi_\gamma(z) = z^\gamma$ , for each  $\gamma \in \mathbb{N}^d$ . Note that  $\Phi$  projects the original finite-dimensional feature space onto the infinite dimensional space of sequences  $\{z^\gamma\}_{\gamma \in \mathbb{N}^d}$ . Hence, for any  $z \in \mathbb{R}^d$  and  $\gamma \in \mathbb{N}^d$ , the  $\gamma$  component of  $\Phi$ ,  $\Phi_\gamma(z)$ , is a real number. Truncating the basis  $\mathcal{B}$  by a given order  $\eta \in \mathbb{N}$ , we define the transformation  $\Phi[\eta] : \mathbb{R}^d \rightarrow \mathbb{R}^{\mathbb{N}_\eta^d}$ ,  $\Phi[\eta]_\gamma(z) = z^\gamma$ , for each  $\gamma \in \mathbb{N}_\eta^d$ . Note that  $\mathbb{R}^{\mathbb{N}_\eta^d}$  is a finite-dimensional space with dimension  $\binom{d+\eta}{d}$ .

For instance, using  $\Phi[3]$ , the data are transformed into:

$$\begin{aligned} X' = \{ & (1, 0, 0, 0, 0, 0, 0, 0, 0, 0), (1, 0, 1, 0, 0, 1, 0, 0, 0, 1), (1, 1, 0, 1, 0, 0, 1, 0, 0, 0), \\ & (1, 1, 1, 1, 1, 1, 1, 1, 1, 1), (1, 1, -1, 1, -1, 1, 1, -1, 1, -1), \\ & (1, -1, 1, 1, -1, 1, -1, 1, -1, 1) \} \subseteq \mathbb{R}^{10} \end{aligned}$$

Then, solving ( $\ell_p$ -SVM) for this new dataset, we get, that in this new feature space (of dimension 10), the optimal coefficients are:

$$\omega^* = (0, 0.1117, 0.1117, -1.3295, 0.4469, -1.3295, 0.1117, -0.6704, -0.6704, 0.1117),$$

$$b^* = 2.1060,$$

which define, when projecting it onto the original feature space, the curve drawn in Figure 2 (center). This solution also perfectly classifies the given points.

If we truncate the Schauder basis up to degree 4 using  $\Phi[4]$  (transforming the data into a 15-dimensional space), we obtain the curve drawn in the right side of Figure 2.

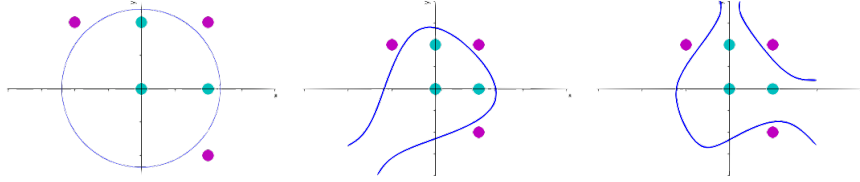


FIGURE 2. Separating curves with the three different settings (left: using the quadratic transformation, center: using  $\Phi[3]$ , right: using  $\Phi[4]$ ).

## 5. EXPERIMENTS

We have performed a series of experiments to analyze the behavior of the proposed methods on real-world benchmark data sets. We have implemented the primal second-order cone formulation (1)–(5) and, in order to find non-linear separators, we consider the following two types of transformations on the data which can be identified with adequate truncated Schauder bases:

- $\Phi[\eta] : \mathbb{R}^d \rightarrow \mathbb{R}^{\mathbb{N}_\eta^d}$ . Its components,  $\Phi[\eta]_\gamma(\mathbf{z}) = z^\gamma$  for  $\gamma \in \mathbb{N}_\eta^d$ , are the monomials (in  $d$  variables) up to degree  $\eta$ .
- $\tilde{\Phi}[\eta] : \mathbb{R}^d \rightarrow \mathbb{R}^{\mathbb{N}_\eta^d}$ , with  $\tilde{\Phi}[\eta]_\gamma(\mathbf{z}) = \exp(-\sigma\|\mathbf{z}\|_2^2) \frac{\sqrt[2]{2\sigma} \mathbf{z}^\gamma}{\sqrt{\gamma_1! \cdots \gamma_d!}}$ , for  $\mathbf{z} \in \mathbb{R}^d$ , for  $\gamma \in \mathbb{N}_\eta^d$  and  $\sigma > 0$ .

Although both transformations have a similar shape (their components consist of monomials of certain degrees), the second one has non-unitary coefficients. Those coefficients come from the construction of the Gaussian transformation which turns out to be the Gaussian kernel. In this second case, the higher the order, the closer the induced (polynomial) kernel to the gaussian kernel. Observe that the following generalized Gaussian operator  $\mathbb{G} : \mathbb{R}^{r \times d} \rightarrow \mathbb{R}$  defined as

$$\begin{aligned} \mathbb{G}[\mathbf{x}_{i_1}, \dots, \mathbf{x}_{i_r}] &= \exp\left(-\sigma \sum_{a,b=1}^r \|x_{i_a} - x_{i_b}\|_2^2\right) \\ &= \exp\left(-\sigma \sum_{a=1}^r \|x_{i_a}\|_2^2\right) \sum_{\gamma \in \mathbb{N}^d} \frac{2\sigma}{\gamma_1! \cdots \gamma_d!} \mathbf{x}_{i_1}^\gamma \cdots \mathbf{x}_{i_r}^\gamma, \end{aligned}$$

is induced by using the transformation  $\tilde{\Phi}$ , i.e.,  $\mathbb{G}[\mathbf{x}_{i_1}, \dots, \mathbf{x}_{i_r}] = \lim_{\eta \rightarrow \infty} \sum_{\gamma \in \mathbb{N}_\eta^d} \tilde{\Phi}[\eta]_\gamma$

where  $\gamma = \sum_{i=1}^r e_{i_i}$ . Hence, the transformation  $\tilde{\Phi}[\eta]$ , for a given  $\eta$ , is nothing but a truncated expansion of the generalized Gaussian operator  $\mathbb{G}$ .

We construct, in our experiments,  $\ell_p$ -SVM separators for  $p \in \{\frac{4}{3}, \frac{3}{2}, 2, 3\}$  by using  $\eta$ -order approximations with  $\eta$  ranging in  $\{1, 2, 3, 4\}$ . The case  $\eta = 1$  coincides with the linear separating hyperplane for both transformations.

The resulting primal Second Order Cone Programming (SOCP) problem was coded in Python 3.6, and solved using Gurobi 7.51 in a Mac OSX El Capitan with an Intel Core i7 processor at 3.3 GHz and 16GB of RAM.

The models were tested in four classical data sets, widely used in the literature of SVM, that are listed in Table 1. They were obtained from the UCI Repository

[32] and LIBSVM Datasets[8]. There, one can find further information about each of them.

Name	# Obs. ( $n$ )	# Features ( $d$ )	Source
cleveland	303	13	UCI Repository
housing	506	13	UCI Repository
german credit	1000	24	UCI Repository
colon	62	2000	LIBSVM Datasets

TABLE 1. Datasets used in our experiments.

In order to obtain stable and meaningful results, we use a 10-fold cross validation scheme to train the model and to test its performance. We report the accuracy of the model, which is defined as:

$$\text{ACC} = \frac{TP + TN}{n} \cdot 100$$

where  $TP$  and  $TN$  are the true positive and true negative predicted values after applying the model built with the training data set to a dataset (in our case to the training or the test sample). ACC is actually the percentage of well-classified observations. We report both the averages ACC for the training data ( $\text{ACC}^{\text{Tr}}$ ) and the test data ( $\text{ACC}^{\text{Test}}$ ), and also the averages CPU times for solving each one of the ten fold cross validation subproblems using the training data. We also report the average percentage of nonzero coefficients of the optimal separating hyperplanes, over the total number of variables of the problem (%NonZ).

$\eta$	$\ell_4$				$\ell_3$				$\ell_2$				$\ell_3$			
	$\text{ACC}^{\text{Tr}}$	$\text{ACC}^{\text{Test}}$	Time	%NonZ	$\text{ACC}^{\text{Tr}}$	$\text{ACC}^{\text{Test}}$	Time	%NonZ	$\text{ACC}^{\text{Tr}}$	$\text{ACC}^{\text{Test}}$	Time	%NonZ	$\text{ACC}^{\text{Tr}}$	$\text{ACC}^{\text{Test}}$	Time	%NonZ
cleveland dataset ( $C = 4$ )																
1	85.11%	82.84%	0.01	100%	85.11%	83.16%	0.01	100%	85.15%	<b>83.48%</b>	0.01	100%	85.33%	83.15%	0.01	100%
2	94.02%	<b>82.57%</b>	0.44	88.86%	93.58%	81.57%	0.40	94.48%	93.33%	81.58%	0.04	98.95%	93.35%	79.61%	0.41	98.31%
3	99.34%	74.93%	5.49	72.02%	99.41%	75.60%	2.87	84.84%	99.67%	78.53%	0.14	98.82%	99.67%	<b>80.23%</b>	2.65	99.66%
4	99.67%	76.56%	28	72.00%	99.67%	76.92%	22.5	81.88%	99.74%	<b>79.21%</b>	0.47	97.54%	100%	78.60%	17.56	99.31%
housing dataset ( $C = 64$ )																
1	88.56%	<b>85.36%</b>	0.01	100%	88.25%	85.16%	0.02	100%	88.10%	84.36%	0.02	100%	87.92%	83.35%	0.04	100%
2	94.93%	78.85%	0.22	90.57%	94.14%	80.03%	0.42	96.67%	92.31%	80.02%	0.14	99.05%	91.15%	<b>81.38%</b>	0.39	98.86%
3	98.60%	<b>80.95%</b>	9.57	57.36%	98.24%	80.00%	6.13	74.84%	97.34%	79.81%	0.51	97.27%	96.07%	78.84%	5.86	99.59%
4	99.23%	<b>79.99%</b>	45.09	50.82%	98.90%	77.78%	31.69	68.32%	98.37%	78.63%	1.59	95.30%	97.98%	78.43%	27.42	98.53%
german credit dataset ( $C = 64$ )																
1	78.53%	<b>76.20%</b>	0.02	99.58%	78.53%	<b>76.20%</b>	0.04	99.58%	78.53%	<b>76.20%</b>	0.05	99.58%	78.54%	<b>76.20%</b>	0.04	99.58%
2	93.03%	67.50%	0.92	96.62%	93.04%	67.60%	2.50	98.15%	92.98%	67.40%	0.50	99.69%	93.00%	<b>67.70%</b>	3.32	99.75%
3	100%	<b>71.90%</b>	85.86	60.93%	100%	70.50%	94.12	78.20%	100%	70.20%	3.14	96.76%	100%	68.90%	98.58	99.65%
colon dataset ( $C = 1$ )																
1	100%	<b>82.14%</b>	20.3	46.14%	100%	80.48%	15.73	64.54%	100%	80.48%	0.05	89.74%	100%	80.48%	14.61	99.44%

TABLE 2. Results of our computational experiments for  $\Phi[\eta]$ .

Since our models depend on two parameters ( $C$  and  $\eta$ ) and one more ( $\sigma$ ) in case of using the transformation  $\tilde{\Phi}[\eta]$ , we first perform a test to find the best choices for  $C$  and  $\sigma$ . For each dataset, we consider a part of the training sample and run the models by moving  $C$  and  $\sigma$  over the grid  $\{2^k : k \in \{-7, -6, \dots, 6, 7\}\}$ . For each dataset, the best combination of parameters is identified and chosen. Then, it is used for the rest of the experiments on such a dataset.

Tables 2 and 3 report, respectively, the average results for the feature transformations  $\Phi[\eta]$  and  $\tilde{\Phi}[\eta]$ . We report the results on those choices of  $\eta$  that result in

$\eta$	$\ell_{\frac{4}{3}}$				$\ell_{\frac{3}{2}}$				$\ell_2$				$\ell_3$			
	ACC <sup>Tr</sup>	ACC <sup>Test</sup>	Time	%NonZ	ACC <sup>Tr</sup>	ACC <sup>Test</sup>	Time	%NonZ	ACC <sup>Tr</sup>	ACC <sup>Test</sup>	Time	%NonZ	ACC <sup>Tr</sup>	ACC <sup>Test</sup>	Time	%NonZ
cleveland dataset ( $C = 2$ and $\sigma = 2^{-6}$ )																
1	85.15%	83.16%	0.01	99.23%	85.11%	83.16%	0.01	99.23%	85.33%	<b>83.48%</b>	0.01	100%	85.22%	<b>83.48%</b>	0.01	100%
2	88.30%	<b>84.19%</b>	0.24	66.86%	88.05%	82.55%	0.28	84.00%	86.72%	80.58%	0.04	99.52%	84.01%	77.26%	0.24	99.81%
3	92.15%	80.87%	4.91	49.25%	92.12%	81.54%	2.77	68.50%	92.41%	<b>81.55%</b>	0.13	96.54%	92.59%	81.20%	2.54	99.68%
4	84.38%	83.47%	19.57	3.53%	84.41%	83.46%	12.83	8.47%	84.71%	83.46%	0.19	41.97%	85.18%	<b>83.48%</b>	15.51	63.50%
housing dataset ( $C = 64$ and $\sigma = 2^{-6}$ )																
1	88.56%	<b>85.36%</b>	0.01	100%	88.25%	85.16%	0.02	100%	88.10%	84.36%	0.02	100%	87.53%	84.71%	0.04	100%
2	89.53%	<b>83.53%</b>	0.25	75.14%	88.84%	82.95%	0.48	88.48%	87.42%	82.94%	0.11	99.24%	86.72%	82.46%	0.66	100%
3	94.01%	80.03%	4.47	37.38%	93.30%	79.82%	4.29	54.52%	91.50%	<b>80.21%</b>	0.25	88.30%	90.36%	79.95%	3.05	99.62%
4	90.80%	82.37%	14.43	4.23%	90.58%	<b>83.36%</b>	20.98	7.56%	88.95%	81.59%	0.17	20.31%	86.69%	82.95%	12.2	65.97%
german credit dataset ( $C = 0.25$ and $\sigma = 2^{-6}$ )																
1	78.35%	<b>79.00%</b>	0.02	99.48%	78.33%	78.88%	0.04	100%	78.25%	78.63%	0.05	100%	78.26%	78.75%	0.04	100%
2	77.29%	74.38%	2.96	90.23%	77.83%	75.00%	2.37	97.62%	79.23%	74.44%	0.45	99.97%	81.15%	<b>75.22%</b>	2.13	100%
3	76.72%	76.75%	57.01	3.39%	92.78%	<b>79.00%</b>	63.64	91.69%	96.36%	77.88%	2.75	99.82%	98.24%	76.57%	48.4	99.99%
colon dataset ( $C = 1$ )																
1	100%	<b>82.14%</b>	20.3	46.14%	100%	80.48%	15.73	64.54%	100%	80.48%	0.05	89.74%	100%	80.48%	14.61	99.44%

TABLE 3. Results of our computational experiments for transformation  $\tilde{\Phi}[\eta]$ .

a good compromise between some improvement in accuracy and complexity on the problem solving. For instance, while for the datasets `cleveland` and `housing`, a degree up to  $\eta = 4$  was considered, for `german credit` a degree of  $\eta = 3$  already allows us to perfectly fit the data ( $\text{ACC}^{\text{Tr}} = 100\%$ ), and for `colon`,  $\eta = 1$ , i.e., the linear fitting, is enough to correctly classify the training sample. The best accuracy results for each  $\eta$  and each dataset are boldfaced. As a general observation of our experiment if  $\Phi[\eta]$  is used, there is no gain (in terms of accuracy on the testing sample) by increasing the value of  $\eta$  since the linear hyperplane is the one where we got the best results. However, such a situation changes when  $\tilde{\Phi}[\eta]$  is used since we found datasets (as `cleveland` or `german`) in which the best accuracy results are obtained for non-linear transformations. It can be also observed that a best fitting for the training data does not always imply the best performance for the test data. This behavior may be due to overfitting.

Concerning the use of different norms, one can observe that there is no a significative best one in terms of accuracy on the test sample, although we obtain most of the best results using  $\ell_{\frac{4}{3}}$ . At this point, we would like to remark that the usual norm used in SVM, the Euclidean norm, does not stand out over the others. On the other hand, the  $\ell_2$ -norm cases are solved in much smaller computational times than the other, since this norm is directly representable as a single quadratic constraint in our model, while the others need to consider auxiliary variables and constraints which increase the complexity for solving the problem. In spite of that, the remaining computational times are reasonable with respect to the size of the instances.

In terms of the number of features used in the hyperplane (those with nonzero optimal  $\omega$ -coefficients), the one which uses the less number of them is, as expected, the  $\ell_{\frac{4}{3}}$ -norm since it is *closest* to the  $\ell_1$ -norm which is known to be highly sparse.

## 6. CONCLUSIONS

The concept of classification margin is on the basis of the support vector machine technology to classify data sets. The measure of this margin has been usually done using Euclidean ( $\ell_2$ ) norm, although some alternative attempts can be found in the



literature, mainly with  $\ell_1$  and  $\ell_\infty$  norms. Here, we have addressed the analysis of a general framework for support vector machines with the family of  $\ell_p$ -norms with  $p > 1$ . Based on the properties and geometry of the considered models and norms we have derived a unifying theory that allows us to obtain new classifiers that subsume most of the previously considered cases as particular instances. Primal and dual formulations for the problem are provided, extending those already known in the literature. The dual formulation permits to extend the so-called *kernel trick*, valid for the  $\ell_2$ -norm case, to more general cases with  $\ell_p$ -norms,  $p > 1$ . The tools that have been used in our approach combine modern mathematical optimization and geometrical and tensor analysis. Moreover, the contributions of this paper are not only theoretical but also computational: different solution approaches have been developed and tested on four standard benchmark datasets from the literature. In terms of separation and classification no clear domination exists among the different possibilities and models, although in many cases the use of the standard SVM with  $\ell_2$ -norm is improved by other norms (as for instance the  $\ell_{4/3}$ ). Analyzing and comparing the different models may open new avenues for further research, as for instance the application to categorical data by introducing additional binary variables in our models as it has been recently done in the standard SVM model, see e.g.[6].

#### ACKNOWLEDGEMENTS

The first and second authors were partially supported by the project MTM2016-74983-C2-1-R (MINECO, Spain). The third author was partially supported by the project MTM2016-74983-C2-2-R (MINECO, Spain). The first author was also partially supported by the research project PP2016-PIP06 (Universidad de Granada) and the research group SEJ-534 (Junta de Andalucía).

#### REFERENCES

- [1] C. Bahlmann, B. Haasdonk, and H. Burkhardt (2002). *On-Line Handwriting Recognition with Support Vector Machines: A Kernel Approach*. In Proceedings of the Eighth International Workshop on Frontiers in Handwriting Recognition (IWFHR'02).
- [2] K.P. Bennett and E.J. Bredensteiner (2000). *Duality and Geometry in SVM Classifiers*. ICML 2000: 57-64
- [3] V. Blanco, J. Puerto, and S. El Haj Ben Ali, *Revisiting several problems and algorithms in continuous location with  $\ell_r$ -norms*, Computational Optimization and Applications **58** (2014), no. 3, 563–595.
- [4] Ch.J. Burges (1998). *A Tutorial on Support Vector Machines for Pattern Recognition*. Data Min. Knowl. Discov. **2**(2), 121-167.
- [5] E. Carrizosa and D. Romero-Morales (2013). *Supervised classification and mathematical optimization*. Computers & Operations Research, **40**(1), 150–165.
- [6] E. Carrizosa, A. Nogales-Gómez, D. Romero-Morales (2017). *Clustering categories in support vector machines*. Omega, **66**, 28–37.
- [7] J. D. Carroll and J. J. Chang (1970). *Analysis of individual differences in multidimensional scaling via an  $N$ -way generalization of Eckart-Young decomposition*, Psychometrika **35**, 283–319.
- [8] C.C. Chang and C.J. Lin (2011). *LIBSVM – A Library for Support Vector Machines*. ACM Transactions on Intelligent Systems and Technology **2**(3), 1–27. Available at <https://www.csie.ntu.edu.tw/~cjlin/libsvm/>
- [9] H. Chen, G. Li and L. Qi (2016). *SOS tensor decomposition: Theory and applications*. Communications in Mathematical Sciences **14** (8), 2073–2100.
- [10] P. Comon, G. Golub, L-H. Lim, Lek-Heng and B. Mourrain (2008). *Symmetric tensors and symmetric tensor rank*. SIAM Journal on Matrix Analysis and Applications **30**(3), 1254–1279

- [11] C. Cortes and V. Vapnik (1995). *Support-Vector Networks*. Mach. Learn. 20(3), 273–297.
- [12] C. Eckart and G. Young (1939). *A principal axis transformation for non-Hermitian matrices*. Bull. Amer. Math. Soc. 4, 118–121.
- [13] H. Edelsbrunner (1987). *Algorithms in combinatorial geometry*. Springer-Verlag, Berlin.
- [14] L. Gonzalez-Abril, F. Velasco, J.A. Ortega, and L. Franco (2011). *Support vector machines for classification of input vectors with different metrics*, Computers & Mathematics with Applications 61(9), 2874–2878.
- [15] T. Harris (2013). *Quantitative credit risk assessment using support vector machines: Broad versus Narrow default definitions*. Expert Syst. Appl. 40(11), 4404–4413.
- [16] D. Henrion, J. B. Lasserre, and J. Loefferberg (2009). *GloptiPoly 3: moments, optimization and semidefinite programming*. Optimization Methods and Software 24(4-5), 761–779.
- [17] C. J. Hillar and L.-H. Lim (2013). *Most tensor problems are NP-hard*. Journal of the ACM 60, 1–39.
- [18] J. Jiang, H. Wu, Y. Li, and R. Yu (2000). *Three-way data resolution by alternating slice-wise diagonalization (ASD) method*. Journal of Chemometrics 14, 15–36.
- [19] V. Kascelan, L. Kascelan, and M. Novovic Buric (2016). *A nonparametric data mining approach for risk prediction in car insurance: a case study from the Montenegrin market*. Economic Research-Ekonomska Istrazivanja 29(1), 545–558.
- [20] E. Kofidis and P. A. Regalia (2002). *On the best rank-1 approximation of higher-order supersymmetric tensors*. SIAM Journal on Matrix Analysis and Applications 23, 863–884.
- [21] K. Ikeda and N. Murata (2005). *Geometrical Properties of Nu Support Vector Machines with Different Norms*. Neural Computation 17(11), 2508–2529.
- [22] K. Ikeda and N. Murata (2005). *Effects of norms on learning properties of support vector machines*. ICASSP (5), 241–244
- [23] J.B Lasserre (2009). *Moments, Positive Polynomials and Their Applications*, Imperial College Press, London.
- [24] J. Lindenstrauss and L. Tzafriri (1977). *Classical Banach Spaces I, Sequence Spaces*. Ergebnisse der Mathematik und ihrer Grenzgebiete 92, Berlin: Springer-Verlag, ISBN 3-540-08072-4.
- [25] Y. Liu, H.H. Zhang, C. Park, and J. Ahn (2007). *Support vector machines with adaptive  $L_q$  penalty*. Comput. Stat. Data Anal. 51(12), 6380–6394.
- [26] A. Majid, S. Ali, M. Iqbal, and N. Kausar (2014). *Prediction of human breast and colon cancers from imbalanced data using nearest neighbor and support vector machines*. Computer Methods and Programs in Biomedicine 113(3), 792–808.
- [27] O.L. Mangasarian (1999). *Arbitrary-norm separating plane*. Oper. Res. Lett., 24 (1–2):15–23.
- [28] J. Mercer (1909). *Functions of positive and negative type and their connection with the theory of integral equations*. Philosophical Transactions of the Royal Society A, 209, 415–446.
- [29] J.P. Pedroso and N. Murata (2001). *Support vector machines with different norms: motivation, formulations and results*. Pattern Recognition Letters 22(12), 1263–1272.
- [30] M. Putinar (1993). *Positive Polynomials on Compact Semi-Algebraic Sets*, Ind. Univ. Math. J. 42: 969–984.
- [31] L. Qi and Y. Song (2014). *An even order symmetric B tensor is positive definite*. Linear Algebra and its Applications 457, 303–312.
- [32] S. Radhimeenakshi (2016). *Classification and prediction of heart disease risk using data mining techniques of Support Vector Machine and Artificial Neural Network*. 3rd International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, 3107–3111.
- [33] H. Chen and L. Qi (2015). *Positive definiteness and semi-definiteness of even order symmetric Cauchy tensors*. Journal of Industrial and Management Optimization 11(4), 1263–1274.
- [34] K.C. Toh, M.J. Todd, and R.H. Tutuncu (1999). *SDPT3 — a Matlab software package for semidefinite programming*, Optimization Methods and Software 11, 545–581.
- [35] V.N. Vapnik (1995). *The Nature of Statistical Learning Theory*. Springer-Verlag New York, Inc., New York, NY, USA.
- [36] V.N. Vapnik (1998). *Statistical Learning Theory*. Wiley-Interscience.

<sup>†</sup>DPT. QUANTITATIVE METHODS FOR ECONOMICS & BUSINESS, UNIVERSIDAD DE GRANADA  
*E-mail address:* [vblanco@ugr.es](mailto:vblanco@ugr.es)

<sup>‡</sup>DPT. STATISTICS & OR, UNIVERSIDAD DE SEVILLA  
*E-mail address:* [puerto@us.es](mailto:puerto@us.es)

\*DPT. STATISTICS & OR, UNIVERSIDAD DE CÁDIZ  
*E-mail address:* [antonio.rodriguezchia@uca.es](mailto:antonio.rodriguezchia@uca.es)