Journal of Machine Learning Research 21 (2020) 1-29 Submitted 9/18; Revised 6/19; Published 02/20

On ℓ_p-Support Vector Machines and Multidimensional Kernels

Víctor Blanco VBLANCO@UGR.ES
IEMath-GR, Universidad de Granada, SPAIN

Justo Puerto PUERTO@UGR.ES
IMUS, Universidad de Sevilla, SPAIN

Antonio M. Rodríguez-Chía ANTONIO.RODRIGUEZCHIA@UCA.ES
Dpt. Statistics & OR, Universidad de Cádiz, SPAIN

Editor: Mark Schmidt

Abstract

In this paper, we extend the methodology developed for Support Vector Machines (SVM) using the ℓ_2-norm (ℓ_2-SVM) to the more general case of ℓ_p-norms with $p > 1$ (ℓ_p-SVM). We derive second order cone formulations for the resulting dual and primal problems. The concept of kernel function, widely applied in ℓ_2-SVM, is extended to the more general case of ℓ_p-norms with $p > 1$ 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, where the dependence on the original data always appear as homogeneous polynomials. We adapt known solution algorithms to efficiently solve the primal and dual resulting problems and some computational experiments on real-world datasets are presented showing rather good behavior in terms of the accuracy of ℓ_p-SVM with $p > 1$.

Keywords: Support Vector Machines, Kernel functions, ℓ_p-norms, Mathematical Optimization.

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. Examples of such techniques are Support Vector Machines (SVM), Classification Trees and k Nearest Neighbours, among many others. In the last decades, SVM has become a popular methodology for supervised classification (Burges, 1998), due to their successful applications, as for instance in writing recognition (Bahlmann et al., 2002), evaluating insurance risks (Kascelan et al., 2016) or credit risk for lending (Harris, 2013) or detecting maligne/benigne tumors (Majid et al., 2014; Radhimeenakshi, 2016). SVM is a mathematical programming tool, that as other optimization-based approaches has helped to the successful development of supervised classification (see Bertsimas and Shioda, 2007, among others).

Support vector machine was originally developed by Vapnik (Vapnik, 1995, 1998) and Cortes and Vapnik (Cortes and Vapnik, 1995), and it 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 the training error and the model complexity. The most popular version of SVM is the one using the Euclidean norm to measure the distance. 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 $\frac{1}{2}\|\omega\|_2^2 + C \cdot R_{emp}(\mathcal{H})$, where $\|\cdot\|_2$ is the ℓ_2-norm and R_{emp} is an empirical measure of the risk incurred using the hyperplane $\mathcal{H}$ to classify the training data. 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, a number of extensions of the original model by using other norms different from the Euclidean one were addressed (see Bennet and Bredensteiner, 2000; Blanco et al., 2018; González et. al, 2011; Ikeda and Murata, 2005a,b; Pedroso and Murata, 2001; Xu et al., 2009). Among many other facts, it is well-known that using ℓ_1 or ℓ_∞ norms (as well as any other polyhedral norms) gives rise to SVM whose induced optimization problems are linear rather than quadratic, making it, in principle, possible solving in an exact way larger size instances (Mangasarian, 2006; Zhu et al., 2004). Moreover, it is also agreed that ℓ_1-SVM tends to generate sparse classifiers that can be more easily interpreted and reduce the risk of overfitting (Gaudioso et al., 2017). The use of more general norms, as the family of ℓ_p, $1 < p < +\infty$, has been also partially investigated (Carrizosa and Romero-Morales, 2013; González et. al, 2011; Liu et al., 2007). For this latter 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. As a matter of fact, it is an intriguing open question whether kernel-like transformations are possible within that more general framework. In the last years there have been important advances in the efficient representation of ℓ_p-norms to be used within optimization models (Blanco et al., 2014, 2018) and also relevant developments in the available solvers to deal with nonlinear optimization models. These facts allow us to handle ℓ_p-SVM with $p \neq 1, 2$, which from our computational experience could result in higher accuracy than the standard ℓ_1-and-ℓ_2-norm models for some datasets (see Section 5). In addition, it is known that the the usage of norms different from the ℓ_2 may improve the fitting of particular datasets to hyperplanes (Bi et al., 2003; Blanco et al., 2018; González et. al, 2011; Ikeda and Murata, 2005a,b; Pedroso and Murata, 2001; Zhu et al., 2004), may better exploit specific properties of those norms, as for instance sparsity thus avoiding redundant and noise features (Gaudioso et al., 2017; Zhu et al., 2004), and in some cases may also help to avoid extra tuning of the parameters of the models. Moreover, using these distance measures may enlarge the types of datasets where a linear separator is suitable.

Hence, the two main goals of this paper are: 1) Methodological goal: to develop a common framework for the analysis of ℓ_p-norm Support Vector Machine (ℓ_p-SVM) with general $p \in \mathbb{Q}$ and $p > 1$ as well as the extension of the Kernel theory, widely applied in ℓ_2-SVM, to the ℓ_p-SVM case; and, 2) Applicability goal: to show how using general norms, within the family of ℓ_p-norms, can improve the classification performance of SVM applied to some actual databases. In the methodological contribution, we shall develop the theory to understand primal and dual versions of SVM using these norms. In addition, we answer in the positive the existence of kernel-like transformations which 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 the application side, we show that the use of ℓ_p-SVM outperforms the classification results with respect to standard SVM applied to four well-known datasets: cleveland, housing, german credit and colon (see Section 5).

In our approach, we reduce all the primal problems to efficiently solvable Second Order Cone Programming (SOCP) problems. The respective dual problems are also SOCP. A thorough geometrical analysis of those problems allows for an extension of the kernel trick, applicable to the Euclidean case, to the more general ℓ_p-SVM. For that extension, we introduce the concept of multidimensional kernel. That analysis proves that the dual problems depend on the original data via homogeneous polynomials and linear functions. To prove that dependency, we rewrite the dual problems as *ad hoc* polynomial optimization problems, although this transformation is instrumental and obviously it does not increase the problem complexity. In addition, we derive a 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. Moreover, we develop two different approaches to find, in practice, the separating hyperplanes. The first one is based on solving some explicit SOCP problems, which can be derived using a rank-one decomposition of a tensor when a kernel function is used. The second one uses truncated expansions of functionals representable in Schauder spaces (Lindenstrauss and Tzafriri, 1977), 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* without specifying any a priori transformation. We report the results of an illustrative battery of computational experiments on four common real-world instances on the SVM field, which are comparable or superior to the standard ℓ_2-SVM.

The rest of the paper is organized as follows. In Section 2, we introduce ℓ_p-support vector machines. We derive primal and dual formulations for the problem, as a second order cone programming problem. In addition, starting from the dual formulation, we transform the problem into a convex polynomial optimization problem involving homogeneous polynomials, which gives us 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 ℓ_2-SVM to the more general case of ℓ_p-SVM with $p > 1$. There, we prove sufficient conditions for the existence of these objects by means of rank-one decompositions of higher dimensional tensors. In Section 4, we propose a methodology to solve the primal problem based on truncated expansion of functionals on the standard Schauder basis of multidimensional monomials. Finally, in Section 5, the results of some computational experiments on real-world datasets are reported.

2. ℓ_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 support vector machines when instead of measuring distances with the Euclidean norm, an ℓ_p-norm is used. In this case, the problem will be formulated as a mathematical programming problem whose objective function depends on the ℓ_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 $i \in \{1, \ldots, n\}$ are identified with the vector $\mathrm{x}_{i\cdot} \in \mathbb{R}^d$, while for $j \in \{1, \ldots, d\}$, the n observations about the j-th measure are denoted by $\mathrm{x}_{\cdot j} \in \mathbb{R}^n$. The ith individual is also classified into a class y_i, with $y_i \in \{-1, 1\}$, for $i = 1, \ldots, n$. The classification pattern is defined by $\mathbf{y} = (y_1, \ldots, y_n) \in \{-1, 1\}^n$.
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, i.e., the individuals of each class belong to each one of the halfspaces determined by the strip. Note that if the data are linearly separable, this constraint can be written as follows:

$$y_i(\omega^t \mathrm{x}_{i\cdot} + b) \geq 1, \quad i = 1, \ldots, n.$$

Since in many cases a linear separator is not possible, misclassification is allowed by adding a variable ξ_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, measured with the appropriate norm. (This misclassifying error is usually called the *hinge–loss* of the ith individual and represents the amount $\max\{0, 1 - y_i(\omega^t \mathbf{x}_{i\cdot} + b)\}$, for all $i = 1, \ldots, n$.) Then, the constraints to be satisfied are:

$$y_i(\omega^t \mathrm{x}_{i\cdot} + b) \geq 1 - \xi_i, \quad \forall i = 1, \ldots, n.$$

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 (Mangasarian, 1999), to obtain that whenever the distance measure is the ℓ_q-norm (with $\frac{1}{p} + \frac{1}{q} = 1$), the margin for ℓ_p-SVM is exactly $\frac{2}{\|\omega\|_p}$ (Recall that $\|\cdot\|_q$ is the dual norm of $\|\cdot\|_p$.). Henceforth, we assume without loss of generality that

$$q = \frac{r}{s} > 1, \text{ with } r, s \in \mathbb{Z}_+ \text{ and } \gcd(r, s) = 1. \tag{1}$$

Next, for the deviation of misclassified observations, one can take the summation of the slack variables ξ_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-of-sample data). Observe that the ℓ_q-norm distance from the observation $\mathrm{x}_{\mathrm{i}\cdot}$ to the hyperplane $\mathcal{H}^+$ (resp. $\mathcal{H}^-$) is given by $\frac{|\omega^t \mathrm{x}_{\mathrm{i}\cdot} + b - 1|}{\|\omega\|_p}$ $\left(\text{resp. } \frac{|\omega^t \mathrm{x}_{\mathrm{i}\cdot} + b + 1|}{\|\omega\|_p}\right)$ (Mangasarian, 1999) being then the misclassifying error ξ_i proportional to such a distance whenever $\mathrm{x}_{\mathrm{i}\cdot}$ is incorrectly classified.

Hence, the ℓ_p-SVM problem can be formulated as:

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

Note 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:

$$\begin{aligned}\min_{t,\xi,b,\omega} \quad & t + C\sum_{i=1}^n \xi_i && && (2)\\ \text{s.t.} \quad & y_i(\omega^t \mathrm{x}_{i\cdot} + b) \geq 1 - \xi_i, && \forall i = 1,\ldots,n, && (3)\\ & t \geq \|\omega\|_p^p, && && (4)\\ & \xi_i \geq 0, && \forall i = 1,\ldots,n, && (5)\\ & \omega \in \mathbb{R}^d, b, t \in \mathbb{R}, && && (6)\end{aligned}$$

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 $p = \frac{r}{r-s}$, see (1)), for $j = 1,\ldots,d$

$$\begin{aligned} & v_j \geq \omega_j && \forall j = 1,\ldots,d,\\ & v_j \geq -\omega_j && \forall j = 1,\ldots,d,\\ & t \geq \sum_{j=1}^d u_j, && \\ & u_j^{r-s} \geq v_j^r, && \forall j = 1,\ldots,d. && (7)\end{aligned}$$

Although the above formulation is still nonlinear, constraints in (7) can be efficiently rewritten as a set of second order cone constraints and then solved via interior point algorithms. The interested reader is referred to (Blanco et al., 2014, Lemma 1) for further details on the explicit and exact SOCP reformulation of (7) and the number of constraints and auxiliary variables needed to represent a p-order cone as an intersection of finitely many (possibly rotated) second-order cones, see Example 4.1 for a detailed description of this reformulation for the case $p = \frac{3}{2}$.

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 linear conic optimization problems, i.e., we assume without loss generality that $1 < p < +\infty$.

In the following result we state that a second reformulation of the problem is also possible using its Lagrangian dual formulation.

Proposition 2.1 *The Lagrangian dual problem of* (ℓ_p-SVM) *can be formulated as a Second Order Cone Programming problem.*

Proof Observe first that (ℓ_p-SVM) is convex and satisfies Slater's constraint qualification (Bertsekas, 1995) therefore it has zero duality gap with respect to the Lagrangian dual problem. Its Lagrangian function is:

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

where $\alpha_i \geq 0$ is the dual variable associated with constraints $y_i(\omega^t\mathrm{x}_{i\cdot} + b) \geq 1 - \xi_i$ and $\beta_i \geq 0$ the one for constraints $\xi_i \geq 0$, for $i = 1, \ldots, n$. The KKT optimality conditions for the problem read as:

$$\frac{\partial L}{\partial \omega_j} = p|\omega_j|^{p-1}\mathrm{sgn}(w_j) - \sum_{i=1}^n \alpha_i y_i x_{ij} = 0, \quad \forall j = 1, \ldots, d, \tag{8}$$

$$\frac{\partial L}{\partial b} = \sum_{i=1}^n \alpha_i y_i = 0, \tag{9}$$

$$\frac{\partial L}{\partial \xi_i} = C - \alpha_i - \beta_i = 0, \quad \forall i = 1, \ldots, n, \tag{10}$$

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

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

Hence, applying conditions (9) and (10), we obtain the following alternative expression of (LD):

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

In addition, from (8) and taking into account that $\frac{1}{p-1} = q - 1$, we can reconstruct the optimal value of ω_j for any $j = 1, \ldots, d$, as follows:

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

and then,

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

Observe that the above two expressions are well-defined for any $q \geq 1$, because by (8), we have that $\mathrm{sgn}(\omega_j)\Big(\sum_{i=1}^n \alpha_i y_i x_{ij}\Big) \geq 0$, i.e.,

$$\mathrm{sgn}(\omega_j) = \mathrm{sgn}\left(\sum_{i=1}^n \alpha_i y_i x_{ij}\right) =: \mathcal{S}_{\alpha,j}. \tag{11}$$

Hence:

$$\omega_j = \frac{1}{p^{q-1}}\mathcal{S}_{\alpha,j}\Big(\mathcal{S}_{\alpha,j}\sum_{i=1}^n \alpha_i y_i x_{ij}\Big)^{q-1}. \tag{12}$$

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 $0 \leq \alpha_i \leq C$, $\forall i = 1,\ldots,n$.

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

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

Introducing the variables δ_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 $q = \frac{r}{s}$, see (1), the problem above is equivalent to the following second order cone optimization problem:

$$\begin{aligned}
\max_{\alpha,u,\delta} \quad & \left(\frac{1}{p^q} - \frac{1}{p^{q-1}}\right)\sum_{j=1}^{d}u_j + \sum_{i=1}^{n}\alpha_i && , \qquad (\mathrm{SOC_{LD}}) \\
s.t. \quad & \delta_j \geq \sum_{i=1}^{n}\alpha_i y_i x_{ij}, && \forall j = 1,\ldots,d, \\
& \delta_j \geq -\sum_{i=1}^{n}\alpha_i y_i x_{ij}, && \forall j = 1,\ldots,d \\
& u_j^s \geq \delta_j^r, && \forall j = 1,\ldots,d, \\
& \sum_{i=1}^{n}\alpha_i y_i = 0, && \\
& 0 \leq \alpha_i \leq C, && \forall i = 1,\ldots,n.
\end{aligned}$$

Note that the above problem is a SOCP problem by simply rewriting the inequalities $u_j^s \geq \delta_j^r$ using (Blanco et al., 2014, Lemma 1) (that result provides an equivalent representation of the q-order cone as a finite intersection of possibly rotated second order cones). ■

The reader may observe that the problem ($\mathrm{SOC_{LD}}$) simplifies further for the cases of integer q ($q = r$ and $s = 1$), and especially if r is even, which results in:

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

3. Multidimensional kernels

As mentioned in the introduction, when nonlinear separators want to be computed to separate the classes, 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 Φ. 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 ℓ_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 answer in the positive the existence of kernel-like transformations for this class of problems and thus, to extend the idea of the kernel trick to ℓ_p-SVM.

In order to perform such an extension, we first derive an alternative reformulation of the Lagrangian dual problem in terms of linear functions and homogeneous polynomials in α. This instrumental reformulation of that problem allows us to get an explicit representation of the problem as a function of the input data. For this analysis we will concentrate on the case where $q = \frac{r}{s}$ with $s = 1$. In order to simplify the proposed formulations we denote by $\mathrm{H}_{\mathbf{y}} = \{\alpha \in [0, C]^n : \sum_{i=1}^{n} \alpha_i y_i = 0\}$, the feasible region of ($\mathrm{P_{LD}}$) where the dual variables α belong to.

Theorem 3.1 *There exists an arrangement of hyperplanes of* $\mathbb{R}^n$*, such that, in each of its full dimensional elements, both* $(\mathrm{P_{LD}})$ *and the separating hyperplane it induces can be expressed using homogeneous polynomials of the original input data.*

Proof Using (11), the first addend (without the constant term) of the objective function of $(\mathrm{P_{LD}})$ can be rewritten as:

$$\sum_{j=1}^{d}\left|\sum_{i=1}^{n}\alpha_i y_i x_{ij}\right|^r = \sum_{j=1}^{d}\left(\mathcal{S}_{\alpha,j}\sum_{i=1}^{n}\alpha_i y_i x_{ij}\right)^r. \tag{13}$$

The above is a piecewise multivariate polynomial in α (recall that y and x are input data) with a finite number of "branches" induced by the different signs of the terms $\mathcal{S}^r_{\alpha,j}$ for all $j = 1,\ldots,d$. Each branch is obtained by a particular vector $(\mathcal{S}_{\alpha,1},\ldots,\mathcal{S}_{\alpha,d})$ of values in $\{-1,+1\}$. The domains of these branches are defined by the arrangement induced by the set of homogeneous hyperplanes $\{\sum_{i=1}^n \alpha_i y_i x_{ij} = 0,\ j = 1,\ldots,d\}$. It is well-known that this arrangement has $O(2^d)$ full dimensional subdivision elements that we shall call *cells*, (see Edelsbrunner, 1987); all of them pointed, closed, convex cones. Since a generic cell is univocally defined by the signs of the expressions $\sum\limits_{i=1}^{n}\alpha_i y_i x_{ij}$ for $j = 1,\ldots,d$, denote by $\mathcal{C}(\mathrm{s}_1,\ldots,\mathrm{s}_d) = \{\alpha \in \mathbb{R}^n : \mathcal{S}_{\alpha,j} = \mathrm{s}_j,\ j = 1,\ldots,d\}$, with $\mathrm{s}_j \in \{-1,1\}$ for all $j = 1,\ldots,d$. Next, for all $\alpha \in \mathcal{C}(\mathrm{s}_1,\ldots,\mathrm{s}_d)$ 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 $(\mathrm{P_{LD}})$ and then to rewrite it as sum of monomials of the same degree. Indeed, denoting by

$$\mathrm{z}^\gamma := z_1^{\gamma_1}\cdots z_n^{\gamma_n},\ \text{for all } z = (z_1,\ldots,z_n) \in \mathbb{R}^n \text{ and } \gamma = (\gamma_1,\ldots,\gamma_n) \in \mathbb{N}^n, \tag{14}$$

we have the following equalities for any $\alpha \in \mathcal{C}(\mathrm{s}_1,\ldots,\mathrm{s}_d)$:

$$\sum_{j=1}^{d}\left|\sum_{i=1}^{n}\alpha_i y_i x_{ij}\right|^r = \sum_{j=1}^{d}\left(\mathrm{s}_j\sum_{i=1}^{n}\alpha_i y_i x_{ij}\right)^r = \sum_{j=1}^{d}\left(\sum_{\gamma\in\mathbb{N}^n_r}\mathrm{s}_j^r c_\gamma \alpha^\gamma y^\gamma \mathrm{x}^\gamma_{\cdot j}\right) = \sum_{\gamma\in\mathbb{N}^n_r} c_\gamma \alpha^\gamma y^\gamma \sum_{j=1}^{d}\mathrm{s}_j^r \mathrm{x}^\gamma_{\cdot j}$$

where $c_\gamma = \binom{r}{\gamma_1,\ldots,\gamma_n} = \dfrac{r!}{\gamma_1!\cdots\gamma_n!}$, and $\mathbb{N}^n_a := \{\gamma \in \mathbb{N}^n : \sum_{i=1}^n \gamma_i = a\}$, for any $a \in \mathbb{N}$.

The above discussion justifies the validity of the following representation of $(\mathrm{P_{LD}})$ within the cone $\mathcal{C}(\mathrm{s}_1,\ldots,\mathrm{s}_d)$:

$$\max_{\alpha}\left(\frac{1}{p^r} - \frac{1}{p^{r-1}}\right)\sum_{\gamma\in\mathbb{N}^n_r} c_\gamma \alpha^\gamma y^\gamma \sum_{j=1}^{d}\mathrm{s}_j^r \mathrm{x}^\gamma_{\cdot j} + \sum_{i=1}^{n}\alpha_i \tag{15}$$

$$\text{s.t. } \mathrm{s}_j\sum_{i=1}^{n}\alpha_i y_i x_{ij} \geq 0, \qquad \forall j = 1,\ldots,d, \tag{16}$$

$$\sum_{i=1}^{n}\alpha_i y_i = 0, \tag{17}$$

$$0 \leq \alpha_i \leq C, \qquad \forall i = 1,\ldots,n. \tag{18}$$

Finally, let us deduce the expression of the separating hyperplane as a function of the optimal solution of ($\mathrm{P_{LD}}$), $\bar{\alpha}$. For a particular $z \in \mathbb{R}^d$ the separating hyperplane is $\mathcal{H} = \{(z_1, \ldots, z_d) \in \mathbb{R}^d : \sum_{j=1}^d \omega_j z_j + b = 0\}$. Using (12), this hyperplane is given by:

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

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

$$\frac{1}{p^{r-1}} \sum_{\gamma \in \mathbb{N}^n_{r-1}} c_\gamma \bar{\alpha}^\gamma y^\gamma \sum_{j=1}^{d} \mathcal{S}^r_{\bar{\alpha},j} \mathrm{x}^\gamma_{\cdot j} z_j + b = 0.$$

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

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

and the result follows. ■

Note that the only observations $\mathrm{x}_{i\cdot}$ which matter in the reconstruction of the separating hyperplane, are those whose associated optimal dual variables, $\bar{\alpha}_i$, is strictly positive. These observations, as in its Euclidean counterpart, will be called the *support vectors*.

Note that the objective function in the alternative dual reformulation, (15), is concave in the domain defined by the sign pattern $(\mathrm{s}_1, \ldots, \mathrm{s}_d)$, since it is equivalent to the representation of absolute values in (13) once restricted to the corresponding cell defined by the signs, namely constraints (16), (17) and (18). Since ($\mathrm{SOC_{LD}}$) is a SOCP problem, then, the above representation is also a SOCP problem (it is nothing but a rewriting of the same problem restricted to a polyhedron in the α-space). In particular, the term $F(\alpha) := \displaystyle\sum_{\gamma \in \mathbb{N}^n_r} c_\gamma \alpha^\gamma y^\gamma \sum_{j=1}^{d} \mathrm{s}^r_j \mathrm{x}^\gamma_{\cdot j}$ is a second order cone representable function in the region $\Big\{\alpha \in \mathbb{R}^n_+ : \mathrm{s}_j \displaystyle\sum_{i=1}^{n} \alpha_i y_i x_{ij} \geq 0, \quad \forall j = 1, \ldots, d\Big\}$. Thus, the optimization problem (15)-(18), for a given accuracy, can be solved efficiently using standard convex optimization techniques. The most common option, due to the convexity of the problem, is to use modern proximal point algorithms which ensure convergence under very general conditions of convexity (Bolte et al., 2014). In addition, one can also resort to transform the problem to a SOCP problem so as to apply specific algorithms for this class of problems. This transformation is made explicit in Remark 3.4, once we have introduced our results on multidimensional kernels.

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 3.1 *For even r, the Lagrangian dual problem,* $(\mathrm{P_{LD}})$*, is given as:*

$$\max_{\alpha \in \mathrm{H}_{\mathbf{y}}} \left(\frac{1}{p^r} - \frac{1}{p^{r-1}}\right) \sum_{\gamma \in \mathbb{N}_r^n} c_\gamma \alpha^\gamma y^\gamma \sum_{j=1}^{d} \mathrm{x}_{\cdot j}^{\gamma} + \sum_{i=1}^{n} \alpha_i. \tag{19}$$

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 no longer needed in this case and the result follows. ■

Remark 3.1 *Observe that formulation* (15)-(18) *can be slightly modified to be valid for the case $q = \frac{r}{s}$ with $s \neq 1$ as follows:*

$$\begin{aligned}
\max_{\alpha, \delta} \quad & \left(\frac{1}{p^q} - \frac{1}{p^{q-1}}\right) \sum_{j=1}^{d} \delta_j + \sum_{i=1}^{n} \alpha_i \\
s.t. \quad & \sum_{\gamma \in \mathbb{N}_r^n} c_\gamma \alpha^\gamma y^\gamma \mathrm{s}_j^r \mathrm{x}_{\cdot j}^{\gamma} - \delta_j^s \leq 0, && \forall j = 1, \ldots, d, \\
& \mathrm{s}_j \sum_{i=1}^{n} \alpha_i y_i x_{ij} \geq 0, && \forall j = 1, \ldots, d, \\
& \sum_{i=1}^{n} \alpha_i y_i = 0, && \\
& 0 \leq \alpha_i \leq C, && \forall i = 1, \ldots, n, \\
& \delta_i \geq 0, && \forall j = 1, \ldots, d.
\end{aligned}$$

Observe that the Lagrangian dual problem $(\mathrm{P_{LD}})$ and the separating hyperplane it induces depend on the input data throughout sums of monomials of $\mathrm{x}_{\cdot j}$ of degree r. In other words, one does not need the specific values of $\mathbf{x}_{\cdot 1}, \ldots, \mathbf{x}_{\cdot d}$ to solve the problem, but only the sum of products $\mathrm{x}_{\cdot j}^{\gamma}$ for $\gamma \in \mathbb{N}_r^n$. This observation 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.

Let us now consider a data set $[\mathbf{x}] = (x_{1\cdot}, \ldots, x_{n\cdot})$ together with their classification pattern $\mathbf{y} = (y_1, \ldots, y_n)$ and $r \in \mathbb{N}$. Given $\Phi : \mathbb{R}^d \to \mathbb{R}^D$, the set-valued function $\mathrm{S}_\Phi : 2^{\mathrm{H}_{\mathbf{y}}} \to 2^{\{-1,1\}^D}$ ($2^{\mathrm{H}_{\mathbf{y}}}$ stands for the power set of $\mathrm{H}_{\mathbf{y}}$), is defined as:

$$\mathrm{S}_\Phi(R) := \Big\{ \mathrm{s} \in \{-1,1\}^D : \mathrm{s}_j = \operatorname{sgn} \left(\sum_{i=1}^{n} \alpha_i y_i \Phi_j(\mathrm{x}_{i\cdot}) \right)^r, \\ \text{for } j = 1, \ldots, D, \text{ for some } \alpha \in R \Big\}.$$

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

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

$$\mathrm{S}_\Phi(\mathrm{ri}(R_k)) = \{\mathrm{s}_{R_k}\} \textit{ for some } \mathrm{s}_{R_k} \in \{-1,1\}^D \textit{ and for all } k \in \mathcal{K}.$$

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

$$\mathcal{C}(\mathrm{s}_1, \ldots, \mathrm{s}_D) = \Big\{\alpha \in \mathrm{H}_{\mathbf{y}} : \mathrm{sgn}\Big(\sum_{i=1}^{n} \alpha_i y_i \Phi_j(\mathrm{x}_{i\cdot})\Big)^r = \mathrm{s}_j, \text{ for } j = 1, \ldots, D\Big\}$$

for any $\mathrm{s} \in \{-1,1\}^D$, and choosing $\{R_k\}_{k\in\mathcal{K}} = \Big\{\mathcal{C}(\mathrm{s}_1, \ldots, \mathrm{s}_D)\Big\}_{\mathrm{s}\in\{-1,1\}^D}$ (observe that each set of this subdivision is defined univocally by a vector $\mathrm{s} \in \{-1,1\}^D$).

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

$$\mathrm{K}[\mathbf{x}]_{R_k,\gamma,\lambda}(z) := \sum_{j=1}^{D} \mathrm{s}^r_{R_k,j} \Phi_j(\mathrm{x})^\gamma \Phi_j(z)^\lambda, \forall z \in \mathbb{R}^d, \forall k \in \mathcal{K}, \tag{20}$$

is called a r-order kernel function of Φ, where $\Phi_j(\mathrm{x}) := (\Phi_j(\mathrm{x}_{1\cdot}), \ldots, \Phi_j(\mathrm{x}_{\mathrm{n}\cdot}))$, see (14)*. For $k \in \mathcal{K}$, $\mathrm{K}[\mathbf{x}]_{R_k,\gamma,\lambda}(z)$ is called the k-th slice of the kernel function.*

The reader can observe that the objective function of $(\mathrm{P_{LD}})$ and the separating hyperplane obtained as a result of solving this problem can be rewritten for the Φ-transformed data using the kernel function (20).

Indeed, using (15) the objective function of the Lagrangian dual problem when using $\Phi(\mathrm{x})$ instead of x is:

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

Since the separating hyperplane is built for $\alpha^* \in R_{k^*}$, the optimal solution of an optimization problem, the expression (19) of this hyperplane is given by:

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

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

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

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

First of all, using that $\mathrm{K}[\mathbf{x}]_{R_k,\gamma,\lambda}(z)$ is a r-order kernel function of Φ, the problem (15)-(18) for a transformation of the original data via Φ, in each element, $k \in \mathcal{K}$, of a suitable Φ-subdivision can be written as:

$$\max_{\alpha} F_k(\alpha) := \left(\frac{1}{p^r} - \frac{1}{p^{r-1}}\right) \sum_{\gamma \in \mathbb{N}_r^n} c_\gamma \alpha^\gamma y^\gamma \mathrm{K}[\mathbf{x}]_{R_k,\gamma,0}(z) + \sum_{i=1}^n \alpha_i \tag{21}$$

$$\text{s.t. } M_j^k \alpha \geq 0, \qquad \forall j = 1, \ldots, m_k, \tag{22}$$

$$\sum_{i=1}^n \alpha_i y_i = 0, \tag{23}$$

$$0 \leq \alpha_i \leq C, \qquad \forall i = 1, \ldots, n. \tag{24}$$

Observe that the problem above is a reformulation of the Lagrangian dual problem, $(\mathrm{P}_{\mathrm{LD}})$, for the Φ-transformed data that only depends on the original data via the r-order kernel function of Φ and the suitable Φ-subdivision, and it can be seen as an extension of the kernel trick to ℓ_p-norms with $p > 1$.

In the particular case where Φ is the identity transformation, the above formulation becomes (15)-(18) whenever the suitable Φ-subdivision consists of the full dimensional elements of the arrangement of hyperplanes $\{\sum_{i=1}^n \alpha_i y_i x_{ij} = 0, \, j = 1, \ldots, d\}$. Furthermore, observe that F_k does not depend on z, since the degree, λ, of such a value is zero in that function.

Remark 3.2 *The general definition of kernel simplifies whenever r is even. In such a case, suitable Φ-subdivisions for any transformation $\Phi : \mathbb{R}^d \to \mathbb{R}^D$ are no longer needed since a single region can be chosen, $\mathrm{H}_{\mathbf{y}}$ (with $|\mathcal{K}| = 1$) where a single sign-pattern appears, $\mathrm{s}_j = (1, \ldots, 1)$. Then, the kernel function becomes:*

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

for $(\gamma, \lambda) \in \mathbb{N}_r^n$ and $\lambda \in \{0, 1\}$, but being it independent of α (since $\mathrm{S}_\Phi(\mathrm{H}_{\mathbf{y}}) = \{(1, \overset{D}{\ldots}, 1)\}$).

Remark 3.3 *For the Euclidean case ($r = 2$), note that usual definition of kernel is $\mathrm{K}(z, z') = \Phi(z)^t \Phi(z')$ 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 $\mathrm{K}(\mathrm{x}_{i_1\cdot}, \mathrm{x}_{i_2\cdot})$ for $i_1, i_2 = 1, \ldots, n$, whereas for classifying an arbitrary observation z, the unique expressions to be evaluated are of the form $\mathrm{K}(\mathrm{x}_{i\cdot}, z)$.*

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

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

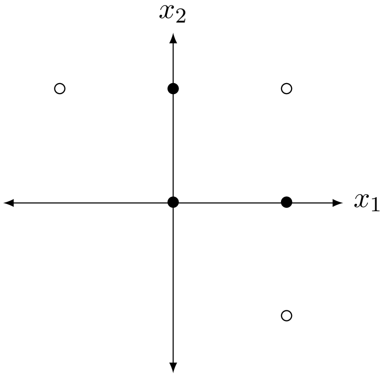

Figure 1: Points of Example 3.1 and their classification patterns.

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

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

for $(\gamma, \lambda) = \mathrm{e}_{i_1} + \mathrm{e}_{n+1}$, $i_1 = 1, \ldots, n$, *where* e_j *denotes the* j*-th canonical* $(n+1)$*-dimensional vector, for* $j = 1, \ldots, n$. *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 Φ.

Example 3.1 *Let us consider six points* $[\mathbf{x}] = \Big((0,0), (0,1), (1,0),\ (1,1),\ (1,-1),\ (-1,1)\Big)$ *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 $\Phi : \mathbb{R}^2 \to \mathbb{R}^3$, *defined as*

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

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

- $\mathrm{sgn}(\alpha_3 - \alpha_4 - \alpha_5 - \alpha_6) = \mathrm{sgn}(-\alpha_1 - \alpha_2) = -1$, *since* $\alpha_1, \alpha_2 \geq 0$,
- $\mathrm{sgn}(-\sqrt[r]{2}\alpha_4 + \sqrt[r]{2}\alpha_5 + \sqrt[r]{2}\alpha_6) = \mathrm{sgn}(\alpha_5 + \alpha_6 - \alpha_4)$.
- $\mathrm{sgn}(\alpha_2 - \alpha_4 - \alpha_5 - \alpha_6) = \mathrm{sgn}(-\alpha_1 - \alpha_3) = -1$, *since* $\alpha_1, \alpha_3 \geq 0$.

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, 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:

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

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

Thus,

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

being then:

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

For even r, because the r-th power of the signs do not affect to the expressions, the r-order kernel function of Φ is given by

$$\mathrm{K}[\mathbf{x}]_{R_k,\gamma,\lambda}(z) = \left(\mathrm{x}_{\cdot 1}^{\gamma} z_1^{\lambda} + \mathrm{x}_{\cdot 2}^{\gamma} z_2^{\lambda}\right)^2,$$

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 $\mathrm{H}_{\mathbf{y}}$ and a set of functions $\{\mathrm{K}[\mathbf{x}]_{R_k,\gamma,\lambda}\}_{k\in\mathcal{K}}$, 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 $\Phi : \mathbb{R}^d \longrightarrow \mathbb{R}^D$ such that, $\{R_k\}_{k\in\mathcal{K}}$ is a suitable Φ-subdivision and

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

where $\mathrm{S}_\Phi(\mathrm{ri}(R_k)) = \{s_{R_k}\}$ with $s_{R_k} \in \{-1, 1\}^D$.

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 3.2, 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 $\mathbb{L}_{j_1\ldots j_r} \in \mathbb{R}$ such that $\mathbb{L}_{j_1\ldots j_r} = \mathbb{L}_{j_{\sigma(1)}\cdots j_{\sigma(1)}}$, for any permutation σ of $\{1, \ldots, r\}$.

Lemma 3.1 *Let $\Phi : \mathbb{R}^d \longrightarrow \mathbb{R}^D$, $\hat{z} \in \mathbb{R}$ and let $\mathcal{S} = \{R_k\}_{k\in\mathcal{K}}$ be a suitable Φ-subdivision of $\mathrm{H}_{\mathbf{y}}$. Then, the k-th slice of any r-order kernel function of Φ 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:

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

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

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

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

Hence, $\mathbb{K}_{i_1 \ldots i_r} = \mathbb{K}_{i_{\sigma(1)} \ldots i_{\sigma(r)}}$, since the multi-indices constructed from (γ, λ) and (γ', λ') coincide. ■

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 *(Comon et al., 2008) Let $\mathbb{K}$ be a real r-order $(n+1)$-dimensional symmetric tensor. Then, there exists $\widehat{D} \in \mathbb{N}$, $v_{\cdot 1}, \ldots, v_{\cdot \widehat{D}} \in \mathbb{R}^{n+1}$ and $\psi \in \mathbb{R}^{\widehat{D}}$. such that $\mathbb{K}$ can be decomposed as*

$$\mathbb{K} = \sum_{j=1}^{\widehat{D}} \psi_j \; v_{\cdot j} \otimes \overset{r}{\cdots} \otimes v_{\cdot j} \, .$$

That is, $\mathbb{K}_{i_1 \cdots i_r} = \displaystyle\sum_{j=1}^{\widehat{D}} \psi_j v_{i_1 j} \cdots v_{i_r j}$ for any $i_1, \ldots, i_r \in \{1, \ldots, n+1\}$. 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 $\psi_1, \ldots, \psi_{\widehat{D}}$ are its eigenvalues.

The following result extends the classical Mercer's Theorem (Mercer, 1909) to r-order kernel functions.

Theorem 3.2 *Let $\{R_k\}_{k \in \mathcal{K}}$ be a subdivision of $\mathrm{H}_{\mathbf{y}}$ 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:*

$$\mathbb{K}^k = \sum_{j=1}^{\widehat{D}} \psi_{kj} v_{\cdot j} \otimes \overset{r}{\cdots} \otimes v_{\cdot j}, \; \forall k \in \mathcal{K},$$

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 $k \in \mathcal{K}$:*

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

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

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

$$\left\{ \begin{array}{lcl} \Phi_j(\mathrm{x}_{i\cdot}) & = & \sqrt[r]{\psi_j} v_{ij}, \text{ for } i = 1, \ldots, n, \\ \Phi_j(\ z\ ) & = & \sqrt[r]{\psi_j} v_{n+1,j}, \end{array} \right. \quad \text{for } j = 1, \ldots, \widehat{D},$$

which is well defined because of the nonnegativity of the eigenvalues ψ_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 Φ-subdivision (actually, for any Φ), since the signs are always positive (or the sign function is always 1).

Hence, for $(\gamma, \lambda) = \sum_{l=1}^{r} \mathrm{e}_{i_l}$,

$$\mathrm{K}[\mathbf{x}]_{R_k,\gamma,\lambda}(z) = \sum_{j=1}^{\widehat{D}} \Phi_j(\mathrm{x})^\gamma \Phi_j(z)^\lambda = \sum_{j=1}^{\widehat{D}} \psi_j v_{i_1 j} \cdots v_{i_r j} = \mathbb{K}^k_{i_1 \ldots i_r},$$

is a r-order kernel function of Φ.

Assume now that r is odd. Observe that $\mathrm{sgn}\Big(\sum_{i=1}^{n} \alpha_i y_i \Phi_j(\mathrm{x}_{i\cdot})\Big) = \mathrm{sgn}\Big(\sum_{i=1}^{n} \alpha_i y_i \sqrt[r]{\psi_j} v_{ij}\Big) = \mathrm{sgn}(\psi_{kj})$, being then

$$\mathrm{S}_\Phi(R_k) = \{(\mathrm{sgn}(\psi_{k1}), \ldots, \mathrm{sgn}(\psi_{k\widehat{D}}))\}.$$

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

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

$$\begin{array}{rcl} \mathrm{K}[\mathbf{x}]_{R_k,\gamma,\lambda}(z) & = & \displaystyle\sum_{j=1}^{\widehat{D}} \mathrm{sgn}\Big(\sum_{i=1}^{n} \alpha_i y_i \Phi_j(\mathrm{x}_{i\cdot})\Big)^r \Phi_j(\mathrm{x})^\gamma \Phi_j(z)^\lambda \\ & = & \displaystyle\sum_{j=1}^{\widehat{D}} \mathrm{sgn}\Big(\sum_{i=1}^{n} \alpha_i y_i \sqrt[r]{\psi_j} v_{ij}\Big)^r \psi_j v_{i_1 j} \cdots v_{i_r j} \\ & = & \mathbb{K}^k_{i_1 \ldots i_r} \end{array}$$

for $(\gamma, \lambda) = \sum_{l=1}^{r} \mathrm{e}_{i_l}$. Hence, $\mathbb{K}^k$ induces a kth-slice of a r-order kernel function of Φ. ■

The decomposition of symmetric 2-order n-dimensional tensors ($n \times n$ symmetric matrices) provided in Lemma 3.2, is equivalent to eigenvalue decomposition (Eckart and Young, 1939) and the symmetric tensor rank coincides with the usual rank of a matrix. Hence, for $r = 2$ (Euclidean case), the conditions of Theorem 3.2 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 (Hillar and Lim, 2013). 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 ψ that minimize $\|\mathbb{K} - \sum_{j=1}^{\widehat{D}} \psi_{kj} v_{\cdot j} \otimes \overset{r}{\cdots} \otimes v_{\cdot j}\|_2^2$. 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 (Carroll and Chang, 1970; Jiang et al., 2000; Kofidis. and Regalia, 2002) for further information about algorithms for decomposing real symmetric tensors.

In some interesting cases, the assumptions of Theorem 3.2 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 (Chen et al., 2016; Qi and Song, 2014) 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 $\mathrm{x}_{i\cdot} \neq \mathbf{0}$, for all $i = 1, \ldots, n+1$, it is well-known that the symmetric r-order $(n+1)$-dimensional tensor, $\mathbb{K}$, with entries:

$$\mathbb{K}_{i_1 \ldots i_r} = \frac{1}{\|\mathrm{x}_{i_1\cdot}\| + \cdots + \|\mathrm{x}_{i_r\cdot}\|}, \qquad i_1, \ldots, i_r = 1, \ldots, n+1,$$

for some norm $\|\cdot\|$ in $\mathbb{R}^d$, is a Cauchy-shaped tensor. Next, $\mathbb{K}$ is positive semidefinite (Chen and Qi, 2015), since $\|\mathrm{x}_{i\cdot}\| > 0$, for all $i = 1, \ldots, n$. Hence, by Theorem 3.2, it induces a r-order kernel function.

Also, one can extend general shapes of kernels which are widely used in ℓ_2-SVM. For instance, the exponential kernel may by extended to a ℓ_r-SVM by the following tensor:

$$\mathbb{K}_{i_1 \ldots i_r} = \exp\left(\sum_{j=1}^{d} x_{i_1 j} \cdots x_{i_r j}\right), \qquad i_1, \ldots, i_r = 1, \ldots, n+1.$$

Remark 3.4 *Observe also that Theorem 3.2 allows to exploit the structure of the dual problems* (15)–(18) *when using kernels. Let us consider the dual problem* (21)–(24) *at each cell in the arrangement which is known to be a SOCP. Under the hypotheses of Theorem 3.2, for each $k \in \mathcal{K}$ and $\gamma \in \mathbb{N}_r^n$, there exist $\widehat{D}$ and ψ_j, $v_{i_1 j}, \ldots, v_{i_r j}$, for $j = 1, \ldots, \widehat{D}$ such that:*

$$\mathrm{K}[\mathbf{x}]_{R_k,\gamma,0}(z) = \sum_{j=1}^{\widehat{D}} \sigma_j^r \psi_j v_{i_1 j} \cdots v_{i_r j},$$

where $\sigma_j = \begin{cases} 1 & \textit{if } r \textit{ is even,} \\ \operatorname{sgn}\Big(\sum_{i=1}^n \alpha_i y_i \sqrt[r]{\psi_j} v_{ij}\Big) & \textit{if } r \textit{ is odd,} \end{cases}$ *and $i_1, \ldots, i_r$ are such that $\gamma = \mathrm{e}_{i1} + \cdots + \mathrm{e}_{i_r}$.*

Hence:

$$\sum_{\gamma \in \mathbb{N}_r^n} c_\gamma \alpha^\gamma y^\gamma \mathrm{K}[\mathbf{x}]_{R_k,\gamma,0}(z) = \sum_{\gamma \in \mathbb{N}_r^n} c_\gamma \alpha^\gamma y^\gamma \sum_{j=1}^{\widehat{D}} (\sigma_j \sqrt[r]{\psi_j} v_{i_1 j}) \cdots (\sigma_j \sqrt[r]{\psi_j} v_{i_r j}) = \sum_{j=1}^{\widehat{D}} \left|\sum_{i=1}^{n} \alpha_i y_i \mathrm{v}_{ij}\right|^r$$

where $\mathrm{v}_{ij} = \sqrt[r]{\psi_j} v_{ij}$ *for all* $i = 1, \ldots, n$ *and* $j = 1, \ldots, \widehat{D}$.

Hence, the problem reads as:

$$\max_{\alpha} \left(\frac{1}{p^r} - \frac{1}{p^{r-1}}\right) \sum_{j=1}^{\widehat{D}} \left|\sum_{i=1}^{n} \alpha_i y_i \mathrm{v}_{ij}\right|^r + \sum_{i=1}^{n} \alpha_i$$
$$s.t.\ (22) - (24).$$

which is a r-order cone programming problem similar to $(\mathrm{P_{LD}})$*. Next, using again the results in (Blanco et al., 2014, Lemma 1) the explicit transformation to a SOCP follows.*

4. Solving the primal ℓ_p-SVM problem

Problem (21)-(24) is expressed as a polynomial optimization problem on α with linear constraints, within each element of the cells partition. Obviously, one can resort to the theory of moments, (Lasserre, 2009), to solve it building hierarchies of Semidefinite optimization 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. Another common option, due to the convexity of the problem, is to use modern proximal point algorithms which ensure convergence under very general conditions of convexity (Bolte et al., 2014). In addition, one can also resort to reformulate the problem to a SOCP problem (see Proposition 2.1 and Remark 3.4) so as to apply specific algorithms for this class of problems. In spite of that simplification, for general ℓ_p-norms, we would need to solve one problem in each full dimensional cell induced by the sign pattern which may be highly time consuming.

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 (Lindenstrauss and Tzafriri, 1977). In particular, $\mathcal{B} = \{\mathrm{z}^\gamma : \gamma \in \mathbb{N}^d\}$, 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 $\Phi : T \longmapsto \mathbb{R}^D$, there exists an expansion such that $\Phi(\mathrm{z}) = \sum_{j=1}^{\infty} \tau_j \mathrm{z}_j$, with $\tau_j \in \mathbb{R}$ and $\mathrm{z}_j \in \mathcal{B}$ for any $j = 1, \ldots, \infty$.

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 (ℓ_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.

Finally, it is worth mentioning that the above methodology can be also applied to the case of ℓ_1-SVM providing a way to deal with transformed data in the original space without mapping them.

Example 4.1 *We illustrate the primal methodology for the same dataset of Example 3.1. If the transformation provided in such an example is used to compute the $\ell_{\frac{3}{2}}$-SVM, we get the following primal formulation:*

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

Note that the constraint $t^2 \ge \|w\|_{\frac{3}{2}}^3$ can be equivalently rewritten, by introducing the auxiliary variables ζ_1, ζ_2 and ζ_3, as:

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

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

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

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 ξ are equal to zero. 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 $\mathbb{R}[z_1, \ldots, z_d]$. One may define the transformation $\Phi : \mathbb{R}^d \to \mathbb{R}^{\mathbb{N}^d}$, $\Phi_\gamma(\mathrm{z}) = \mathrm{z}^\gamma$, for each $\gamma \in \mathbb{N}^d$. Note that Φ projects out 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 γ component of Φ, $\Phi_\gamma(z)$, is a real number. Let us denote by $\mathbb{N}^d_{\le\eta} = \bigcup_{\mu \le \eta} \mathbb{N}^d_{\le\mu}$. Truncating the basis $\mathcal{B}$ by a given order $\eta \in \mathbb{N}$, we define the transformation $\Phi[\eta] : \mathbb{R}^d \to \mathbb{R}^{\mathbb{N}^d_{\le\eta}}$, $\Phi[\eta]_\gamma(\mathrm{z}) = \mathrm{z}^\gamma$, for each $\gamma \in \mathbb{N}^d_\mu$ with $\mu \le \eta$. Note that $\mathbb{R}^{\mathbb{N}^d_{\le\eta}}$ is a finite-dimensional space with dimension $\binom{d+\eta}{d}$.

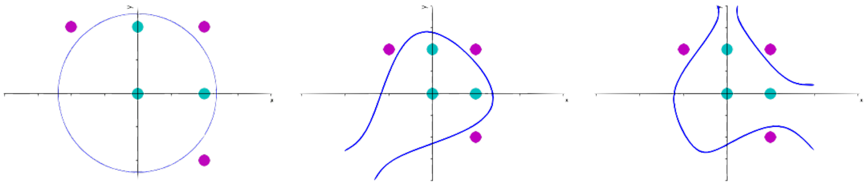

Figure 2: Separating curves with the three different settings (left: using the quadratic transformation, center: using $\Phi[3]$, right: : using $\Phi[4]$).

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 (ℓ_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.*

5. Experiments

We have performed a series of experiments to analyze the behavior of the proposed methods on some well-known real-world benchmark data sets. We have implemented the primal second-order cone formulation (2)–(6) and, in order to find non-linear separators, we consider the following two types of transformations on the data that can be identified with adequate truncated Schauder bases:

- $\Phi[\eta] : \mathbb{R}^d \to \mathbb{R}^{\mathbb{N}^d_\eta}$. Its components, $\Phi[\eta]_\gamma(\mathrm{z}) = z^\gamma$ for $\gamma \in \mathbb{N}^d_{\leq\eta}$, are the monomials (in d variables) up to degree η.
- $\widetilde{\Phi}[\eta] : \mathbb{R}^d \to \mathbb{R}^{\mathbb{N}^d_\eta}$, with $\widetilde{\Phi}[\eta]_\gamma(\mathbf{z}) = \exp(-\sigma\|z\|_2^2)\dfrac{\sqrt[r]{2\sigma}\mathbf{z}^\gamma}{\sqrt[r]{\gamma_1!\cdots\gamma_d!}}$, for $\mathbf{z} \in \mathbb{R}^d$, for $\gamma \in \mathbb{N}^d_{\leq\eta}$ 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 $\mathbb{G} : \mathbb{R}^{r\times d} \to \mathbb{R}$ defined as

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

is induced by using the transformation $\widetilde{\Phi}$, i.e., $\mathbb{G}[\mathbf{x}_{i_1\cdot}, \ldots, \mathbf{x}_{i_r\cdot}] = \lim\limits_{\eta\to\infty} \sum\limits_{\gamma\in\mathbb{N}^d_\eta} \widetilde{\Phi}[\eta]_\gamma$ where $\gamma = \sum_{l=1}^r \mathrm{e}_{i_l}$. Hence, the transformation $\widetilde{\Phi}[\eta]$, for a given η, is nothing but a truncated expansion of the generalized Gaussian operator $\mathbb{G}$.

We construct, in our experiments, ℓ_p-SVM separators for $p \in \{\frac{4}{3}, \frac{3}{2}, 2, 3\}$ by using η-order approximations with η 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) problems were 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 five classical data sets, widely used in the literature of SVM, that are listed in Table 1. They were obtained from the UCI Repository (Radhimeenakshi, 2016), LIBSVM Datasets(Chang and Lin, 2011) and Keel Datasets (Alcalá et al., 2011). There, one can find further information about each one 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
`page blocks`	5472	10	Keel 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:

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

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 ($\mathrm{ACC}^{\mathrm{Tr}}$) and the test data ($\mathrm{ACC}^{\mathrm{Test}}$), and also the average 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). Since our models depend on two parameters (C and η) and one more (σ) in case of using

	$\ell_{\frac{4}{3}}$				$\ell_{\frac{3}{2}}$				ℓ_2				ℓ_3			
η	ACC$^{\text{Tr}}$	ACC$^{\text{Test}}$	Time	%NonZ	ACC$^{\text{Tr}}$	ACC$^{\text{Test}}$	Time	%NonZ	ACC$^{\text{Tr}}$	ACC$^{\text{Test}}$	Time	%NonZ	ACC$^{\text{Tr}}$	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%	**83.48%**	0.01	100%	85.33%	83.15%	0.01	100%
2	94.02%	**82.57%**	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%	**80.23%**	2.65	99.66%
4	99.67%	76.56%	28	72.00%	99.67%	76.92%	22.5	81.88%	99.74%	**79.21%**	0.47	97.54%	100%	78.60%	17.56	99.31%
	`housing` dataset ($C = 64$)															
1	88.56%	**85.36%**	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%	**81.38%**	0.39	98.86%
3	98.60%	**80.95%**	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%	**79.99%**	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%	**76.20%**	0.02	99.58%	78.53%	**76.20%**	0.04	99.58%	78.53%	**76.20%**	0.05	99.58%	78.54%	**76.20%**	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%	**67.70%**	3.32	99.75%
3	100%	**71.90%**	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%	**82.14%**	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%
	`page blocks` dataset ($C = 64$)															
1	94.63%	**93.49%**	0.23	99.00%	94.51%	93.42%	0.45	99.00%	94.09%	92.95%	0.19	100.00%	93.52%	92.45%	0.53	100.00%
2	96.54%	95.80%	1.18	54.39%	96.40%	**95.89%**	1.10	65.76%	96.11%	95.54%	0.47	90.30%	95.71%	95.05%	1.15	98.48%
3	97.19%	96.03%	3.10	30.28%	97.19%	**96.11%**	4.56	37.52%	97.13%	96.03%	2.33	63.22%	96.85%	96.09%	4.89	97.97%
4	97.23%	**96.11%**	11.53	24.88%	97.23%	95.98%	24.36	31.11%	97.20%	95.94%	8.69	53.32%	97.06%	**96.11%**	13.96	96.91%

Table 2: Average results obtained using the monomial-based, $\Phi[\eta]$, transformation.

the transformation $\widetilde{\Phi}[\eta]$, we first perform a test to find the best choices for C and σ. For each dataset, we consider a part of the training sample and run the models by moving C and σ 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. We run the models for η ranging in $\{1, 2, 3, 4\}$ except for those datasets in which a perfect classification is found for the training sample for all the instances (`german credit` and `colon`). Tables 2 and 3 report, respectively, the average results for the feature space transformations $\Phi[\eta]$ and $\widetilde{\Phi}[\eta]$. We report the results on those choices of η that result in a good compromise between some improvement in accuracy and complexity on the problem solving by tuning it using a similar 10-fold strategy over the training sample. For instance, while for the datasets `cleveland`, `housing` and `page blocks` a degree up to $\eta = 4$ was considered, for `german credit` a degree of $\eta = 3$ already allows us to perfectly fit the data (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 η 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 η since the linear hyperplane is the one where we got the best results. However, such a situation changes when $\widetilde{\Phi}[\eta]$ 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.

	$\ell_{\frac{4}{3}}$				$\ell_{\frac{3}{2}}$				ℓ_2				ℓ_3			
η	ACC$^{\text{Tr}}$	ACC$^{\text{Test}}$	Time	%NonZ	ACC$^{\text{Tr}}$	ACC$^{\text{Test}}$	Time	%NonZ	ACC$^{\text{Tr}}$	ACC$^{\text{Test}}$	Time	%NonZ	ACC$^{\text{Tr}}$	ACC$^{\text{Test}}$	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%	**83.48%**	0.01	100%	85.22%	**83.48%**	0.01	100%
2	88.30%	**84.19%**	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%	**81.55%**	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%	**83.48%**	15.51	63.50%
	`housing` dataset ($C = 64$ and $\sigma = 2^{-6}$)															
1	88.56%	**85.36%**	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%	**83.53%**	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%	**80.21%**	0.25	88.30%	90.36%	79.95%	3.05	99.62%
4	90.80%	82.37%	14.43	4.23%	90.58%	**83.36%**	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%	**79.00%**	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%	**75.22%**	2.13	100%
3	76.72%	76.75%	57.01	3.39%	92.78%	**79.00%**	63.64	91.69%	96.36%	77.88%	2.75	99.82%	98.24%	76.57%	48.4	99.99%
	`colon` dataset ($C = 1$ and $\sigma = 2^{-4}$)															
1	100%	**82.14%**	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%
	`page blocks` dataset ($C = 64$ and $\sigma = 2^{-6}$)															
1	94.56%	**94.07%**	0.21	98.89%	94.44%	93.97%	0.41	98.89%	93.98%	93.44%	0.20	100.00%	93.43%	92.91%	0.54	100.00%
2	93.72%	**94.75%**	0.69	43.56%	93.49%	94.60%	0.89	48.48%	93.08%	94.27%	0.30	74.05%	92.58%	93.76%	0.76	95.45%
3	94.94%	**94.78%**	3.23	18.49%	94.70%	94.28%	3.31	23.18%	94.22%	94.31%	1.16	40.40%	93.87%	94.47%	2.88	93.75%
4	93.60%	93.81%	3.36	5.11%	93.48%	**94.02%**	4.01	6.62%	93.32%	93.99%	1.17	12.57%	92.98%	93.97%	4.84	89.45%

Table 3: Average results obtained using the Gaussian-based, $\widetilde{\Phi}[\eta]$, transformation.

Concerning the use of different norms, one can observe that there is not a clear 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 outperform the others. On the other hand, the ℓ_2-norm cases are solved in smaller computational times, since this norm is directly representable as a single quadratic constraint in our model. The remaining norms need to consider auxiliary variables and constraints which slightly increase the complexity for solving the problem, although, in all cases computational times to solve the instances are reasonable with respect to their size. The most time consuming instance, with 173 seconds, corresponds to the `german credit` dataset for $p = \frac{3}{2}$ and $\widetilde{\Phi}[3]$. Concerning the accuracy, we have run the classical *RBF* (ℓ_2-)SVM (see Table 4) using the `scikit-learn` library of Python and tuning the parameters efficiently. We report in Table 4 the accuracy averages of a 10-fold cross-validation (with the same 10 training and test sets used in our experiments above). There, one can see that our representation outperforms, in term of accuracy, the standard *RBF*-SVM for 4 out of the 5 datasets. In particular, for `housing`, `german credit`, `colon` and `page blocks`, one can find a ℓ_p-norm with $p \in \{\frac{3}{4}, \frac{3}{2}, 3\}$ for which the ℓ_p-SVM results in a better average accuracy for the test sample. Observe that the results obtained for RBF ℓ_2-SVM (Table 4) are different from those obtained for our ℓ_2-SVM (Table 3) since our Gaussian methodology is an approximation (by a truncated Schauder basis) of the RBF kernel.

In terms of the number of features used in the hyperplane (those with nonzero optimal ω-coefficients), the one which uses the least number of them is, as expected, the $\ell_{\frac{4}{3}}$-norm since it is the *closest* to the ℓ_1-norm which is known to be highly sparse. Also, as expected, the higher the p the more features are required in the ℓ_p-SVM.

As can be seen from table 2 and 3, it seems advisable to use the truncated polynomial bases and the truncated Gaussian transformations since their use implies, in most cases, an increase of the accuracy.

Dataset	$\mathrm{ACC}^{\mathrm{Tr}}$	$\mathrm{ACC}^{\mathrm{Test}}$
`cleveland`	85.85%	85.16%
`housing`	88.63%	85.35%
`german credit`	81.36%	77.30%
`colon`	100.00%	80.24%
`page blocks`	99.30%	92.31%

Table 4: Results for RBF ℓ_2-SVM.

Finally, as mentioned above, although the dual approach presented in this paper does not directly apply to the ℓ_1-norm, the Φ-approximations presented above, are suitable to solve the primal problem also with the ℓ_1-norm. The results obtained with this distance measure are reported in Table 5. One observes that the ℓ_1-norm usually provides more sparse classifiers, although, this does not mean to obtain better accuracies.

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 (ℓ_2) norm, although some alternative attempts can be found in the literature, mainly with ℓ_1 and ℓ_∞ norms. Here, we have addressed the analysis of a general framework for support vector machines with the family of ℓ_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 ℓ_2-norm case, to more general cases with ℓ_p-norms, $p > 1$. The tools that have been used in our approach combine modern mathematical optimization, geometry 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 ℓ_2-norm is outperformed 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 Carrizosa et al., 2017); or the incorporation of robust loss functions, as those presented in (Brooks, 2011), to the models.

	ℓ_1 (Φ)				$\ell_1(\widetilde{\Phi})$			
η	ACC$^{\mathrm{Tr}}$	ACC$^{\mathrm{Test}}$	Time	%NonZ	ACC$^{\mathrm{Tr}}$	ACC$^{\mathrm{Test}}$	Time	%NonZ
	`cleveland` dataset							
1	85.08%	**83.83%**	0.01	89.23%	84.78%	83.16%	0.01	85.38%
2	95.12%	80.24%	0.04	57.43%	89.25%	**85.15%**	0.05	32.38%
3	99.49%	74.59%	0.68	15.21%	92.59%	79.57%	0.57	10.68%
4	99.49%	74.59%	4.93	3.58%	84.31%	82.16%	3.46	0.57%
	`housing` dataset							
1	88.45%	**83.17%**	0.01	99.23%	88.45%	83.17%	0.01	99.23%
2	96.11%	79.22%	0.10	52.29%	91.59%	**83.53%**	0.09	29.14%
3	99.54%	79.80%	1.03	13.66%	95.43%	82.36%	1.05	9.43%
4	99.54%	79.80%	8.43	3.21%	92.47%	80.81%	8.96	1.19%
	`german credit` dataset							
1	78.53%	**76.20%**	0.13	99.58%	78.58%	**76.90%**	0.09	93.75%
2	92.99%	68.30%	2.41	81.26%	72.58%	70.30%	1.74	8.22%.
3	100.00%	69.70%	36.76	12.75%	85.17%	75.30%	24.93	5.05%
	`colon` dataset							
1	100.00%	**85.48%**	0.35	1.63%	100.00%	**85.48%**	0.36	1.63%
	`page blocks`							
1	94.78%	93.60%	0.19	77.00%	94.73%	94.19%	0.21	77.78%
2	96.61%	**95.97%**	0.77	35.50%	94.26%	95.13%	1.06	24.05%
3	97.34%	95.85%	5.26	17.62%	95.82%	**95.51%**	8.90	11.42%
4	97.34%	95.85%	28.05	5.03%	94.41%	94.63%	25.31	2.59%

Table 5: Average Results for the ℓ_1-norm.

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

J. Alcalá-Fdez, A. Fernandez, J. Luengo, J. Derrac, S. García, L. Sánchez, F. Herrera. *KEEL Data-Mining Software Tool: Data Set Repository, Integration of Algorithms and Experimental Analysis Framework.* Journal of Multiple-Valued Logic and Soft Computing 17:2-3 (2011) 255–287.

C. Bahlmann, B. Haasdonk, and H. Burkhardt (2002). *On-Line Handwriting Recognition with Support Vector Machines: A Kernel Approach.* In Proc. of the 8th Int. Workshop on Frontiers in Handwriting Recognition.

K.P. Bennett and E.J. Bredensteiner (2000). *Duality and Geometry in SVM Classifiers.* ICML 2000: 57-64

D. P. Bertsekas (1995). *Nonlinear programming.* Belmont, MA: Athena Scientific.

D. Bertsimas and R. Shioda (2007). *Classification and Regression via Integer Optimization.* Operations Research 55(2), 252–271.

J. Bi,K. Bennett, M. Embrechts, C. Breneman, and M. Song (2003). *Dimensionality reduction via sparse support vector machines.* Journal of Machine Learning Research, 3(Mar), 1229-1243.

V. Blanco, J. Puerto, and S. El Haj Ben Ali, *Revisiting several problems and algorithms in continuous location with ℓ_τ-norms*, Computational Optimization and Applications **58** (2014), no. 3, 563–595.

V. Blanco, J. Puerto and R. Salmerón (2018). *Locating hyperplanes to fitting set of points: A general framework*, Computers & Operations Research, 95, 172–193.

J. Bolte, S. Sabach and M. Teboulle (2014). *Proximal alternating linearized minimization for nonconvex and nonsmooth problems*, Math. Program. 146, 459-494.

J. P. Brooks (2011). *Support Vector Machines with the Ramp Loss and the Hard Margin Loss.* Operations Research 59(2), 467–479.

Ch.J. Burges (1998). *A Tutorial on Support Vector Machines for Pattern Recognition.* Data Min. Knowl. Discov. 2(2), 121-167.

E. Carrizosa and D. Romero-Morales (2013). *Supervised classification and mathematical optimization.* Computers & Operations Research, 40(1), 150–165.

E. Carrizosa, A. Nogales–Gómez, D. Romero-Morales (2017). *Clustering categories in support vector machines.* Omega, 66, 28–37.

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.

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/`

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.

H. Chen, G. Li and L. Qi (2016). *SOS tensor decomposition: Theory and applications.* Communications in Mathematical Sciences 14 (8), 2073–2100.

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

C. Cortes and V. Vapnik (1995). *Support-Vector Networks.* Mach. Learn. 20(3), 273–297.

C. Eckart and G. Young (1939). *A principal axis transformation for non-Hermitian matrices.* Bull. Amer. Math. Soc. 4. 118–121.

H. Edelsbrunner (1987). *Algorithms in combinatorial geometry.* Springer-Verlag, Berlin.

M. Gaudioso, E. Gorgone, M. Labbé, and A.M. Rodríguez-Chía (2017). *Lagrangian relaxation for SVM feature selection*, Computers & Operations Research, 87, 137–145.

L. González-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.

T. Harris (2013). *Quantitative credit risk assessment using support vector machines: Broad versus Narrow default definitions.* Expert Syst. Appl. 40(11), 4404–4413.

C. J. Hillar and L.-H. Lim (2013). *Most tensor problems are NP-hard.* Journal of the ACM 60, 1–39.

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.

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.

E. Kofidis and P. A. Regalia (2002). *On the best rank-1 approximation of higher-order supersymmetric tensors.* SIAM J. on Matrix Analysis and Applications 23, 863–884.

K. Ikeda and N. Murata (2005). *Geometrical Properties of Nu Support Vector Machines with Different Norms.* Neural Computation 17(11), 2508-2529.

K. Ikeda and N. Murata (2005). *Effects of norms on learning properties of support vector machines.* ICASSP (5), 241-244

J.B Lasserre (2009). *Moments, Positive Polynomials and Their Applications*, Imperial College Press, London.

J. Lindenstrauss and L. Tzafriri (1977). *Classical Banach Spaces I, Sequence Spaces.* Ergebnisse der Mathematik und ihrer Grenzgebiete 92, Berlin: Springer-Verlag.

Y. Liu, H.H. Zhang, C. Park, and J. Ahn (2007). *Support vector machines with adaptive Lq penalty.* Comput. Stat. Data Anal. 51(12), 6380-6394.

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.* Comp. Meth. & Progr.in Biomedicine 113(3), 792–808.

O.L. Mangasarian (1999). *Arbitrary-norm separating plane.* Oper. Res. Lett., 24 (1– 2):15–23.

Mangasarian, O. L. (2006). *Exact 1-norm support vector machines via unconstrained convex differentiable minimization.* Journal of Machine Learning Research 7, 1517-1530.

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.

J.P. Pedroso and N. Murata (2001). *Support vector machines with different norms: motivation, formulations and results.* Pattern Recognition Letters 22(12), 1263-1272.

L. Qi and Y. Song (2014). *An even order symmetric B tensor is positive definite.* Linear Algebra and its Applications 457, 303–312.

S. Radhimeenakshi (2016). *Classification and prediction of heart disease risk using data mining techniques of Support Vector Machine and Artificial Neural Network.* 3rd Int. Conf. on Computing for Sustainable Global Development, 3107–3111.

V.N. Vapnik (1995). *The Nature of Statistical Learning Theory.* Springer-Verlag New York.

V.N. Vapnik (1998). *Statistical Learning Theory.* Wiley-Interscience.

H. Xu, C. Caramanis, and S. Mannor (2009). *Robustness and regularization of support vector machines.* Journal of Machine Learning Research, 10(Jul), 1485-1510.

J. Zhu, S. Rosset, R. Tibshirani and T.J. Hastie (2004). $1-$*norm support vector machines*, In Advances in neural information processing systems (pp. 49-56).