Journal of Machine Learning Research 16 (2015) 323-369 Submitted 9/12; Revised 9/14; Published 3/15

Geometric Intuition and Algorithms for Eν–SVM

Álvaro Barbero ALVARO.BARBERO@UAM.ES
Department of Computer Science and Knowledge–Engineering Institute
Autonomous University of Madrid
Madrid, Spain

Akiko Takeda TAKEDA@MIST.I.U-TOKYO.AC.JP
Department of Mathematical Informatics
The University of Tokyo
Tokyo, Japan

Jorge López J.LOPEZ@UAM.ES
Department of Computer Science and Knowledge–Engineering Institute
Autonomous University of Madrid
Madrid, Spain

Editor: Sathiya Keerthi

Abstract

In this work we address the Eν–SVM model proposed by Pérez–Cruz *et al.* as an extension of the traditional ν support vector classification model (ν–SVM). Through an enhancement of the range of admissible values for the regularization parameter ν, the Eν–SVM has been shown to be able to produce a wider variety of decision functions, giving rise to a better adaptability to the data. However, while a clear and intuitive geometric interpretation can be given for the ν–SVM model as a nearest–point problem in reduced convex hulls (RCH–NPP), no previous work has been made in developing such intuition for the Eν–SVM model. In this paper we show how Eν–SVM can be reformulated as a geometrical problem that generalizes RCH–NPP, providing new insights into this model. Under this novel point of view, we propose the RapMinos algorithm, able to solve Eν–SVM more efficiently than the current methods. Furthermore, we show how RapMinos is able to address the Eν–SVM model for any choice of regularization norm $\ell_{p\geq 1}$ seamlessly, which further extends the SVM model flexibility beyond the usual Eν–SVM models.

Keywords: SVM, Eν–SVM, nearest point problem, reduced convex hulls, classification

1. Introduction

Let us address the classification problem of learning a decision function f from $\mathcal{X} \subseteq \mathbb{R}^n$ to $\{\pm 1\}$ based on m training samples (X_i, y_i), with $i \in M = \{1, ..., m\}$. We assume that the training samples are i.i.d., following the unknown probability distribution $P(X, y)$ on $\mathcal{X} \times \{\pm 1\}$.

Building on the well–known support vector machine (SVM) model developed in Cortes and Vapnik (1995), a variation of it, termed ν–SVM, was proposed in Schölkopf et al. (2000) as

$$\begin{aligned} \min_{W,b,\rho,\xi} \quad & \frac{1}{2}\|W\|_2^2 - \nu\rho + \frac{1}{m}\sum_{i\in M}\xi_i \\ \text{s.t.} \quad & \begin{cases} y_i\left(W\cdot X_i + b\right) \geq \rho - \xi_i, & i \in M, \\ \xi_i \geq 0, & i \in M, \\ \rho \geq 0. \end{cases} \end{aligned} \tag{1}$$

In this formulation the value of ν is made to lie in $[0,1]$, but actually there is a value $\nu_{\min} > 0$ such that if $\nu \in [0, \nu_{\min}]$, then we obtain the trivial solution $W = b = \rho = \xi = 0$. To tackle this, Pérez-Cruz et al. (2003) proposed generalizing (1) by allowing the margin ρ to be negative and enforcing the norm of W to be unitary:

$$\begin{aligned} \min_{W,b,\rho,\xi} \quad & -\nu\rho + \frac{1}{m}\sum_{i\in M}\xi_i \\ \text{s.t.} \quad & \begin{cases} y_i\left(W\cdot X_i + b\right) \geq \rho - \xi_i, & i \in M, \\ \xi_i \geq 0, & i \in M, \\ \|W\|_2^2 = 1. \end{cases} \end{aligned} \tag{2}$$

With this modification, a non–trivial solution can be obtained even for $\nu \in [0, \nu_{\min}]$. This modified formulation was called extended–ν-SVM (Eν-SVM), and has been shown to be able to generate a richer family of decision functions, thus producing better classification results in some settings. In addition to this, Takeda and Sugiyama (2008) arrived independently to the same model by minimizing the conditional value–at–risk (CVaR) risk measure, which is often used in finance. Letting the cost function be $f(W, b, X_i, y_i) = -y_i(W \cdot X_i + b)/\|W\|$, the CVaR risk measure is defined as the mean of the $(1-\nu)$–tail distribution of f for $i \in M$ (Rockafellar and Uryasev, 2002).

One of the advantages of the ν–SVM formulation (1) comes from its multiple connections to other well–known mathematical optimization problems, some of them allowing for intuitive geometric interpretations. A schematic of such connections is presented in Figure 1. Connections 1 and 2 were introduced in the pioneer work of Bennett and Bredensteiner (2000), showing how the SVM could be interpreted geometrically. Alternatively, and following the equivalence of the SVM and ν–SVM models (connection 3, shown in Schölkopf et al., 2000), Crisp and Burges (2000) arrived to the same geometrical problem (connections 4 and 5). Such problem, known in the literature as reduced convex hull nearest–point problem (RCH–NPP), consists of finding the closest points in the reduced convex hulls of the points belonging to the positive and negative classes. This can be formulated as

$$\begin{aligned} \min_{\lambda_+,\lambda_-} \quad & \frac{1}{2}\left\| \sum_{i\in M_+}\lambda_i X_i - \sum_{i\in M_-}\lambda_i X_i \right\|_2^2 \\ \text{s.t.} \quad & \begin{cases} \sum_{i\in M_+}\lambda_i = \sum_{i\in M_-}\lambda_i = 1, \\ 0 \leq \lambda_i \leq \eta,\ i \in M, \end{cases} \end{aligned} \tag{3}$$

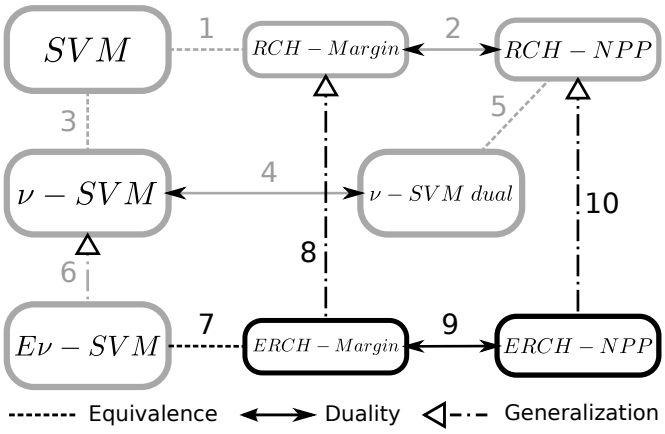


Figure 1: Relationships between the SVM, ν–SVM and other mathematical optimization problems. Connections and problems in gray were previously known, while connections and models in black are introduced in this paper.

where we denote $M_{\pm} = \{i : y_i = \pm 1\}$, and η is the reduction coefficient of the reduced convex hulls. A specific value of ν in (1) corresponds to a specific value of η in (3). Broadly speaking, the bigger ν is, the smaller η is, and the more the hulls shrink towards their barycenters.

Using the same notation, the intermediate RCH–Margin formulation in Figure 1 has the following form:

$$\min_{W,\alpha,\beta,\xi} \quad \frac{1}{2}\|W\|_2^2 + \beta - \alpha + \eta \sum_{i \in M} \xi_i \tag{4}$$

$$\text{s.t.} \quad \begin{cases} W \cdot X_i \geq \alpha - \xi_i, & i \in M_+, \\ W \cdot X_i \leq \beta + \xi_i, & i \in M_-, \\ \xi_i \geq 0, & i \in M. \end{cases}$$

At the light of these relationships and the fact that Eν–SVM is essentially a generalization of ν–SVM (connection 6, Pérez-Cruz et al., 2003), it seems natural to assume that similar connections and geometric interpretations should exist for Eν–SVM. Nevertheless, no work has been previously done along this line. Therefore, in this paper we exploit these known ν–SVM connections to develop a novel geometric interpretation for the Eν–SVM model. We will show how similar connections can be proved for Eν–SVM, and how this provides a better insight into the mathematical problem posed by this generalized model, allowing us to develop a new algorithm for Eν–SVM training.

On top of this, we demonstrate how the Eν–SVM formulation allows to extend the SVM models through the use of general $\ell_{p\geq 1}$–norm regularizations, instead of the usual ℓ_2–norm regularization. Previously, SVM models with other particular values of p have been proposed, such as ℓ_1–SVM by Zhu et al. (2003) or ℓ_∞–SVM in Bennett and Bredensteiner

(2000), acknowledging the usefulness of different ℓ_p–norms to enforce different degrees of sparsity in the model coefficients. Some work has also been done in approximating the NP–hard non–convex non–continuous ℓ_0–norm within SVM models, by methods such as iterative reweighing of ℓ_1–SVM models (Shi et al., 2011) or through expectation maximization in a Bayesian approach (Huang et al., 2009), and also in the context of least–squares support vector machines (López et al., 2011b). In spite of this, to date no efficient implementation seems to have been offered for the general $\ell_{p\geq 1}$–SVM. Similarly, no methods have been proposed either to solve an equivalent $\ell_{p\geq 1}$ version of the ERCH–NPP.

The contributions of this work on these matters are the following:

- We show how the Eν–SVM problem (2) is equivalent to an extended version of the reduced convex hull margin (RCH–Margin) problem (connections 7 and 8 in Figure 1).
- We introduce the extended reduced convex hulls nearest–point problem (ERCH–NPP), which is both a dual form of the Eν–SVM (connection 9) and a generalization of RCH–NPP (connection 10).
- For the case when the reduced convex hulls do not intersect, we show how ERCH–NPP can be reduced to the RCH–NPP problem.
- For the intersecting case we analyse how the problem becomes non–convex, and propose the RapMinos algorithm, which uses the acquired geometric insight to find a local minimum of ERCH–NPP faster than the currently available Eν–SVM solvers.
- All derivations are performed for the general $\ell_{p\geq 1}$ regularization, thus boosting the Eν–SVM model capability even further, and also providing means to solve RCH–NPP for such range of norms.
- A publicly available implementation of RapMinos is provided.

The rest of the paper is organized as follows: Section 2 describes the recasting of (2) as a geometrical problem. Section 3 shows that this geometrical problem is in fact a generalization of the standard RCH–NPP problem (3), able to find non–trivial solutions even in the case where the convex hulls intersect. In Section 4 we analyse the structure of the optimization problem posed by the ERCH–NPP problem. Based on this, Section 5 develops the RapMinos algorithm and shows its theoretical properties, while in Section 6 we present experimental results on its practical performance. Finally, Section 7 discusses briefly the results obtained and related future work.

2. Geometry in Eν-SVM

In this section we will introduce the geometric ideas behind Eν–SVM (2) by proving connections 7 and 9 in Figure 1, thus arriving to the ERCH–NPP problem. We also generalize its formulation not only to cover the ℓ_2–norm W regularization, but an arbitrary ℓ_p–norm with $p \geq 1$.

To begin with, let us define the ERCH–Margin (extended reduced–convex–hull margin) problem and its connections with Eν–SVM.

Proposition 1 *The ERCH–Margin (extended reduced–convex–hull margin) problem, defined as*

$$\min_{W:\|W\|_p=1} \quad \min_{\alpha,\beta,\xi} \quad \beta - \alpha + \eta \sum_{i\in M} \xi_i \tag{5}$$
$$s.t. \quad \begin{cases} W \cdot X_i \geq \alpha - \xi_i, & i \in M_+, \\ W \cdot X_i \leq \beta + \xi_i, & i \in M_-, \\ \xi_i \geq 0, & i \in M. \end{cases}$$

is equivalent to the Eν–SVM problem (connection 7 in Figure 1).

Proof Take (2) and multiply its objective function by $2/\nu$ [1]. Let us also consider the ℓ_p–norm, and separate the constraint $\|W\|_p = 1$ from the problem, obtaining:

$$\min_{W:\|W\|_p=1} \quad \min_{b,\rho,\xi} \quad -2\rho + \frac{2}{\nu m} \sum_{i\in M} \xi_i \tag{6}$$
$$\text{s.t.} \quad \begin{cases} y_i \left(W \cdot X_i + b\right) \geq \rho - \xi_i, & i \in M, \\ \xi_i \geq 0, & i \in M. \end{cases}$$

Denoting now $\eta = 2/(\nu m)$, $\alpha = \rho - b$ and $\beta = -\rho - b$, direct substitution makes the above problem become the ERCH–Margin problem. ■

The geometry behind this formulation is summarized in Figure 2. There we have a feasible estimate (W, α, β, ξ) which gives two parallel hyperplanes: $W \cdot X = \alpha$ and $W \cdot X = \beta$. We are seeking to optimize two conflicting goals: on the one hand we want to maximize the signed distance between both hyperplanes, given by $\alpha - \beta$, and on the other hand we want the hyperplane $W \cdot X = \alpha$ to leave as many positive points as possible to its left. The same is applicable to the hyperplane $W \cdot X = \beta$, which should leave as many negative points as possible to its right. In the configuration illustrated, preference has been given to correct classification, so that the hyperplanes "cross", and $\beta > \alpha$. Thus, the signed distance between the hyperplanes is negative in this case.

In the general case, the trade–off between these two conflicting goals is regulated by the penalty factor $\eta = 2/(\nu m)$. The slack variables ξ_i allow for errors when the hyperplanes do not leave the points to their proper side. The penalty factor keeps the errors at bay, so finally we reach a compromise between separation of the hyperplanes and correct classification.

We now move one step further and define the ERCH–NPP problem and its connection with ERCH–Margin.

Proposition 2 *The ERCH–NPP (extended reduced–convex–hull nearest–point problem) problem, defined as*

1. Note that this precludes the use of $\nu = 0$, but in practice such a value is not interesting, since (2) would only minimize the errors, which tends to overfitting.

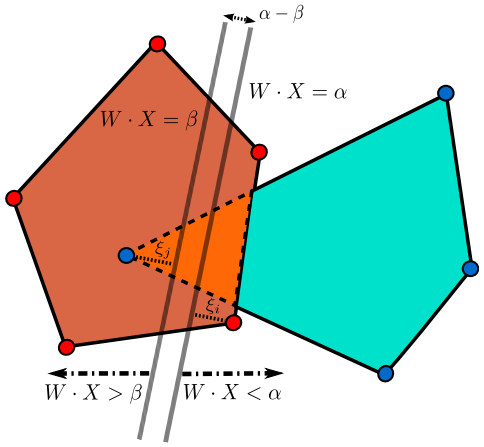


Figure 2: Illustration of the ERCH–Margin problem. The extreme positive (negative) points are printed in red (blue). The current estimate gives two parallel hyperplanes $W \cdot X = \alpha$ and $W \cdot X = \beta$ that try to separate the two classes as well as possible, while keeping far from each other. Errors are quantified by slack variables ξ_i, with two examples highlighted.

$$\min_{W:\|W\|_p=1} \quad \max_{X_+ \in \mathcal{U}_+, X_- \in \mathcal{U}_-} \quad W \cdot X_- - W \cdot X_+. \tag{7}$$

with reduced convex hulls

$$\mathcal{U}_\pm = \left\{ \sum_{i \in M_\pm} \lambda_i X_i : \sum_{i \in M_\pm} \lambda_i = 1,\ 0 \le \lambda_i \le \eta \right\},$$

is the dual problem of ERCH–Margin (connection 9 in Figure 1).

Proof The Lagrangian for the inner minimization problem in ERCH–Margin (5) reads

$$\begin{aligned} \mathcal{L} \ = \ & \beta - \alpha + \eta \sum_{i \in M} \xi_i - \sum_{i \in M_+} \lambda_i \left(W \cdot X_i - \alpha + \xi_i \right) \\ & + \sum_{i \in M_-} \lambda_i \left(W \cdot X_i - \beta - \xi_i \right) - \sum_{i \in M} \mu_i \xi_i, \end{aligned} \tag{8}$$

where we introduced the Lagrange multipliers $\lambda_i \ge 0, \mu_i \ge 0,\ i \in M$, associated to the inequality constraints of (5). Differentiating with respect to the variables being minimized and equating to zero gives

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial \alpha} &= -1 + \sum_{i \in M_+} \lambda_i = 0 \quad \Rightarrow \quad \sum_{i \in M_+} \lambda_i = 1, \\
\frac{\partial \mathcal{L}}{\partial \beta} &= 1 - \sum_{i \in M_-} \lambda_i = 0 \quad \Rightarrow \quad \sum_{i \in M_-} \lambda_i = 1, \\
\frac{\partial \mathcal{L}}{\partial \xi_i} &= \eta - \lambda_i - \mu_i = 0 \quad \Rightarrow \quad 0 \leq \lambda_i \leq \eta,\ i \in M.
\end{aligned}$$

Substituting all the above in the Lagrangian (8) yields the partial dual formulation of (5):

$$\min_{W:\|W\|_p=1} \quad \max_{\lambda} \quad \sum_{i \in M_-} \lambda_i W \cdot X_i - \sum_{i \in M_+} \lambda_i W \cdot X_i \tag{9}$$
$$\text{s.t.} \quad \begin{cases} \sum_{i \in M_+} \lambda_i = \sum_{i \in M_-} \lambda_i = 1, \\ 0 \leq \lambda_i \leq \eta,\ i \in M. \end{cases}$$

Now, considering the constraints of (9) and problem (3), we are confined to the reduced convex hulls whose reduction coefficient is in this case $\eta = 2/(\nu m)$. If we have $2/(\nu m) \geq 1$, we just work in the standard convex–hulls of both subsamples. By making use of the reduced convex hulls $\mathcal{U}_\pm$ and defining $X_\pm = \sum_{i \in M_\pm} \lambda_i X_i$, problem (9) can be written more succinctly as

$$\min_{W:\|W\|_p=1} \quad \max_{X_+ \in \mathcal{U}_+, X_- \in \mathcal{U}_-} \quad W \cdot X_- - W \cdot X_+,$$

which is ERCH–NPP.

■

Once we know we are working with reduced convex hulls, further geometrical intuition can be given on what we are doing. Recall that the quantity $(W \cdot X_0 + b)/\|W\|_p$ gives the signed distance from a specific point X_0 to the hyperplane $W \cdot X + b = 0$, in terms of the ℓ_p–norm. Note that in this case we always have unitary W vectors. Since we only care about the orientation of the solution hyperplane (W, b) and not about its magnitude, problem (7) can be rewritten as

$$\max_{W,b} \quad \min_{X_+ \in \mathcal{U}_+, X_- \in \mathcal{U}_-} \quad \frac{W \cdot X_+ + b}{\|W\|_p} - \frac{W \cdot X_- + b}{\|W\|_p}, \tag{10}$$

so that we can regard that Eν-SVM finds a solution that maximizes the margin, where by "margin" we mean the smallest signed distance between the two reduced convex hulls.

There are two cases depending on the value of the reduction coefficient $2/(\nu m)$:

- If the coefficient is small enough, the reduced convex hulls will not intersect, so there exists some hyperplane W producing a perfect separation between them. Therefore, $(W^* \cdot X_+^* + b^*)/\|W^*\|_p > 0$ and $(W^* \cdot X_-^* + b^*)/\|W^*\|_p < 0$ must hold at optimality.
- If it is large enough, they will intersect, so there is no W producing perfect separation. Therefore, it is $(W^* \cdot X_+^* + b^*)/\|W^*\|_p < 0$ and $(W^* \cdot X_-^* + b^*)/\|W^*\|_p > 0$ that hold at optimality.

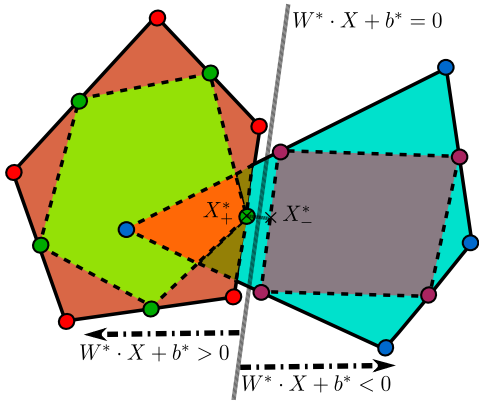


Figure 3: Case where the reduced convex hulls do not intersect ($\mu = 1/2$). The color convention is the same as in Figure 2, whereas the extreme points of the positive (negative) reduced hulls are printed in green (purple). The optimal solution is given by W^*, b^*, X_+^* and X_-^*. Observe that X_+^* (X_-^*) lies in the positive (negative) side of the hyperplane.

In the following section we will see how in the first case the problem can be reduced to the standard RCH–NPP problem, while the second case cannot be captured by such problem. This will lead to the conclusion that ERCH–NPP is a generalization of RCH–NPP (connection 10 in Figure 1), and that ERCH–Margin is a generalization of RCH–Margin (connection 8).

3. Relationship with RCH–NPP

Here we will see that ERCH–NPP (9) is in fact a generalization of RCH–NPP (3). Using the notation of the previous section, (3) can be expressed as

$$\min_{X_+ \in \mathcal{U}_+, X_- \in \mathcal{U}_-} \frac{1}{2} \|X_+ - X_-\|_q^q \quad \equiv \quad \min_{X_+ \in \mathcal{U}_+, X_- \in \mathcal{U}_-} \|X_+ - X_-\|_q \,, \tag{11}$$

where the reduction coefficient in $\mathcal{U}_\pm$ is $\eta = 2/(\nu m)$, and we again allow the use of a general ℓ_q–norm with $q \geq 1$ to measure the distance between the hulls [2].

2. While we acknowledge the interest in $q < 1$ norms in the field of Machine Learning, the use of such norms introduces an additional level of non–convexity into the problem, and thus is out of the scope of this paper.

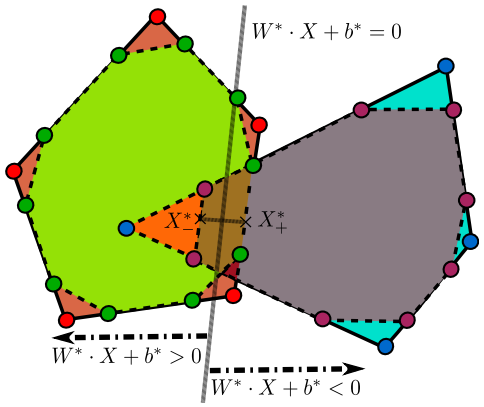


Figure 4: Case where the reduced convex hulls intersect ($\mu = 3/4$), with the same color and solution convention than in Figure 3. Observe that X_+^* (X_-^*) lies now in the negative (positive) side of the hyperplane.

If the reduction parameter $\eta = 2/(\nu m)$ is not small enough, the classes might overlap as in Figure 4, and (11) thus generates the trivial solution $X_+^* = X_-^*$, so that $W^* = 0$. The same happens with ν–SVMs, where ν must be large enough to obtain meaningful solutions. What we intend to show next is that, exactly as Eν–SVM extended ν–SVM to allow for all the range of possible values of ν (that is, $\nu \in (0, \nu_{\max}]$, with $\nu_{\max} = 2\min\{|M_+|, |M_-|\}/m$), ERCH also extends RCH to allow for all the possible values for η.

To this aim, first we show the following lemma, whose is based on the fact that if the hulls do not intersect, any solution with $\|W\|_p < 1$ is actually worse than the one obtained by trivially rescaling W so that $\|W\|_p = 1$. That is to say, relaxing the constraint in such a way does not modify the solution of the optimization, since the optimum is guaranteed to remain at the same place.

Lemma 3 *If the reduced convex hulls do not intersect, we can replace the constraint* $\|W\|_p = 1$ *in* (5) *with* $\|W\|_p \leq 1$.

Proof As was discussed above, if the reduced convex hulls do not intersect, a hyperplane W^* and a bias b^* exist such that $W^* \cdot X_+ + b^* > 0 \; \forall \; X_+ \in \mathcal{U}_+$, $W^* \cdot X_- + b^* < 0 \; \forall \; X_- \in \mathcal{U}_-$. Therefore, at the optimum of (7) and (9) the value of the inner maximum must be negative.

Since the inner problem of (9) is the dual of the inner problem of (5) and both problems are convex (linear, in fact), by strong duality the value of their objective functions is equal at the optimum (Rockafellar, 1970; Luenberger and Ye, 2008). Hence, the inner minimum of (5) must be negative as well. Therefore, for any optimal solution $(W^*, \alpha^*, \beta^*, \xi^*)$ we get the optimal objective value

$$\mathcal{P}^* = \beta^* - \alpha^* + \eta \sum_{i \in M} \xi_i^* < 0.$$

To see that we can replace the constraint $\|W\|_p = 1$ with $\|W\|_p \leq 1$, let us suppose an optimal solution $(W^*, \alpha^*, \beta^*, \xi^*)$ such that $\|W^*\|_p < 1$. We can then build another solution $(W', \alpha', \beta', \xi')$, with $W' = W^*/\|W^*\|_p$, $\alpha' = \alpha^*/\|W^*\|_p$, $\beta' = \beta^*/\|W^*\|_p$ and $\xi' = \xi^*/\|W^*\|_p$.

This solution is obviously feasible, because the constraints of (5) hold. Moreover, $\|W'\|_p = 1$ and the objective value is now

$$\mathcal{P}' = \beta' - \alpha' + \eta \sum_{i \in M} \xi'_i = \frac{\mathcal{P}^*}{\|W^*\|_p} < \mathcal{P}^*,$$

where the last inequality holds because $\mathcal{P}^* < 0$ and $\|W^*\|_p < 1$. We are minimizing in (9), so this new solution $(W', \alpha', \beta', \xi')$ is actually better than $(W^*, \alpha^*, \beta^*, \xi^*)$, which contradicts the supposed optimality of the latter. Therefore, we can safely replace $\|W\|_p = 1$ with $\|W\|_p \leq 1$. ■

The following definition and remark will also be used:

Definition 4 *The convex conjugate $\hat{f} : \hat{X} \to \mathbb{R} \cup +\infty$ of a functional $f : X \to \mathbb{R} \cup +\infty$ is $\hat{f}(\hat{x}) = \sup_{x \in X} \{\hat{x} \cdot x - f(x)\} = -\inf_{x \in X} \{f(x) - \hat{x} \cdot x\}$, where $\hat{X}$ denotes the dual space to X and the dot product operation (dual pairing) is a function $\hat{X} \times X \to \mathbb{R}$ (Rockafellar, 1970).*

Remark 5 *If $f(x) = cg(x)$, with $c > 0$ a scalar, then $\hat{f}(\hat{x}) = c\hat{g}(\hat{x}/c)$.*

Theorem 6 *The ERCH equivalent formulations* (5)–(10) *give a solution for the RCH formulation* (11) *when the reduced convex hulls do not intersect, provided that $1/p + 1/q = 1$.*

Proof By Lemma 3, we can now write problem (5) as a single minimization problem of the form

$$\min_{W,\alpha,\beta,\xi} \quad \beta - \alpha + \eta \sum_{i \in M} \xi_i \qquad (12)$$
$$\text{s.t.} \quad \begin{cases} W \cdot X_i \geq \alpha - \xi_i, & i \in M_+, \\ W \cdot X_i \leq \beta + \xi_i, & i \in M_-, \\ \xi_i \geq 0, & i \in M, \\ \|W\|_p \leq 1, \end{cases}$$

whose Lagrangian is

$$\begin{aligned} \mathcal{L} \;=\; & \beta - \alpha + \eta \sum_{i \in M} \xi_i - \sum_{i \in M_+} \lambda_i \left(W \cdot X_i - \alpha + \xi_i\right) \\ & + \sum_{i \in M_-} \lambda_i \left(W \cdot X_i - \beta - \xi_i\right) - \sum_{i \in M} \mu_i \xi_i \\ & + \delta \left(\|W\|_p - 1\right). \end{aligned}$$

However, since the ℓ_p–norm is not necessarily differentiable, we cannot proceed now as in Section 2. To derive the dual problem, we must take into account that it consists of finding the maximum, with respect to the Lagrange multipliers, of the infimum of the Lagrangian, where this infimum is with respect to the primal variables (Boyd and Vandenberghe, 2004). In our case the primal variables are W, α, β and ξ, whereas the Lagrange multipliers are λ_i, μ_i and δ. Thus, the dual of (12) translates to

$$\max_{\lambda\geq 0,\mu\geq 0,\delta\geq 0} \quad \inf_{W,\alpha,\beta,\xi} \quad \{\mathcal{L}\}\,, \tag{13}$$

where $\mathcal{L}$ is the expression above. Splitting the infimum among the different variables, we want to find

$$\begin{gathered}\inf_W \left\{\sum_{i\in M_-} \lambda_i W\cdot X_i - \sum_{i\in M_+} \lambda_i W\cdot X_i + \delta\|W\|_p\right\} + \\ \inf_\alpha \left\{-\alpha + \alpha\sum_{i\in M_+}\lambda_i\right\} + \inf_\beta \left\{\beta - \beta\sum_{i\in M_-}\lambda_i\right\} + \\ \sum_{i\in M}\inf_{\xi_i}\{\eta\xi_i - \lambda_i\xi_i - \mu_i\xi_i\} - \delta.\end{gathered}$$

For α, β and ξ we can find the infima just by differentiating and equating to 0:

$$\begin{aligned}\frac{\partial\mathcal{L}}{\partial\alpha} &= \sum_{i\in M_+}\lambda_i - 1 = 0 \quad\Rightarrow\quad \sum_{i\in M_+}\lambda_i = 1,\\ \frac{\partial\mathcal{L}}{\partial\beta} &= 1 - \sum_{i\in M_-}\lambda_i = 0 \quad\Rightarrow\quad \sum_{i\in M_-}\lambda_i = 1,\\ \frac{\partial\mathcal{L}}{\partial\xi_i} &= \eta - \lambda_i - \mu_i = 0 \quad\Rightarrow\quad 0\leq\lambda_i\leq\eta,\ \ i\in M.\end{aligned}$$

As for W, we can write the infimum as $\inf\left\{-\sum_{i\in M}\lambda_i y_i W\cdot X_i + \delta\|W\|_p\right\}$. This expression follows the form of the convex conjugate as presented in Definition 4, where we can identify the functional $f(W) = \delta\|W\|_p$ and the dual variable $\hat{W} = \sum_{i\in M}\lambda_i y_i X_i$. Moreover, assuming for the moment that $\delta > 0$ and using Remark 5, we have $g(W) = \|W\|_p$.

Since the convex conjugate of the ℓ_p–norm is given by

$$\hat{g}(\hat{W}) \;=\; \begin{cases}0 & \text{if } \|\hat{W}\|_q\leq 1,\\ +\infty & \text{otherwise,}\end{cases}$$

where $1/p + 1/q = 1$ (see Boyd and Vandenberghe, 2004), we get in our case that the term $\inf\left\{-\sum_{i\in M}\lambda_i y_i W\cdot X_i + \delta\|W\|_p\right\}$ equals

$$-\hat{f}\left(\hat{W}\right) = -\delta\hat{g}\left(\frac{\hat{W}}{\delta}\right) \;=\; \begin{cases}0 & \text{if } \left\|\frac{\hat{W}}{\delta}\right\|_q\leq 1,\\ -\infty & \text{otherwise,}\end{cases}$$

which can be rewritten as

$$-\delta\hat{g}\left(\frac{1}{\delta}\sum_{i\in M}\lambda_i y_i X_i\right) \;=\; \begin{cases} 0 & \text{if } \left\|\sum_{i\in M}\lambda_i y_i X_i\right\|_q \leq \delta, \\ -\infty & \text{otherwise.} \end{cases}$$

The optimum will be located in the region where the convex conjugate is finite, so (13) is equivalent to

$$\begin{aligned} \max_{\lambda,\delta} \quad & -\delta \\ \text{s.t.} \quad & \begin{cases} \sum_{i\in M_+}\lambda_i = \sum_{i\in M_-}\lambda_i = 1, \\ 0 \leq \lambda_i \leq \eta, \;\; i \in M, \\ \left\|\sum_{i\in M}\lambda_i y_i X_i\right\|_q \leq \delta. \end{cases} \end{aligned} \tag{14}$$

On the other hand, when $\delta = 0$ the infimum on W is just $\inf\left\{-\sum_{i\in M}\lambda_i y_i W\cdot X_i\right\}$. Differentiating with respect to W we obtain that $X_+ = X_-$, so that $W = 0$. Consequently, $\left\|\sum_{i\in M}\lambda_i y_i X_i\right\|_q = 0 = \delta$, which satisfies (14).

Observe that the non–negativity constraints of the Lagrange multipliers in (13) are subsumed in the constraints above. This can be further rewritten, removing δ, as

$$\begin{aligned} \min_{\lambda} \quad & \left\|\sum_{i\in M}\lambda_i y_i X_i\right\|_q \\ \text{s.t.} \quad & \begin{cases} \sum_{i\in M_+}\lambda_i = \sum_{i\in M_-}\lambda_i = 1 \\ 0 \leq \lambda_i \leq \eta, \;\; i \in M, \end{cases} \end{aligned}$$

that is, problem (11).

■

Therefore, when the hulls do not intersect, ERCH–NPP results in the standard RCH–NPP problem. It is worth noting that Bennett and Bredensteiner (2000) already described how RCH–NPP relates to RCH–Margin (which is a particular case of our ERCH–Margin formulation 5 for non–intersecting hulls), for the ℓ_1, ℓ_2 and ℓ_∞–norms. Nevertheless, their proof was omitted due to space constraints. We cover general p and q, which include all these as particular cases.

Addressing now the non–intersecting case, we introduce another lemma, analogous to Lemma 3.

Lemma 7 *If the reduced convex hulls intersect, we can replace the constraint* $\|W\|_p = 1$ *in* (5) *with* $\|W\|_p \geq 1$.

Proof Just follow a similar argument to the one presented for Lemma 3. By the nature of problem (7), and since there is overlap, we obtain $W^* \cdot X_+^* + b^* < 0$ and $W^* \cdot X_-^* + b^* > 0$ for any optimal W^*, X_+^*, X_-^* (see Figure 4).

Therefore, at the optimum of (7) and (9) the value of the inner maximum must be positive. Supposing that $\|W^*\|_p > 1$ allows us to build an alternative feasible solution $(W^*/\|W^*\|_p, \alpha^*/\|W^*\|_p, \beta^*/\|W^*\|_p, \xi^*/\|W^*\|_p)$, whose norm is unitary and whose primal value is less than that of our hypothetical optimal solution, contradicting thus this optimality. ■

The problem can be then rewritten as

$$\min_{W} \min_{\alpha,\beta,\xi} \quad \beta - \alpha + \eta \sum_{i \in M} \xi_i$$
$$\text{s.t.} \quad \begin{cases} W \cdot X_i \geq \alpha - \xi_i, & i \in M_+, \\ W \cdot X_i \leq \beta + \xi_i, & i \in M_-, \\ \xi_i \geq 0, & i \in M, \\ \|W\|_p \geq 1. \end{cases}$$

In contrast to the derivation in Theorem 6, obtaining the dual of this problem is counterproductive. Since the constraint $\|W\|_p \geq 1$ is non–convex, a non–zero dual gap is bound to appear. Therefore, solving the dual problem would only provide an approximate solution to the ERCH. Instead of following such a derivation, we take the ERCH–NPP formulation in 7 and plug in the modified constraint on W, obtaining

$$\min_{\|W\|_p \geq 1} \quad \max_{X_+ \in \mathcal{U}_+, X_- \in \mathcal{U}_-} \quad W \cdot X_- - W \cdot X_+.$$

The immediate advantage of this formulation of the ERCH is that, whatever the data points X, a trivial solution $W = 0$ is never obtained. In comparison, the RCH–NPP model always produces the trivial solution whenever the reduced hulls intersect. Joining this and the facts above, it is immediate that ERCH–NPP can be regarded as a generalization of RCH–NPP.

Theorem 8 *ERCH–NPP is a generalization of RCH–NPP (connection 10 in Figure 1).*

Proof Given the data points for which to solve ERCH–NPP, the reduced convex hulls formed by such points might or might not intersect. If they do not intersect, by Theorem 6 the solution of the ERCH–NPP problem is exactly the solution of RCH–NPP. If they do intersect, then RCH–NPP fails to find a non–trivial solution, while ERCH–NPP does not, by Lemma 7. Therefore, ERCH–NPP covers all feasible cases for RCH–NPP plus a new set, hence being a generalization of RCH–NPP. ■

Note that ERCH–Margin in (5) is nothing but RCH–Margin in (4), with the additional requirement $\|W\|_p = 1$. Regarding the above two possible cases, we have seen that if the

reduced convex hulls do not intersect we can substitute this constraint with $\|W\|_p \leq 1$, so that we obtain the solution of RCH–NPP and, by strong duality, that of RCH–Margin. When they do intersect, RCH–NPP and RCH–Margin give a trivial 0 solution, whereas ERCH–NPP and ERCH–Margin do not, since we can use the constraint now that $\|W\|_p \geq 1$. Thus, it can be stated as follows:

Corollary 9 *ERCH–Margin is a generalization of RCH–Margin (connection 8 in Figure 1).*

4. Structure of the ERCH–NPP

The actual problem of solving ERCH–NPP

$$\min_{W:\|W\|_p=1} \quad \max_{X_+\in\mathcal{U}_+, X_-\in\mathcal{U}_-} \quad W\cdot X_- - W\cdot X_+,$$

is non–trivial, the main reason being that the constraint $\|W\|_p = 1$ imposes a non–convex feasible set. This might lead to local minima among other issues, which in turn make the optimization process difficult.

As described in the previous section, if the reduced convex hulls for the given data points do not intersect, then the problem above can be reduced to the standard RCH–NPP. Therefore, in such case the optimization can be performed by just employing one of the available solvers for RCH–NPP, such as the RCH–SK and RCH–MDM methods proposed respectively in Mavroforakis and Theodoridis (2006) and López et al. (2011a).

Of course, such methods cannot be applied in the intersecting hulls case, which is actually the one of most interest, since it cannot be addressed by the RCH–NPP model. It is therefore necessary to develop an optimization algorithm suitable for the general ERCH–NPP case; to do so we will first analyze the structure of the optimization problem posed by ERCH–NPP.

It is clear that we can recast the problem to solve as the minimization of a function

$$\min_{\|W\|_p=1} \quad f(W), \tag{15}$$

where

$$\begin{aligned} f(W) &= \max_{X_+\in\mathcal{U}_+, X_-\in\mathcal{U}_-} \{W\cdot X_- - W\cdot X_+\}, \\ &= \max_{X_-\in\mathcal{U}_-} \{W\cdot X_-\} - \min_{X_+\in\mathcal{U}_+} \{W\cdot X_+\}. \end{aligned} \tag{16}$$

This can be further rewritten in the following form

$$f(W) = \max_{X\in\mathcal{M}} W\cdot X, \tag{17}$$

where $\mathcal{M}$ is the Minkowski polygon of the data, which is obtained through the Minkowski difference $\mathcal{M} = \mathcal{U}_- \ominus \mathcal{U}_+$ defined as the set

$$X \ominus Y \equiv \{z | z = x - y, x \in X, y \in Y\}.$$

The Minkowski polygon has been used historically in the context of RCH–NPP to design efficient solvers (Mavroforakis et al., 2007; Keerthi et al., 2000). The properties of the Minkowski difference guarantee that the difference of two convex sets is also a convex set (Ericson, 2005), and so in our problem $\mathcal{M}$ fancies this property. In this paper we will exploit both representations (16) and (17) to take advantage of the structure of the problem.

Interestingly, the maximum and minimum in Equation (16) can be obtained efficiently from the observations in the work of Mavroforakis and Theodoridis (2006) about the extreme points of reduced convex hulls. As they show, any extreme point in a reduced convex hull can be expressed in the form

$$X_E = \sum_{i=1}^{\lfloor 1/\eta \rfloor} \eta X_i + (1 - \lfloor 1/\eta \rfloor \eta)\, X_{\lceil 1/\eta \rceil},$$

that is, the convex combination of $\lceil 1/\eta \rceil$ points, where $\lfloor 1/\eta \rfloor$ of them are given a weight of η and an additional one the remaining weight $1 - \lfloor 1/\eta \rfloor \eta$ (if it is non–zero). Using this property, they note that the extreme points with minimum margin for a given W can be found as

$$\arg\min_{X \in \mathcal{U}} \{W \cdot X\} \;=\; \sum_{i=1}^{\lfloor 1/\eta \rfloor} \eta X_i^{inc} + (1 - \lfloor 1/\eta \rfloor \eta)\, X_{\lceil 1/\eta \rceil}^{inc},$$

where the X_i^{inc} are the original points X_i sorted increasingly by their margin values

$$W \cdot X_1^{inc} \;\leq\; W \cdot X_2^{inc} \;\leq \ldots \leq\; W \cdot X_N^{inc}.$$

These observations can also be applied here to find the value of $f(W)$, as

$$\arg\max_{X_- \in \mathcal{U}_-} \{W \cdot X_-\} \;=\; \sum_{i=1}^{\lfloor 1/\eta \rfloor} \eta X_{i_-}^{dec} + (1 - \lfloor 1/\eta \rfloor \eta)\, X_{\lceil 1/\eta \rceil_-}^{dec}, \tag{18}$$

$$\arg\min_{X_+ \in \mathcal{U}_+} \{W \cdot X_+\} \;=\; \sum_{i=1}^{\lfloor 1/\eta \rfloor} \eta X_{i_+}^{inc} + (1 - \lfloor 1/\eta \rfloor \eta)\, X_{\lceil 1/\eta \rceil_+}^{inc}, \tag{19}$$

where the X_-^{dec} are the points from the negative class sorted by margin decreasingly, and the X_+^{inc} are the points from the positive class sorted by margin increasingly:

$$W \cdot X_{1_-}^{dec} \;\geq\; W \cdot X_{2_-}^{dec} \;\geq \ldots \geq\; W \cdot X_{m_-}^{dec},$$
$$W \cdot X_{1_+}^{inc} \;\leq\; W \cdot X_{2_+}^{inc} \;\leq \ldots \leq\; W \cdot X_{m_+}^{inc}.$$

The computation of $f(W)$, hence, can be easily done by just performing these sortings, which only require $O(m \log(m))$ operations. This ability to find the value of $f(W)$ for a fixed W is the key for computing the gradient of $f(W)$. Supposing $Z_+ = \arg\min_{X_+ \in \mathcal{U}_+} \{W \cdot X_+\}$ and $Z_- = \arg\max_{X_- \in \mathcal{U}_-} \{W \cdot X_-\}$ and that both Z_+ and Z_- are singletons (no other choices of $X_\pm$ attain the minimum/maximum values), the gradient is clearly $\nabla f(W) = \frac{\partial}{\partial W}(W \cdot Z_- - W \cdot Z_+) = Z_- - Z_+$.

It might happen, however, that Z_+ or Z_- (or both) is a set of points instead of a singleton. If that is the case, which takes place in practice quite often, a set of gradients are possible, constituting the subdifferential

$$\begin{aligned}\frac{\partial f}{\partial W} &= \frac{\partial}{\partial W}\left(\max_{X_-\in\mathcal{U}_-}\{W\cdot X_-\} - \min_{X_+\in\mathcal{U}_+}\{W\cdot X_+\}\right),\\ &= \frac{\partial}{\partial W}\max_{X_-\in\mathcal{U}_-}\{W\cdot X_-\} - \frac{\partial}{\partial W}\min_{X_+\in\mathcal{U}_+}\{W\cdot X_+\},\\ &= \frac{\partial}{\partial W}\max_{X_-\in\mathcal{U}_-}\{W\cdot X_-\} + \frac{\partial}{\partial W}\max_{X_+\in\mathcal{U}_+}\{-W\cdot X_+\}.\end{aligned}$$

Invoking the property that the subdifferential of the maximum of a set of convex functions (linear, in this case) at a given point is the convex hull of the subdifferentials of the functions attaining such maximum at that point (Boyd and Vandenberghe, 2007) [3], we obtain that

$$\begin{aligned}\frac{\partial f}{\partial W} &= \operatorname{conv}\left\{X \,\middle|\, X\cdot W = \max_{X_-\in\mathcal{U}_-} W\cdot X_-\right\}\\ &\quad - \operatorname{conv}\left\{X \,\middle|\, X\cdot W = \min_{X_+\in\mathcal{U}_+} W\cdot X_+\right\},\end{aligned} \tag{20}$$

where conv stands for standard convex hull.

A more intuitive way to understand this subdifferential is to note that the orderings X_-^{dec} and X_+^{inc} need not be unique, since it might well happen that, for instance, $W\cdot X_{i_-}^{dec} = W\cdot X_{(i+1)_-}^{dec}$, and so the relative position of these two elements in the ordering is arbitrary. For these multiple orderings the assignment of weights to obtain $Z_- = \arg\max_{X_-\in\mathcal{U}_-}\{W\cdot X_-\}$ can produce a set of different Z_- vectors, thus explaining the non–singleton subdifferential. Note however that not every reordering produces a different subgradient, since as shown in equations (18-19) the $\lfloor 1/\eta\rfloor$ first $X_{i_\pm}$ vectors in the orderings receive all the same weight η, while all the vectors from the $\lceil 1/\eta\rceil + 1$ have no weight in the combination. In particular, swaps in the ordering of two vectors $W\cdot X_{i_\pm} = W\cdot X_{(i+1)_\pm}$ with equal weight in such combination produce no change in the resulting subgradient. Therefore, only equalities involving the $X_{\lceil 1/\eta\rceil_\pm}$ vector can produce different subgradients. These observations will become useful when discussing the stepsize selection of our proposed algorithm (Section 5.4).

With the subdifferential at hand, one could easily design a subgradient projection (SP) method (Bertsekas, 1995) to solve problem (15). For clarity of the explanations to follow, an outline of this method for the minimization of a general function $f(x)$ constrained to some set X is presented as Algorithm 1. As detailed in the pseudocode, the algorithm basically alternates update steps and projection steps. In the former, the current estimate of the solution is updated by following the negative of some subgradient belonging to the subdifferential, while in the latter the updated solution is moved back to the feasible region

3. This property can be inferred from the observations in Clarke (1990, p. 10–11).

Algorithm 1 Subgradient Projection (SP) method for $\min_{x \in X} f(x)$

Initialization: chose $x^0 \in X$, $t = 0$.
while stopping criterion not met **do**
  Compute a subgradient g^t of $f(x^t)$.
  Select an updating stepsize s^t.
  Update step: $z^{t+1} = x^t - s^t g^t$.
  Projection step: $x^{t+1} = P\left[z^{t+1}\right]_X$
  $t \leftarrow t + 1$.
end while
return x^t.

through an Euclidean projection. This method, though fairly simple, is bound to perform poorly, since it uses little information about the problem at hand. Furthermore, due to the non–convex nature of the problem it is not easy to give any guarantees on convergence.

In spite of SP presenting these drawbacks, we show here how building on top of it and introducing adaptations for this particular problem, it is able to find a solution for ERCH–NPP efficiently. We enhance the SP algorithm by modifying its four basic operations: the computation of the updating direction, the updating stepsize selection, the projection operator, and the initialization procedure.

To guide such modifications, we first introduce the following theorem, which forms the base of our algorithm:

Theorem 10 *The optimum of ERCH-NPP when the reduced hulls intersect is located at a non–differentiable point.*

The details of the proof for this theorem are not relevant for the discussion to follow, so it is relegated to the Appendix. Its importance rather stems from the fact that we can guide the optimization procedure to look just for non–differentiable points in the search space, and still be able to reach the optimum.

5. The RapMinos Algorithm

We describe now the distinctive elements of our proposed solver for ERCH-NPP: the RAdially Projected MInimum NOrm Subgradient (RapMinos) algorithm.

5.1 Updating Direction

The first thing to adapt is the direction used for the update. Using the negative of an arbitrary subgradient, as in SP, can result in non–decreasing updating directions (Bertsekas, 1995), which in turn can make hard to provide any guarantees on convergence. Therefore, we introduce a modification that guarantees descent in the objective function in every iteration, and also allows to perform optimality checks easily. To do so, we need to resort to the concept of minimum–norm subgradient (MNS) from the literature of non–smooth optimization (Clarke, 1990):

Definition 11 *Consider a non–smooth function $f(x)$, and its subdifferential set $\partial f(x)$ at a point x. The minimum–norm subgradient $g^*(x)$ is then*

$$g^*(x) = \underset{g \in \partial f(x)}{\arg\min} ||g||,$$

for some proper norm $||\cdot||$.

In an unconstrained problem, the direction given by $d = -g^*(x)$ is guaranteed to be a descent direction. When constraints are introduced, however, such a guarantee is harder to obtain. We nevertheless are able to meet it through the following theorem:

Theorem 12 (Descent directions for ERCH) *Consider the Lagrangian of problem (15)*

$$L(W, \lambda) = f(W) + \lambda(||W||_p - 1),$$

with $\lambda \in \mathbb{R}$ the Lagrange coefficient. Now consider the subdifferential set of the Lagrangian,

$$\Gamma(W) = \partial f(W) + \lambda \partial ||W||_p,$$

and suppose that the current W is feasible, so that $||W||_p = 1$. Then the element with minimum norm in $\Gamma(W)$,

$$\gamma^*(W) = \underset{\gamma \in \Gamma(W)}{\arg\min} ||\gamma||, \tag{21}$$

meets $||\gamma^*(W)|| = 0$ if W is a local minimum of the problem. Else, the direction $d = -\gamma^*(W)$ is guaranteed to be a descent direction.

Once again, the proof of the theorem is relegated to the Appendix to avoid technical clutter in the discussion. The theorem itself provides a powerful tool to obtain both descent directions and a reliable check for optimality, as we will see. But of course, a procedure must be devised to find the appointed MNS of the Lagrangian in (21). A helpful observation for doing so is the fact that the optimal value of the Lagrange coefficient λ can be determined in closed form. Consider (21), and observe that the diversity in the set $\Gamma(W)$ is given by the possible elements of the subdifferentials $\partial f(W)$ and $\partial ||W||_p$, and the Lagrange coefficient λ. To simplify notation, let us define $g \in \partial f(W)$, $n \in \partial ||W||_p$ elements of the subdifferentials. By considering the problem just in terms of λ we can write

$$\begin{aligned} & \min_{\lambda} ||g + \lambda n||_2^2, \\ = \; & \min_{\lambda} ||g||_2^2 + \lambda^2 ||n||_2^2 + 2\lambda g \cdot n. \end{aligned}$$

Note that even if the MNS is defined for any proper norm, we have employed the ℓ_2–norm here to ease the calculations. Computing now the derivative and solving for λ we obtain

$$\begin{aligned} \frac{\partial}{\partial \lambda} &= 2\lambda ||n||_2^2 + 2g \cdot n = 0, \\ \lambda^* &= -\frac{g \cdot n}{n \cdot n} = -P\left[g\right]_n, \end{aligned}$$

which is precisely the negative of the coefficient for the Euclidean projection of g on n, $P\left[g\right]_n$. With this in mind and assuming that the subdifferential $\partial ||W||_p$ is a singleton n [4], problem (21) is simplified down to

$$\begin{aligned} \min_g \quad & ||g - P\left[g\right]_n \cdot n||_2^2, \\ \text{s.t.} \quad & g \in \partial f(W), \end{aligned} \tag{22}$$

and the resulting updating direction d would be $d = -(g^* - P\left[g^*\right]_n \cdot n)$ with g^* the minimizer of the problem. Before discussing how this minimizer is found, first we show how the computation of the vector $n = \partial ||W||_p$ is performed.

Since we have assumed that $||W||_p = 1$, we can safely temporarily replace the constraint by $||W||_p^p = 1$, which eases the calculations. The derivative is then

$$\frac{\partial ||W||_p^p}{\partial W} = \frac{\partial}{\partial W} \sum_i |W_i|^p .$$

It is easier to develop this derivative by considering each entry of the gradient vector separately,

$$\begin{aligned} \left[\frac{\partial ||W||_p^p}{\partial W}\right]_k &= \frac{\partial}{\partial W_k} \sum_i |W_i|^p , \\ &= p\,|W_k|^{p-1} \frac{\partial}{\partial W_k} |W_k|, \\ &= p\,|W_k|^{p-1} \operatorname{sign}(W_k), \end{aligned} \tag{23}$$

where the subgradient $\frac{\partial}{\partial W}|W_i|$ is the sign function

$$\operatorname{sign}(x) = \begin{cases} 1 & \text{if} \quad x > 0, \\ -1 & \text{if} \quad x < 0, \\ 0 & \text{if} \quad x = 0. \end{cases}$$

A few technicalities have been omitted in this derivation: we refer to the Appendix for the details.

Now that we have a way to compute n, we show how to find the minimizer g^* of problem (22). This is easy to do upon realizing that it can be rewritten as a modified standard RCH-NPP. To do so, first observe that

4. This is not met for the particular cases of norms $p = 1$ and $p = \infty$, as they present non–differentiable points. However, taking this assumption produces no harm in practice. Refer to the Appendix for further discussion on this issue.

$$\begin{aligned} g - P\,[g]_n \cdot n &= g - \frac{g \cdot n}{n \cdot n} n, \\ &= g - \frac{nn^T}{n \cdot n} g, \\ &= \left(\mathcal{I} - \frac{nn^T}{n \cdot n}\right) g, \\ &= \mathcal{N} g, \end{aligned}$$

where $\mathcal{I}$ is the identity matrix and $\mathcal{N} = \mathcal{I} - \frac{nn^T}{n \cdot n}$ transformation matrix. The problem then becomes

$$\begin{aligned} \min_{g} \quad & ||\mathcal{N} g||_2^2 \,, \\ \text{s.t.} \quad & g \in \partial f(W). \end{aligned} \tag{24}$$

To realize the underlying connections with RCH–NPP, we shall rewrite explicitly the constraint $g \in \partial f(W)$. To do so, remember that g can be expressed as the difference of two extreme points (see Eq. 20) and these in turn as a convex combination of the data in each class (Eqs. 18 and 19). Therefore we have that $g = \sum_{i \in M_-} \mu_i X_i - \sum_{i \in M_+} \mu_i X_i$ for some combination weights μ_i, which should be set according to the margin orderings (as explained in Eqs. 18-19). To be more precise, let us define the index sets

$$\begin{aligned} \mathcal{S}_+ &= \left\{ i \;\middle|\; i \in M_+ \,,\; W \cdot X_i = W \cdot X^{inc}_{\lceil 1/\eta \rceil_+} \right\}, \\ \mathcal{S}_- &= \left\{ i \;\middle|\; i \in M_- \,,\; W \cdot X_i = W \cdot X^{dec}_{\lceil 1/\eta \rceil_-} \right\}, \\ \mathcal{Q}_+ &= \left\{ i \;\middle|\; i \in M_+ \,,\; W \cdot X_i < W \cdot X^{inc}_{\lceil 1/\eta \rceil_+} \right\}, \\ \mathcal{Q}_- &= \left\{ i \;\middle|\; i \in M_- \,,\; W \cdot X_i > W \cdot X^{dec}_{\lceil 1/\eta \rceil_-} \right\}. \end{aligned}$$

These sets can be explained as follows. At a differentiable point W the orderings X^{inc}_+ and X^{dec}_- are unique, and so $\mathcal{S}_\pm$ is a singleton containing just the index corresponding to $X^{inc/dec}_{\lceil 1/\eta \rceil_\pm}$, which is the only pattern with weight $(1 - \lfloor 1/\eta \rfloor\, \eta)$, while the sets $\mathcal{Q}_\pm$ contain the indices of all patterns with weight η. At a non–differentiable point, however, the sets $\mathcal{S}_\pm$ contain the indices of those patterns that can be swapped in the ordering while keeping the same objective value in (7), since they have equal margin. While the patterns indexed by $\mathcal{Q}_\pm$ still maintain a fixed weight η, the weights of the patterns indexed by $\mathcal{S}_\pm$ can be rearranged to obtain different subgradients. We are able to represent implicitly the whole subdifferential with $\mathcal{S}_\pm$ and $\mathcal{Q}_\pm$. Indeed, we can define the constant $C = \sum_{i \in \mathcal{Q}_-} \eta X_i - \sum_{i \in \mathcal{Q}_+} \eta X_i$, which only contains fixed terms, and rewrite our direction problem (24) as

$$\begin{aligned} \arg\min_{\mu} \quad & \left\| \mathcal{N}\left(C + \sum_{i\in\mathcal{S}_-} \mu_i X_i - \sum_{i\in\mathcal{S}_+} \mu_i X_i \right) \right\|_2^2, \\ \text{s.t.} \quad & \begin{cases} \sum_{i\in S_-} \mu_i + \sum_{i\in Q_-} \eta = 1, \\ \sum_{i\in S_+} \mu_i + \sum_{i\in Q_+} \eta = 1, \\ 0 \le \mu_i \le \eta, \quad \forall\, i \in S_\pm. \end{cases} \end{aligned}$$

Note that the constraints are nothing but the RCH–NPP constraints (problem 3), though taking into account that the points in the $Q_\pm$ sets have fixed weight η. Using this fact and defining $\tilde{X} = \mathcal{N}X$, $\tilde{C} = \mathcal{N}C$ we get the simplified problem

$$\begin{aligned} \arg\min_{\mu_i, i\in S_\pm} \quad & \left\| \tilde{C} + \sum_{i\in\mathcal{S}_-} \mu_i \tilde{X}_i - \sum_{i\in\mathcal{S}_+} \mu_i \tilde{X}_i \right\|_2^2, \\ \text{s.t.} \quad & \begin{cases} \sum_{i\in S_-} \mu_i + |Q_-|\eta = 1, \\ \sum_{i\in S_+} \mu_i + |Q_+|\eta = 1, \\ 0 \le \mu_i \le \eta, \quad \forall\, i \in S_\pm, \end{cases} \end{aligned} \tag{25}$$

where only the μ weights of the non–fixed points in $\mathcal{S}_\pm$ need to be optimized over. This problem is solved trivially by introducing some small modifications into an RCH–NPP solver; more details on this are given in the implementation section (6.1). The relevant fact here is that we can obtain a descent direction in our ERCH algorithm by solving problem (25), and this can be done efficiently by invoking an RCH solver.

5.2 Geometric Intuition of Updating Direction

Even though involved arguments from non–smooth optimization have been used to obtain the updating direction, it turns out that an easy geometric intuition can be given for it. But before introducing it, some definitions from geometry are needed:

Definition 13 *Supporting hyperplane: given a set $X \in \mathbb{R}^n$, a hyperplane h_X supports X if X is entirely contained in one of the two closed half–spaces determined by $h_X(x)$ and h_X contains at least one point from X.*

Definition 14 *Supporting hyperplane at a point: given a closed set $X \in \mathbb{R}^n$ and a point x in the boundary of X, a hyperplane $h_X(x)$ supports X at x if it is supports X and contains x. If the set X is convex, $h_X(x)$ is guaranteed to exist (Boyd and Vandenberghe, 2004).*

We further introduce the definition of supporting projection as

Definition 15 *Supporting projection: given a closed convex set $X \in \mathbb{R}^n$, a point $x \in X$ at a boundary of X and a vector v originating at x, we define the supporting projection of v on*

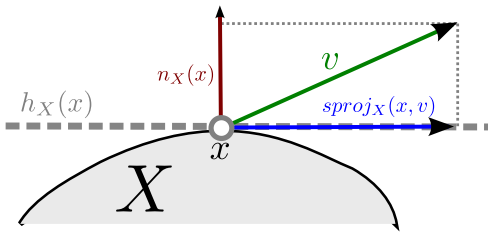


Figure 5: Depiction of the geometric concepts of supporting hyperplane at a point and of supporting projection. The hyperplane $h_X(x)$ supports the set X at the point x. The supporting projection of v is then obtained by projecting v onto $h_X(x)$, which is equivalent to removing from v its projection on the normal vector $n_X(x)$.

X, $\text{sproj}_X(x, v)$ *as the Euclidean projection of* v *on the supporting hyperplane at* x, $h_X(x)$. *That is to say*

$$\begin{aligned} \underset{X}{\text{sproj}}(x, v) &= P\left[v\right]_{h_X(x)} = v - P\left[v\right]_{n_X(x)} n_X(x), \\ &= v - \frac{v \cdot n_X(x)}{n_X(x) \cdot n_X(x)} n_X(x), \end{aligned} \tag{26}$$

for $n_X(x)$ *the normal vector defining* $h_X(x)$. *This is equivalent to removing from* v *its projection on the normal vector* $n_X(x)$.

An illustrating example on these concepts is given in Figure 5.

Using these, we can see that our updating direction takes the form

$$d = -(g^* - P\left[g^*\right]_n n) = -\underset{||W||_p=1}{\text{sproj}}\ (W, g^*), \tag{27}$$

since the normal vector $n_{||W||_p=1}(W)$ is nothing but the derivative $\partial||W||_p = n$. That is to say, our proposed direction follows the negative of the supporting projection of g^*, with g^* the subgradient that produces the smallest such projection.

5.3 Projection Operator

Now that the updating direction is well defined, we move on to defining a suitable projection operator, which is required to meet our assumption above about W being feasible at every iteration ($||W||_p = 1$). Instead of using Euclidean projection, as is the rule in SP, we instead employ radial projection on the ℓ_p unit–ball (Figueiredo and Karlovitz, 1967), which is defined as

$$R_p\left[x\right] = \begin{cases} x & \text{if} \quad ||x||_p \leq 1, \\ x/||x||_p & \text{if} \quad ||x||_p > 1. \end{cases} \tag{28}$$

One major advantage of using this operator instead of Euclidean projections is its simplicity and generality for any norm p. Furthermore we have the following property:

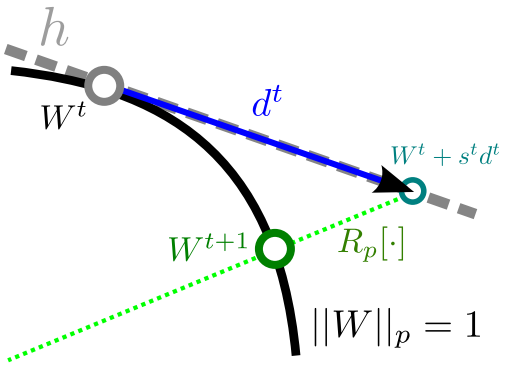


Figure 6: Example of an updating step within the RapMinos algorithm. The point W^t is updated by a displacement along the supporting hyperplane $h = h_{||W||_p=1}(W^t)$ following direction d^t, and then mapped back to the feasible region by means of a radial projection.

Lemma 16 *The radial projection $R_p[x]$ never increases the ℓ_p norm of x, i.e., $||R_p[x]||_p \leq ||x||_p$.*

Proof This is immediate from the definition, since for $||x||_p \leq 1$ the projection leaves x unchanged, and for $||x||_p \geq 1$, $R_p[x] = x/||x||_p$, and so $||R_p[x]||_p = ||x/||x||_p||_p = ||x||_p/||x||_p = 1 \leq ||x||_p$. ■

It must be noted that applying this projection operator to the ERCH–NPP problem could, in principle, lead to infeasible W values, since for $||W||_p < 1$ the projection leaves W unchanged, i.e., $R_p[W] = W$. This violates the constraint $||W||_p = 1$, producing an infeasible W at the end of the iteration. Fortunately, it is easy to show that this situation cannot happen during our algorithm.

Lemma 17 *For a given W^t vector with $||W^t||_p = 1$ and any stepsize $s^t \in \mathbb{R}$, the update $W^{t+1} = R_p\left[W^t + s^t d^t\right]$ with d^t as defined in (27) meets $||W^{t+1}||_p = 1$.*

Proof The proof follows from the fact that the displaced point $W^t + s^t d^t$ is guaranteed to lie in the supporting hyperplane $h_{||W||_p=1}(W^t)$, given the nature of the updating direction d^t and the fact that the $||W^t||_p = 1$, i.e., W^t lies in the border of the convex set $||W^t||_p \leq 1$ (see Figure 6). Because of the properties of a supporting hyperplane, every point in h is guaranteed to be outside or in the border of the set $||W^t||_p \leq 1$, and so $||W^t + s^t d^t||_p \geq 1$. Therefore, using the definition of radial projection, $||W^{t+1}||_p = ||R_p\left[W^t + s^t d^t\right]||_p = 1$. ■

Thus, we are guaranteed to remain in the feasible set throughout the whole algorithm as long as $||W^0||_p = 1$, which is easy to meet.

5.4 Stepsize Selection

Standard subgradient projection methods generally employ a constant or diminishing stepsize rule. Here, however, we can take advantage of Theorem 10 to select a more informed stepsize. Since the optimum of the ERCH is guaranteed to lie at a non–differentiable point, once an updating direction has been selected it makes sense to consider just those stepsizes that land on one of such points.

Recall from the beginning of the section that a non–differentiable point (that is, one where a non–singleton subdifferential arises) can be characterized through the orderings $W \cdot X_{i_-}^{dec}$ and $W \cdot X_{i_+}^{inc}$ as those values of W for which these orderings are not unique, i.e., some elements might be swapped without violating the ordering. In particular, only situations where equalities with the vectors $W \cdot X_{\lceil 1/\eta \rceil_\pm}$ arise can produce non–singleton subdifferentials. Therefore, we can identify non–differentiable points along the updating direction as those values of the stepsize s^t for which $W^t + s^t d^t$ produces one of such equalities, that is to say

$$(W^t + s^t d^t) \cdot X_{\lceil 1/\eta \rceil_\pm} = (W^t + s^t d^t) \cdot X_{i_\pm},$$

for some other $X_{i_\pm}$ vector in the ordering. Since several of such points can appear along the direction d^t, our approach here is to move on to the nearest of them. That is, we select the minimum stepsize (different from 0) that lands on a non–differentiable point. This approach is sensible because by moving further away we could step into a different smooth region where our current estimate of the subgradient (and thus d) is no longer valid. This results in the stepsize rule

$$s^t = \min_{i_\pm \in C_+ \bigcup C_-} \left\{ \frac{X_{\lceil 1/\eta \rceil_\pm} \cdot W^t - X_{i_\pm} \cdot W^t}{X_{i_\pm} \cdot d^t - X_{\lceil 1/\eta \rceil_\pm} \cdot d^t} \right\}, \tag{29}$$

which is obtained from solving the equality above for s^t, and taking the minimum over all of the possible equalities. The sets C_+, C_- arise from the fact that not all data points need to be checked. These sets are defined as

$$C_+ = \left\{ i \in M_+ : \begin{array}{l} X_i \cdot d^t > X_{\lceil 1/\eta \rceil_+} \cdot d^t, i < \lceil 1/\eta \rceil_+, \\ X_i \cdot d^t < X_{\lceil 1/\eta \rceil_+} \cdot d^t, i > \lceil 1/\eta \rceil_+. \end{array} \right\},$$
$$C_- = \left\{ i \in M_- : \begin{array}{l} X_i \cdot d^t < X_{\lceil 1/\eta \rceil_-} \cdot d^t, i < \lceil 1/\eta \rceil_-, \\ X_i \cdot d^t > X_{\lceil 1/\eta \rceil_-} \cdot d^t, i > \lceil 1/\eta \rceil_-. \end{array} \right\}.$$

The choice of these sets becomes clear by realizing that any point not in this set produces a negative or undefined s^t value, which is useless in our method since we are interested in advancing by following the updating direction.

We state now the following proposition, whose proof is immediate by construction of the stepsize, as presented above:

Proposition 18 RapMinos *explores a non–differentiable point at each iteration.*

Algorithm 2 RapMinos method for ERCH-NPP

Inputs: data (X, y), norm $p \in [1, \infty]$, stopping tolerance ϵ.
Initialization: chose $W^0 = W_{\eta_{min}}$, $t = 0$, $stop = \infty$.
while $stop > \epsilon$ **do**
  Find Lagrangian MNS γ_t^* solving problem (25).
  Find stepsize s^t using (29).
  Update step: $V^{t+1} = W^t - s^t \gamma_t^*$.
  Radial projection step: $W^{t+1} = R_p[V^{t+1}]$ (Eq. 28).
  Stopping criterion: $stop = ||\gamma_t^*||_\infty$.
  $t \leftarrow t + 1$.
end while
return W^t.

5.5 Initialization

While any feasible W s.t. $||W||_p = 1$ is a valid starting point, the choice of such point will determine the local minima the algorithm ends up in. As we discuss later in the experimental section, falling in a bad local minimum can result in poor classification accuracy. Therefore, it is relevant to start the optimization at a sensible W point. To do so, we propose the following heuristic. Let us consider the minimum possible value for η, which is $\eta_{min} = 1/\min\{M_+, M_-\}$. At this value each class hull gets reduced to a unique point, its barycenter, where every pattern is assigned the same weight in the convex combination. For such η, the ERCH–NPP is trivially solved by computing W as the difference between both barycenters, $W_{\eta_{min}}$. While such W will not be the solution for other values of η, intuitively we see that it will be already positioned in the general direction of the desired W_η. Although we cannot give any theoretical guarantees on such choice being a good starting point, we will see in the experimental section 6.5 how it performs well in practice.

5.6 Full Algorithm and Convergence Analysis

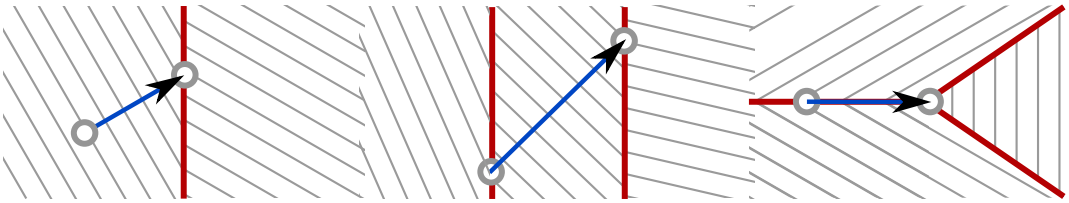
After joining the improvements presented in the previous subsections, the main steps of the full RapMinos method are presented in Algorithm 2. We show now how the iteration of such steps guarantees convergence to a local minimum of the problem. The main argument of the proof is that the RapMinos algorithm visits a region of the function at each step, but always improving the value of the objective function. Since the number of such regions is finite, the algorithm must stop at some point, having found a local minimum. The details of the proof are presented in what follows.

First we will require the following lemma:

Lemma 19 *Consider the update $W^{t+1} = R_p[W^t + s^t d^t]$ with d^t defined as in (27) and s^t defined as in (29). This update never worsens the value of the objective function, i.e., $f(W^{t+1}) \leq f(W^t)$. Furthermore, if $W^{t+1} = W^t$ then W^t is a local minimum, else $f(W^{t+1}) < f(W^t)$.*

Proof

Theorem 12 already shows that at a local minimum the update direction selected by RapMinos is null. If not at a local minimum, the updating direction is guaranteed to be



(a) (b) (c)

Figure 7: Depiction of possible scenarios arising during a RapMinos update. (a) Start in a smooth region, stop at a non–differentiable intersection between smooth regions. (b) Start at an intersection between regions, traverse a smooth region until another intersection is found. (c) Start at an intersection between regions, move along a boundary until intersection with a new smooth region is found.

a descent direction. Therefore $f(W^t + \delta d^t) \leq f(W^t)$ for some small $\delta > 0$. Consider now the structure of the objective and subgradient functions, as shown in Eqs. 16 and 20. Note that $f(W)$ is piece–wise linear, the subgradient set being a unique gradient in the interior of the linear regions, while being non–singleton in the intersections of such regions. With this in mind, the following three cases regarding the status of W^t are possible, which are also depicted in Figure 7:

- W^t is a differentiable point. Then W^t lies in a linear region, where the subgradient set is a unique constant gradient. Because of this, the Minimum Norm Subgradient of the Lagrangian is also constant throughout the whole region, and d^t remains a descent direction until a non–differentiable point marking the frontier to another region is reached (Figure 7a).

- W^t is a non–differentiable point, which means W^t is in the intersection of two or more linear regions, and $W^t + \delta d^t$ for some infinitesimal $\delta > 0$ steps in the interior of one linear region. Since the gradient in this region is included in the subgradient of W^t (see Eq. 20) and $f(W^t + \delta d^s) < f(W^t)$ is guaranteed, then moving further along this region must keep the same rate of improvement (since the region is linear), until a non–differentiable point marking the frontier to another region is reached (Figure 7b).

- W^t is a non–differentiable point and $W^t + \delta d^t$ follows an intersection of regions (e.g., follows an edge of the problem's surface). Then the MNS of the Lagrangian is not changed and d^t remains a descent direction until an intersection with a new linear region is found. This case is observed when selecting the stepsize in Eq. 29 (Figure 7c).

Whatever the case, improvement in the objective is guaranteed until the next non-differentiable point is reached. Therefore $f(W^t + s^t d^t) < f(W^t)$.

Including now the radial projection, we have that

$$\begin{aligned} f(W^{t+1}) &= f(R_p[W^t + s^t d^t]) = f\left(\frac{W^t + s^t d^t}{||W^t + s^t d^t||_p}\right), \\ &= \frac{f(W^t + s^t d^t)}{||W^t + s^t d^t||_p} < f(W^t + s^t d^t), \\ &< f(W^t), \end{aligned}$$

since $||W^t + s^t d^t||_p > 1$ (see proof for Lemma 17) and $f(cW) = cf(W)$ for c constant. ■

With this tool we are ready to prove convergence of RapMinos :

Theorem 20 *The* RapMinos *algorithm finds a local minimum in a finite number of steps.*

Proof By Proposition 18, RapMinos explores a vertex or edge of $f(W)$ at each iteration. As $f(W)$ is piece–wise linear, the number of such regions is finite, so at some point the method could step again into a previously visited point. However, this is not possible, since because of Lemma 19, each iteration must either stop at a local minimum or strictly improve the objective value, thus avoiding to return to a previous point. Therefore, RapMinos converges to a local minimum in a finite number of steps. ■

6. Experimental Results

We present now experimental results supporting our proposed ERCH model and the corresponding RapMinos algorithm, as well as details on implementation.

6.1 Implementation

The RapMinos algorithm was implemented in Matlab, and is publicly available for download [5]. The code includes an adapted RCH–NPP algorithm (Clipped–MDM, see López et al., 2011a, 2008) to solve the MNS problem (Eq. 25). The adaptation involves modifying the algorithm to accept the sets of points $\mathcal{Q}_\pm$, which must always retain a coefficient $\mu_i = \eta$ and thus are not optimized over, but nevertheless should be taken into account when computing the objective value. This can be done easily by adapting the initialization and extreme points computation at the end of the algorithm: for further details please refer to the code itself.

A point of technical difficulty in the implementation is the bookkeeping of the index sets $\mathcal{Q}_\pm$, $\mathcal{S}_\pm$. While these could be recomputed from scratch each time they are needed, it is far more efficient to update them throughout the iterations. To do so, at the initialization of RapMinos these sets are built using the initial vector W^0. After that, during the algorithm iterations, these sets are updated at two situations:

- When computing the stepsize using (29), the pattern (or patterns) that produce the min are added to their respective $\mathcal{S}_\pm$ set. This is done because, by definition of the

5. Project web page: `https://bitbucket.org/albarji/rapminos` . Source code and packages available.

stepsize rule, the margin of this pattern after the update equals that of $W \cdot X_{\lceil 1/\eta \rceil}$, and this is what defines the $\mathcal{S}_{\pm}$. This pattern is also removed from the set $\mathcal{Q}_{\pm}$ in the case it was part of it.

- After each MNS computation the values of the weights μ_i for the patterns in the sets $\mathcal{S}_{\pm}$ are checked. If any of them turns out to be 0, it is removed from $\mathcal{S}_{\pm}$, since such pattern has no longer an influence in the subgradient. If it happens to be valued η, then the pattern is transferred to the corresponding $\mathcal{Q}_{\pm}$ set.

Because of numerical errors amounting during the algorithm iterations, such checks are always done with a certain tolerance value. Also, for the same reason, it could happen that an update of the algorithm worsens the value of the objective function, even if this is theoretically impossible thanks to Lemma 19. To address this, our implementation stops whenever a worsening is detected.

Regarding the quality of the solution obtained, it should be noted that the RapMinos algorithm solves the intersecting ERCH–NPP case, and most of its assumptions are based on this fact. To avoid convergence problems if the problem is actually non–intersecting, our implementation first invokes a standard RCH–NPP solver. If the solution W obtained has norm close to zero, the problem might be intersecting. To check whether there is a real intersection we solve the following linear program

$$\begin{aligned} \min_{\lambda,\eta} \quad & \eta, \\ \text{s.t.} \quad & \begin{cases} \sum_{i\in M_+} \lambda_i X_i = \sum_{i\in M_-} \lambda_i X_i, \\ \sum_{i\in M_+} \lambda_i = 1, \sum_{i\in M_-} \lambda_i = 1, \\ 0 \le \lambda_i \le \eta, \ \forall\, i. \end{cases} \end{aligned} \tag{30}$$

which finds the minimum value of η for which the reduced convex hulls intersect. If the user–selected value of η is larger than the one found here, then the hulls intersect, and we continue with the execution of RapMinos . Otherwise, a solution is obtained by solving the equivalent ℓ_p RCH–NPP (Eq. 11) through a generalized RCH–NPP solver; details on this solver are outlined in the Appendix.

6.2 Augmented Model Capacity: Synthetic Data Sets

We first show how the augmented ν range extension of the Eν–SVM model, and thus ERCH–NPP, can improve the classification accuracy of the SVM. As shown in Section 3, the ERCH–NPP model is able to generate non–trivial solutions for those cases where the reduced hulls of the data intersect, on top of all the solutions attainable by the standard RCH–NPP model for non–intersecting hulls. We hypothesize that this capability ought to be specially useful in classification problems where the convex hulls of positive and negative classes have a significant intersecting area, as RCH–NPP would only be able to find useful solutions for a small range of η values. A similar hypothesis was previously proven for other margin based methods when replacing the regularization constraint $\|W\|_p \le 1$ by $\|W\|_p = 1$ or replacing the reduced convex hulls $\mathcal{U}_{\pm}$ by different class shapes (for example, ellipsoids), as shown in Takeda et al. (2013).

To test this, we generated a series of artificial data sets with increasingly larger intersecting areas. We defined conditional probabilities for label +1 and label -1, denoted by $p(X|+1)$ and $p(X|-1)$, as multivariate normal distributions. The mean vector and the variance–covariance matrix of $p(X|+1)$ were defined by the null vector $(0,\ldots,0)^\top \in \mathbb{R}^n$ and the identity matrix $\mathbf{I}_n \in \mathbb{R}^{n\times n}$, respectively (i.e., standard normal distribution). For the other conditional probability, $p(X|-1)$, we randomly generated the variance-covariance matrix having eigenvalues $0.1^2,\ldots,1.5^2$, wherein the square roots of the eigenvalues were numbers placed at even intervals from 0.1 to 1.5. The mean vector of $p(X|-1)$ was defined by $\frac{r}{\sqrt{n}}(1,\ldots,1)^\top \in \mathbb{R}^n$, with r a distance parameter between classes. The larger the r, the smaller the intersecting area between classes. The training sample size and test sample size were set to $m = 2\times 10^3$ and $\widetilde{m} = 10^4$, respectively, while the number of features was chosen as $n = 10$.

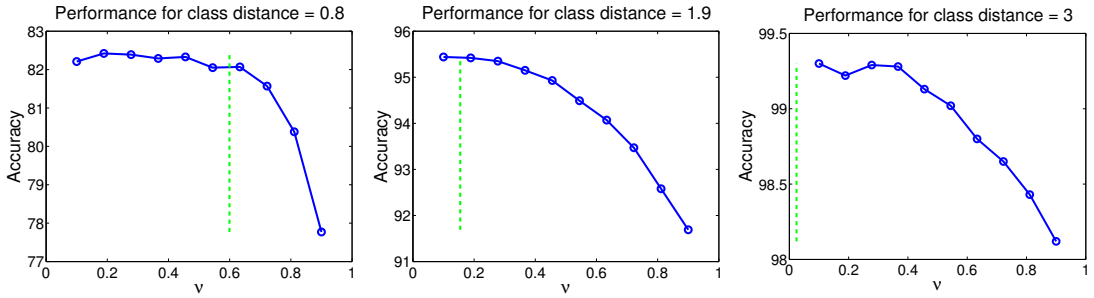


Figure 8: Performance of the Eν–SVM model for a classification problem with different degrees of distance between class centers. For each distance choice, accuracy of the trained classifier is shown for a range of ν values. The green dashed lines represent the ν threshold below which the reduced hulls intersect, hence producing a non–convex problem.

Figure 8 shows the obtained accuracy levels with RapMinos for the range $\nu \in [0.1, 0.9]$ and a selection of class distances. The threshold for which the reduced–convex–hulls intersect is also shown, below which the problem becomes non–convex and only the ERCH–NPP model can find meaningful solutions. As expected, when the distance between class means is large, this threshold becomes smaller, as a smaller ν implies a larger η, i.e., a smaller reduction on the convex hulls is required for them to become separable. For those cases where the distance between classes is small, the intersecting range of ν shows an improvement on accuracy over the non–intersecting range, thus backing up the fact that the augmented range of ERCH–NPP (and so Eν–SVM) can lead to more accurate models.

6.3 Augmented Model Capacity: Real–World Data Sets

We now test the benefits provided by the augmented model capacity on real–world data sets, obtained from the benchmark repository at Rätsch (2000), but instead of making use of the default 100 training–test partitions provided there we generated our own random splits of each data set as done in Takeda and Sugiyama (2009). In particular, we took 4/5 of the data set as training data and the remaining 1/5 as testing data. For each data set

we identified the ν_{limit} value for which the class hulls start intersecting, and solved ERCH–NPP for two ranges of ν values of 100 points each, one above ν_{limit} (convex range), and the other below it (non–convex range). To solve the ERCH–NPP in the non–convex range we resorted to the presented RapMinos method, while for the convex range we applied the standard ν–SVM solver provided in LIBSVM (Chang and Lin, 2001).

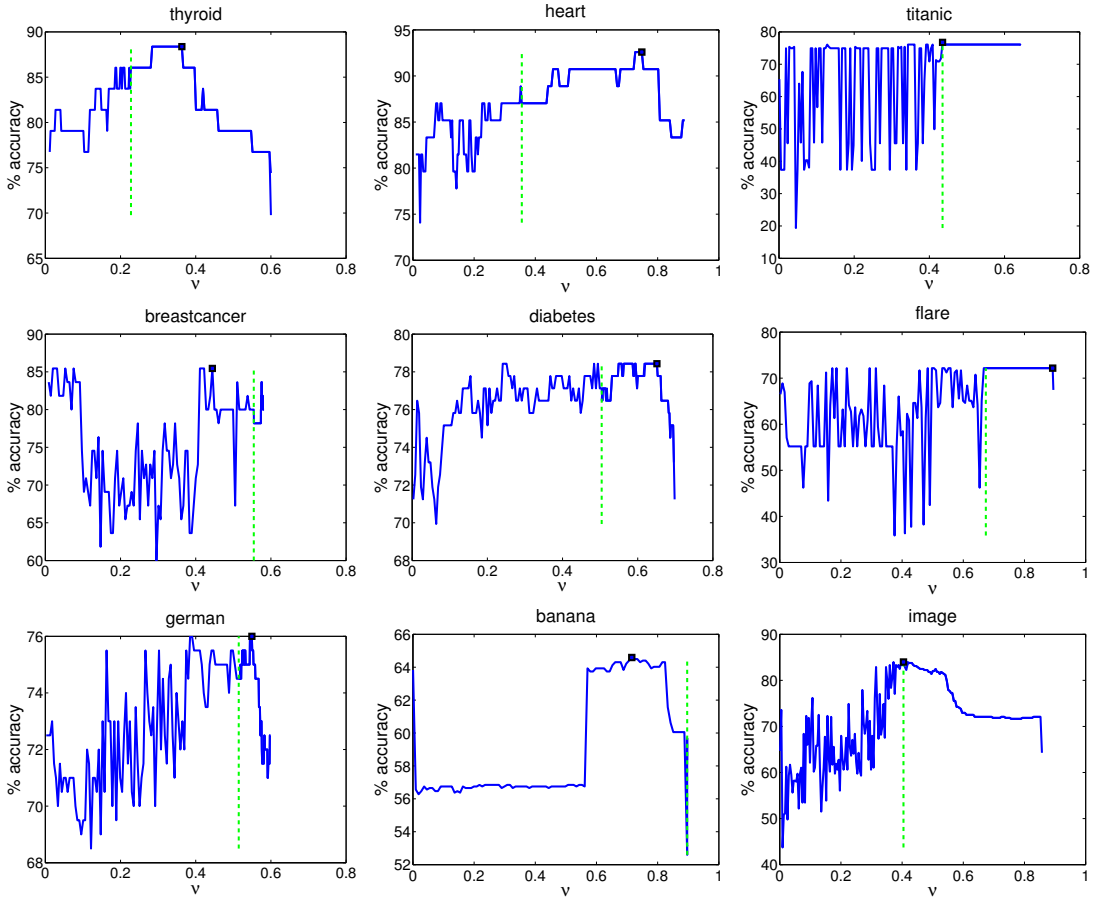


Figure 9: Performance of the Eν–SVM model for a set of real–world data sets. The square markers denote the best performing ν choice.

Figure 9 shows the accuracy levels obtained with RapMinos for the full range of ν values. While for a number of the data sets the augmented ν range does not provide noticeable benefits, for *titanic*, *breastcancer*, *ringnorm* and specially *banana* higher levels of accuracy are attainable.

Table 1 presents top accuracy values in the whole ν range for the standard ν–SVM and the augmented Eν–SVM model tuned with different ℓ_p–norm choices. The results seem to confirm our hypothesis stating that the ability to select an arbitrary ℓ_p regularization in the model leads to an increase in the model capacity: in 8 out of 13 data sets we find that the model is able to obtain higher accuracy values than both the ν–SVM and ℓ_2 Eν–SVM

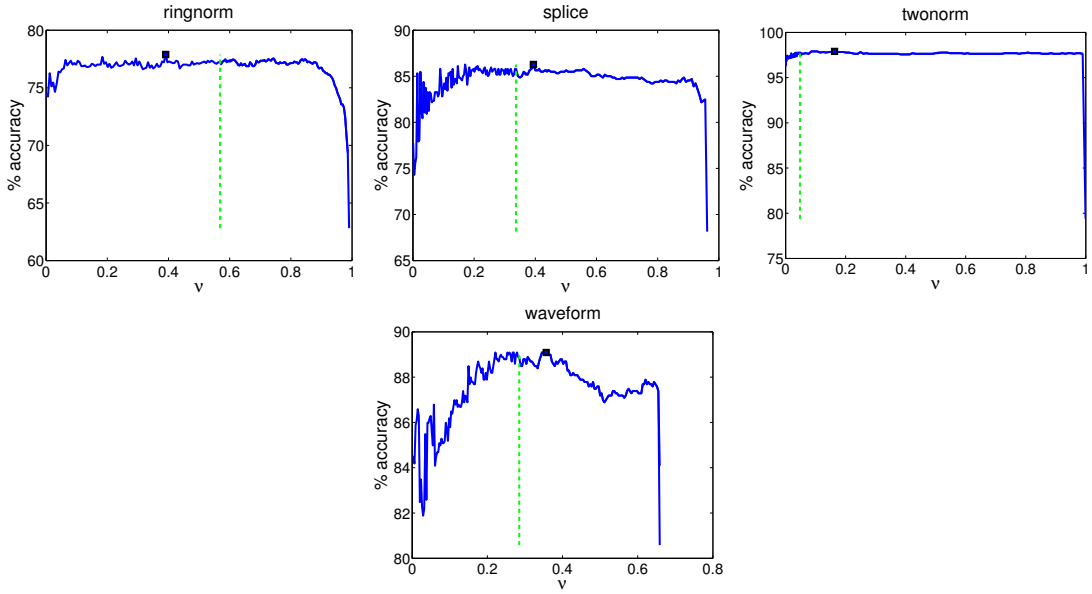


Figure 9: (continued) Performance of the Eν–SVM model for a set of real–world data sets. The square markers denote the best performing ν choice.

models. For illustration purposes we also include the accuracy curves for a sample of the data sets in Figure 10.

6.4 Runtime Experiments

To show the advantage in terms of efficiency and stability of the proposed RapMinos algorithm we present here a comparison against a reference Eν–SVM method. Recall the Eν–SVM problem is dual to the ERCH–NPP discussed here (see Proposition 2), so in principle similar solutions should be obtained through both approaches, although it should be noted that the existence of local minima in both models can lead to different results. The method of choice for the Eν–SVM problem is the one presented in Takeda and Sugiyama (2008) [6], which finds a solution by approximating the non–linear Eν–SVM problem by a series of linear optimization problems; such linear problems, in turn, are solved by invoking an interior–point method.

We worked again with the data sets from the benchmark repository at Rätsch (2000), but since we wanted to test the algorithms in the intersecting range of data, instead of selecting ν as the value maximizing validation accuracy we fixed it at a value slightly below the separable limit ν_{min}. Table 2 shows training times for the reference Eν–SVM and the RapMinos algorithms, together with the accuracy levels obtained in the test splits. A basic subgradient projection method solving ERCH–NPP (see Algorithm 1) is also included in the table to check whether the theoretical improvements provided by RapMinos have noticeable effects in practice.

6. This method turns out to be a subtle modification of the original Eν–SVM method by Pérez-Cruz et al. (2003).

Data set	ν–SVM	ERCH–RapMinos				
	ℓ_2	ℓ_2	ℓ_1	$\ell_{1.5}$	ℓ_3	ℓ_∞
thyroid	88.4%	88.4%	86.0%	88.4%	**95.3%**	90.7%
heart	92.6%	92.6%	90.7%	**94.4%**	92.6%	92.6%
titanic	76.1%	**76.8%**	76.1%	76.8%	76.8%	**77.2%**
breastcancer	83.6%	**85.5%**	81.8%	83.6%	83.6%	83.6%
diabetes	78.4%	78.4%	78.4%	**79.1%**	78.4%	78.4%
flare	72.2%	72.2%	72.2%	72.2%	72.2%	70.3%
german	76.0%	76.0%	76.5%	76.0%	76.0%	**77.0%**
banana	53.2%	**64.6%**	64.2%	64.5%	61.1%	64.6%
image	84.0%	84.0%	71.2%	81.8%	79.4%	78.4%
ringnorm	77.6%	**77.9%**	77.8%	77.7%	**78.0%**	**78.0%**
splice	86.3%	86.3%	86.0%	86.3%	85.8%	85.8%
twonorm	97.9%	97.9%	97.9%	97.9%	97.8%	**98.0%**
waveform	89.1%	89.1%	89.2%	**89.3%**	**89.3%**	89.1%

Table 1: Test accuracies for ν–SVM and the ERCH model trained with RapMinos , for different values of the ℓ_p–norm. Numbers in bold in the RapMinos ℓ_2 mark when the ERCH model performs better than the standard ν–SVM. Also marked in bold are those cases where a non–standard ℓ_p norm produces further improvement.

The first thing to observe is that the Eν–SVM algorithm used failed to produce a solution for some of the data sets. These failures stem from instability issues of the interior–point solver, which at some situations was unable to find a suitable interior point. Opposite to this, RapMinos always found a solution. Not only that, but also did so in considerably less time and with a higher degree of accuracy in the solution. This last fact can be explained by realizing that while the Eν–SVM approach finds a solution by using a series of linear approximations to the non–convex Eν–SVM problem, RapMinos instead addresses the non–convex ERCH–NPP problem directly. As a whole, RapMinos is able to find better–quality solutions consistently at a lower computational cost.

Regarding the improvements of RapMinos over a basic subgradient projection method, Table 2 shows how RapMinos was able to find a solution much faster for most of the data sets. Some notable exceptions are *breastcancer*, *diabetes* and *flare*, where the simple subgradient method finds a good solution quite fast. Table 3 reveals additional insight into this: the solutions found by RapMinos tend to produce better objective values. Which is to say, subgradient projection might return a solution faster in some settings, but performs a worse optimization job. It is thus clear that RapMinos is a better solver for the ERCH–NPP problem than a basic subgradient projection method, as we hypothesized when we proposed the method.

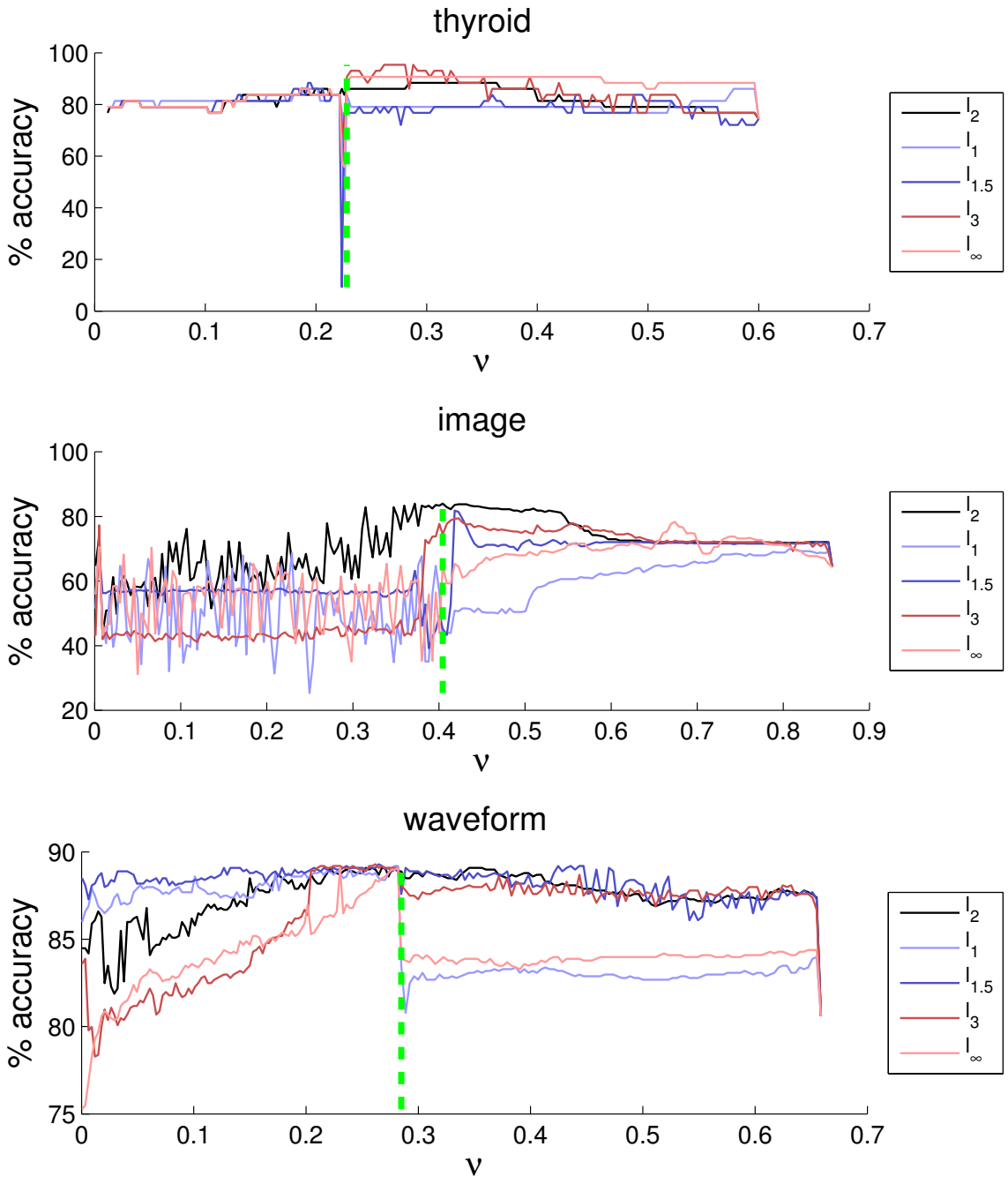


Figure 10: Performance of the Eν–SVM model for a set of real–world data sets and different values of the ℓ_p–norm.

Data set	Eν–SVM solver		Subgrad. Proj.		RapMinos	
	Accuracy	Time	Accuracy	Time	Accuracy	Time
thyroid	80.8%	1.46	86.3%	20.29	86.3%	0.15
heart	82.6%	1.72	73.9%	21.46	73.9%	0.37
titanic	76.5%	1.80	77.82%	29.84	72.9%	0.30
breastcancer	78.7%	1.05	76.6%	0.23	72.3%	0.35
diabetes	73.5%	3.21	75.1%	0.10	74.7%	0.47
flare	–	–	63.0%	0.01	63.3%	0.20
german	66.56%	2.98	77.3%	1.48	77.3%	1.33
banana	–	–	60.5%	44.96	60.5%	0.53
image	–	–	82.1%	35.89	75%	1.15
ringnorm	77.1%	26.85	77.1%	81.95	77.1%	7.24
splice	51.9%	33.27	83.7%	46.42	84.2%	9.76
twonorm	97.7%	24.05	97.2%	61.33	97.2%	11.02
waveform	78.8%	14.85	86.9%	54.21	86.9%	7.94

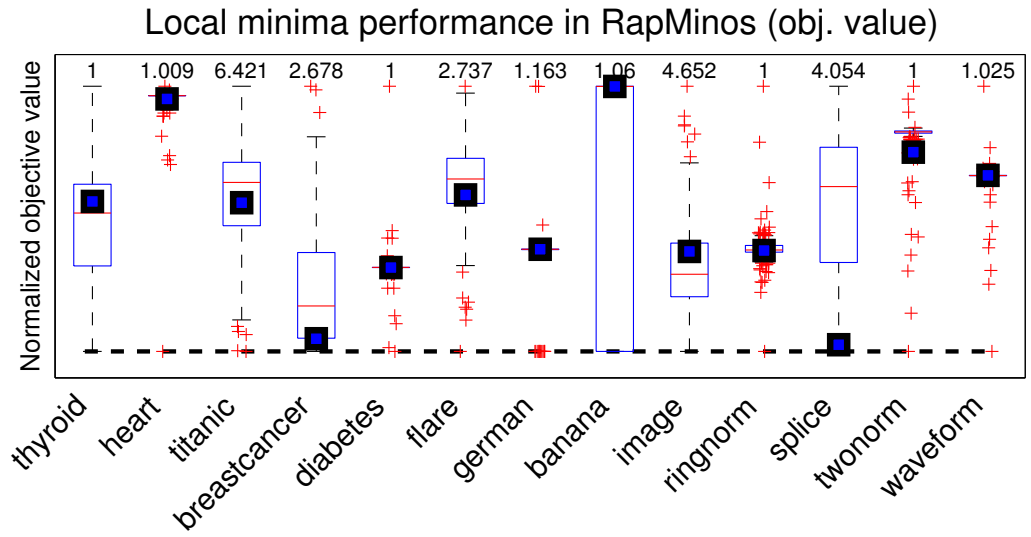
Table 2: Execution times (in seconds) and accuracy in the test set for the reference Eν–SVM solver, the proposed RapMinos algorithm and a simple subgradient projection method. Entries marked with – stand for executions where the Eν–SVM solver failed to produce a solution at all.

6.5 Quality of Local Minima

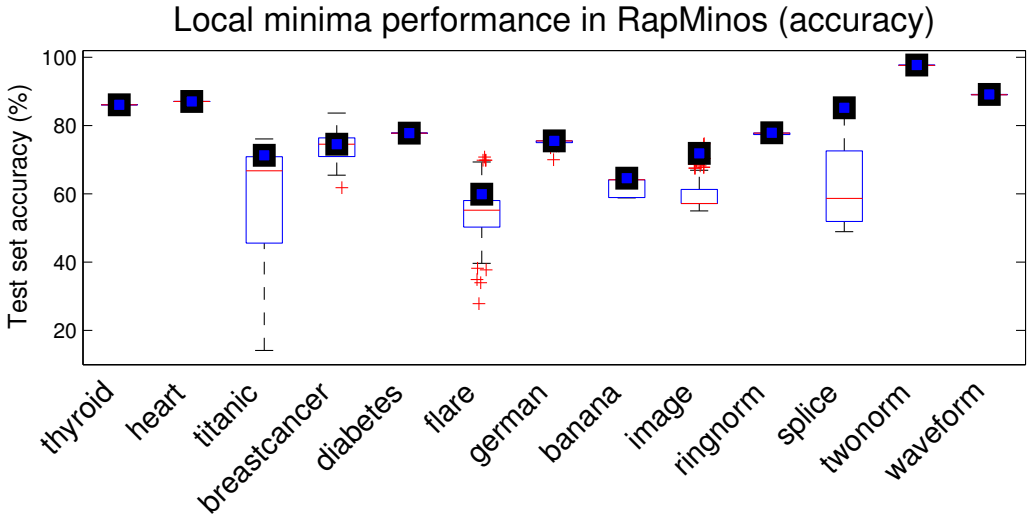
Since in the intersecting case of ERCH–NPP the optimization problem becomes non–convex (see Section 4), RapMinos only finds a local minimum of the problem. Such local minimum might or might not have an objective value similar to the overall global minimum of the problem, and so it might be the case that RapMinos finds a "bad local minimum" where a poor solution is obtained. This kind of problem is quite similar to the issues appearing in multilayer neural network training (Duda et al., 2001), where the non–linearity of the model allows to find only locally optimal solutions. Although several approaches have been proposed to address this issue, the most effective ones involve heuristics for model weights initialization that, while not guaranteeing global optimality, provide some practical means to avoid bad local minima.

In section 5.5 we proposed a heuristic to select the starting point for RapMinos . We will show now that such initialization strategy proves to be helpful in avoiding local minima. For doing so, for each data set in section 6.3 we ran the RapMinos algorithm using the presented approach, and compared the value of the objective function (Equation 7) against 200 runs with random starting points. We fixed $p = 2$ and chose ν as the one giving the highest validation performance, and for those data sets where ν was in the separable range, we chose a ν value slightly below the one for which hulls start intersecting. This way all tests were run for the intersecting case.

Figure 11a presents box plots on the distribution of such objective value for all data sets, comparing also against the value obtained with the proposed initialization. Being



(a)



(b)

Figure 11: Distribution of a) objective values (lower is better) and b) accuracies (higher is better), obtained by RAPMINOS for several data sets. The box plots represent the distribution of objective values and accuracies for the runs with random initialization, and the square markers the value obtained when using the proposed initialization heuristic. Objective values are normalized to present the best minimum found at the bottom line, while the worst one is shown at the top along with a multiplier representing how far away it is from the best value (worst = multiplier · best). A multiplier value of 1 is shown when the best and worst values are equal down to the fourth significant digit.

Data set	RapMinos	Subgrad. Proj.
thyroid	0.179	0.233
heart	0.589	0.586
titanic	-0.808	2.283
breastcancer	1.194	1.411
diabetes	0.679	0.748
flare	5e-09	-0.001
german	-9e-07	0.053
banana	1.072	1.109
image	-4e-06	0.011
ringnorm	0.080	0.099
splice	1.305	1.287
twonorm	0.755	0.755
waveform	1.783	1.759

Table 3: Objective values after optimization in RapMinos and a simple subgradient projection method. Lower is better.

a heuristic procedure, our proposal does not guarantee good local minima in all cases, though nevertheless finds solutions closer to the overall best minimum more frequently than employing a random initialization. Figure 11b presents analogous results when measuring accuracy on the test set, where again a random initialization performs worse than our proposed heuristic initialization.

7. Conclusions and Further Work

In this work we have given a geometrical interpretation of the Eν–SVM formulation, establishing connections from this model to other well–known models in the SVM family. Not surprisingly, while Eν–SVM generalizes ν–SVM to cover the case where ν is too small, this new interpretation generalizes the usual geometric viewpoint of ν–SVM finding the nearest points of two non–intersecting reduced convex hulls (RCH–NPP). Specifically, it also allows these reduced–convex–hulls to intersect, that is, it also covers the case where the reduction η coefficient is too large.

We have also proposed the RapMinos method and shown how it is able to solve the ERCH–NPP problem efficiently and for any choice of $\ell_{p\geq 1}$–norm. This not only allows to build Eν–SVM models faster than with previously available methods, but also provides even more modeling capabilities to the SVM through the flexibility to work with these different norms.

From the light of the experiments, it would seem that the Eν–SVM model can improve classification accuracy for those problems where there is a significant intersection between class hulls. The added ℓ_p–norm flexibility has also proven to be useful to increase classification accuracy in a number of data sets, extending further the applicability of the model.

A number of interesting extensions to this work, which would require further research efforts, are possible. While the RapMinos method finds a solution efficiently and we provide some empirical evidence on it being a reasonably good local minimum, the method is still far from finding global minima. Even though finding global minimizers for non–convex problems is a daunting challenge, a globalization strategy based on concavity cuts has already been developed for the Eν–SVM model (Takeda and Sugiyama, 2008). Whether this approach is also applicable to the dual ERCH–NPP problem is an open issue. Finally, in this paper we have only addressed linear models. Extending the methods here to address kernelized models is also an open problem.

Acknowledgments

This work has been partially supported by Spain's TIN 2010–21575–C02–01 and TIN 2013–42351–P projects and by Cátedra UAM–IIC en Modelado y Predicción.

Appendix A. Proof for Theorem 10 (Optimum at Non–Differentiable Points)

Consider the Minkowski polygon representation of the ERCH-NPP (Eq. 17). If the constraint $||W||_p = 1$ is ignored, the problem would become

$$\min_{W} \quad \max_{X \in \mathcal{M}} \quad W \cdot X.$$

This problem clearly involves the minimization of a piece–wise linear function, where the pieces are determined by the inner maximization $\max_{X \in \mathcal{M}} \quad W \cdot X$. Consider now one of such pieces, which we shall denote S. For every $W \in S$ the inner maximization problem selects the same solution X_S, and so the minimization in this piece can be written as

$$\min_{W \in S} \quad W \cdot X_S.$$

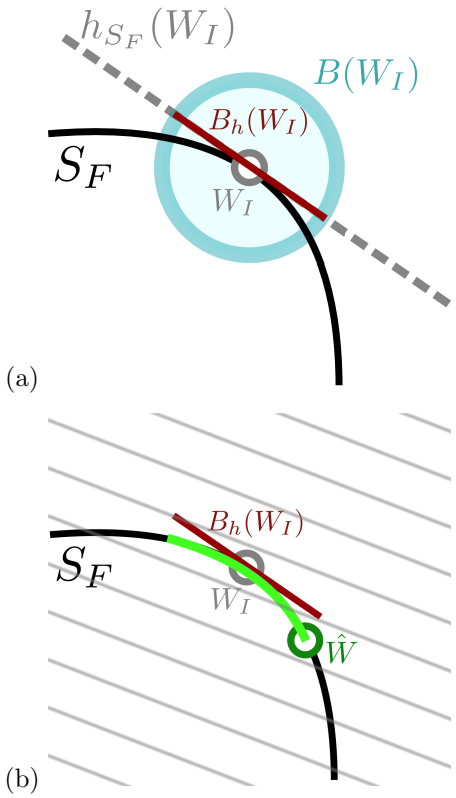
Since this is a linear problem, the optimum necessarily lies at a boundary point of S, that is, at the frontier with another linear region of the global problem, this frontier being a non–differentiable region. However, when taking the constraint back into account we have

$$\min_{W \in S, ||W||_p = 1} \quad W \cdot X_S.$$

which is no longer a linear problem, since the norm constraint on W defines a non–convex feasible set. Hence, the minimum in this linear region need not lie at an extreme. Nevertheless, we show in what follows that this property is still met regardless of this constraint.

Let us denote S_F as the feasible region within S, that is, $S_F \equiv \{W | W \in S, ||W||_p = 1\}$. This region is a surface which is a subset of the ℓ_p unit–ball. To show that the minimum in this region always lies at an extreme point we will assume that, on the contrary, the optimum is in a non–extreme point W_I. We will then see that there always exists another point in a neighborhood of W_I presenting a better or equal value of the objective function.

Consider the supporting hyperplane of S_F at W_I, $h_{S_F}(W_I)$ (see Definition 13). This hyperplane can always be defined for any interior point of S_F as the hyperplane tangent to S_F at W_I. This hyperplane leaves all of S_F at one side. Consider also a ball $B(W_I)$ of small radius $r > 0$, centered on W_I, which shall be understood as a neighborhood of W_I. Let us define $B_h(W_I)$ as the intersection of this ball and the supporting hyperplane, $B_h(W_I) \equiv B(W_I) \bigcap h_{S_F}(W_I)$. This set does define a convex set, since it is the intersection of a hyperplane and a sphere. Because of that, the objective function $W \cdot X_S$ for $W \in B_h(W_I)$ always has a minimizer at an extreme of the set. More precisely, $\exists\, v^* \in B_h(W_I), v^* \neq W_I$ so that $v^* \cdot X_S \leq W_I \cdot X_S$. Thus, there exists a small displacement along a support hyperplane from a non–extreme point W_I that cannot worsen the value of the objective function. But of course, v^* might not be a feasible point, since by the properties of the supporting hyperplane all the points $v \in B_h(W_I)$ have $||v||_p \geq 1$.



(c) Visual example of the concepts introduced for the proof of Theorem 10. a shows the feasible region within a linear region of the problem (S_F), the supporting hyperplane at an interior point of this region ($h_{S_F}(W_I)$), the ball defining the neighborhood ($B(W_I)$) and its intersection with the supporting hyperplane ($B_h(W_I)$). b shows how this intersection can be projected back to the feasible region S_F, and how an extreme of it is able to obtain a better value of the objective function (represented through its level sets as gray lines).

The next step is showing that projection of v^* back to the feasible region S_F still guarantees that the projected point cannot be worse than the initial W_I in terms of the value of the objective function. First, it must be realized that the radial projection $R_p[v]$ for

any $v \in B_h(W_I)$ always results in feasible points inside S_F. This is immediate by realizing that the radial projection just rescales the norm of its vector argument, and so

$$\arg\min_{X\in\mathcal{M}} R_p\left[v\right]\cdot X = \arg\min_{X\in\mathcal{M}} \frac{v\cdot X}{||v||_p} \equiv \arg\min_{X\in\mathcal{M}} v\cdot X,$$

i.e., the solution of the internal problem does not change, and so the projected v remains in the same linear region S. Using then the properties of the radial projection, $||R_p\left[v\right]||_p = 1$, and so $R_p\left[v\right] \in S_F$.

Now note that since we also have that $\forall\ v \in B_h(W_I)$, $||v||_p \geq 1$, then the radially projected points can be defined as $R_p\left[v\right] = \frac{v}{||v||_p} = c(v)\ v$, for some scalar $c(v) \in (0,1]$. Also, since $W\cdot X_S \geq 0$, $\forall\ W \in S_F$ (because of the intersecting hulls, see Lemma 7), we can establish the following chain of relationships

$$\begin{aligned}\min_{v\in B_h(W_I)} R_p\left[v\right]\cdot X_S &= \min_{v\in B_h(W_I)} c(v)\ v\cdot X_S \\ &\leq \min_{v\in B_h(W_I)} v\cdot X_S \\ &\leq W_I\cdot X_S.\end{aligned}$$

Therefore, any non–extreme point $W_I \in S_F$ has always a feasible neighbor which presents an equal or better value of the objective function, and so W_I cannot be optimal (or at least there exists another point with an equally optimal value). Extending this argument to every non–extreme point in S_F, we can conclude that there exists an extreme point W_E such that $W_E\cdot X_S \leq W\cdot X_S$, $\forall\ W \in S_F$. Consequently, a minimizer of the global problem always lies at the intersection between two linear regions, that is to say, at a non–differentiable point. ■

Appendix B. Proof for Theorem 12 (Descent Directions for ERCH)

To prove this theorem we need to resort to some tools from the field of non–convex non–smooth analysis, most of them contained in Clarke (1990). Nevertheless, for completeness of the paper we will briefly introduce such required tools here.

Consider a general constrained optimization problem in the form

$$\begin{aligned}\min_{x\in X}\quad & f(x),\\ \text{s.t.}\quad & g_i(x)\leq 0,\quad i=1,\ldots,n,\end{aligned}$$

where any equality constraint in the form $h(x) = 0$ can also be taken into account by producing two inequality constraints $h(x) \leq 0$, $h(x) \geq 0$.

We introduce now the concept of relative subdifferential as

Definition 21 *Relative subdifferential: given the set $S \subseteq X$, the S–relative subdifferential of f at x, $\partial|_S f(x)$ is defined as*

$$\partial|_S f(x) = \left\{\xi\,|\xi_i\to\xi, \xi_i\in\partial f(y_i), y_i\in S, y_i\to x\right\},$$

that is to say, it is the set of subgradients appearing when approaching x from a succession of points y_i tending to x. In the event that $x \notin S$, $\partial|_S f(x) = \emptyset$.

Consider now the augmented objective function

$$F(x) = \max\left\{f(x) - f(x^*), g_1(x), \ldots, g_n(x)\right\},$$

where $f(x^*)$ is the optimal value of the original objective function. Observe that at the optimum of the original problem, $F(x^*) = 0$, since all constraints are met ($g_i(x) \leq 0$) and the first term takes the value 0. Let us define the set

$$\Gamma(x) = \text{conv}\left\{\partial f(x), \partial|_{G_1(x)} f(x), \ldots, \partial|_{G_n(x)} f(x)\right\},$$

where $G_i(x)$ is the set of points for which the constraint $g_i(x)$ is not feasible ($g_i(x) > 0$). $\Gamma(x)$ can be interpreted as a kind of subdifferential of the Lagrangian. We then have two results associated with this set (Clarke, 1990, Theorem 6.2.2. and Proposition 6.2.4.):

- If x is a local minimum of the problem, then $0 \in \Gamma(x)$.
- Else, let γ be the element of $\Gamma(x)$ with minimum norm. Then $d = -\gamma$ is a descent direction in $F(x)$.

In other words, if we are not already at the optimum, performing a small step in the direction of d reduces the value of the augmented function $F(x)$. Note that, given the form of $F(x)$, this guarantees that either the objective function $f(x)$ or the violation in some constraint is reduced.

Let us apply now these tools to the ERCH problem $\min_W f(W)$ s.t. $||W||_p = 1$. The augmented function $F(W)$ comes easily as

$$F(W) = \max\left\{f(W) - f(W^*), ||W||_p - 1, 1 - ||W||_p\right\},$$

where the equality constraint has been rewritten as two inequalities. Now, taking into account the fact that in our algorithm we guarantee $||W||_p = 1$ at every iteration, the max in $F(W)$ is always attained for the first term when not at the optimum. Also because of this we have that $\partial|_{G_1}||W||_p = \partial|_{(||W||_p > 1)}||W||_p = \partial||W||_p$, and similarly for $\partial|_{G_2}||W||_p$. That is to say, the relative subdifferential coincides with the standard one. Therefore, the set $\Gamma(W)$ results to be

$$\Gamma(W) = \text{conv}\left\{\partial f(W), \partial||W||_p, -\partial||W||_p\right\}.$$

We can rewrite this set in a more convenient form as

$$\begin{aligned}\Gamma(W) &= \mu_1 \partial f(W) + \mu_2 \partial||W||_p - \mu_3 \partial||W||_p, \\ &= \mu_1 \partial f(W) + (\mu_2 - \mu_3)\partial||W||_p,\end{aligned}$$

where the convex coefficients meet the usual constraints $\sum_i \mu_i = 1$, $0 \leq \mu_i \leq 1$. It should be realized now that the gradient of the norm $\partial||W||_p$ is the 0 vector only at the origin $W = 0$,

which is an infeasible point. Therefore, at the optimal W^* it will be necessary to combine this gradient with $\partial f(W)$ to produce the 0 vector bound to appear at a local minimum in $\Gamma(W)$, and so the coefficient must be non–zero, $\mu_1 > 0$. We can then divide the expression by μ_1 [7], obtaining

$$\begin{aligned}\Gamma(W) &\equiv \partial f(W) + \frac{\mu_2 - \mu_3}{\mu_1}\partial||W||_p, \\ &= \partial f(W) + \lambda\partial||W||_p,\end{aligned}$$

for $\lambda = \frac{\mu_2-\mu_3}{\mu_1} \in \mathbb{R}$. It is realized now that the expression obtained for $\Gamma(W)$ is actually the standard subdifferential of the Lagrangian.

Invoking now the properties of the set $\Gamma(x)$ stated above, it is immediate that at local minimum $\arg\min_W ||\Gamma(W)|| = 0$. Descent in the original function $f(x)$ is also obtained by realizing that the direction $d = -\arg\min_W ||\Gamma(W)||$ guarantees descent in $F(W)$, and so at a point $W' = W + sd$, with $s > 0$ sufficiently small,

$$\begin{aligned}f(W') - f(W*) &< \max\left\{f(W') - f(W^*), ||W'||_p - 1,\right. \\ &\qquad \left. 1 - ||W'||_p\right\}, \\ &= F(W') < F(W), \\ &= \max\left\{f(W) - f(W^*), ||W||_p - 1,\right. \\ &\qquad \left. 1 - ||W||_p\right\}, \\ &= f(W) - f(W^*),\end{aligned}$$

since at W the constraints are met. Therefore $f(W') < f(W)$, and so d is also a descent direction for $f(W)$, concluding the proof. ■

Appendix C. Computation of the Derivative of the Constraint

Depending on the actual value of the norm parameter $p \geq 1$, the norm function $||W||_p^p$ might produce a singleton or a set of subgradients. For even p the norm function is smooth an thus produces a singleton subgradient in the form

$$\begin{aligned}\left[\frac{\partial||W||_p^p}{\partial W}\right]_k &= \frac{\partial}{\partial W_k}\sum_i (W_i)^p, \\ &= p\,(W_k)^{p-1}.\end{aligned}$$

However, for an odd or non–integer value of p the absolute value function cannot be disposed of, and the set of subgradients produced takes the form

7. Even though this transformation changes the scaling of the points in the set $\Gamma(W)$, note that the argument remains legit, since we are only interested in extracting a direction vector from $\Gamma(x)$, and therefore scaling is not relevant.

$$\begin{aligned}\left[\frac{\partial||W||_p^p}{\partial W}\right]_k &= \frac{\partial}{\partial W_k}\sum_i |W_i|^p\,,\\ &= p\,|W_k|^{p-1}\,\frac{\partial}{\partial W_k}|W_k|,\\ &= p\,|W_k|^{p-1}\,\mu_k,\end{aligned}$$

where the coefficients μ_k take the values

$$\mu_k = \begin{cases} 1 & \text{if} \quad W_k > 0,\\ -1 & \text{if} \quad W_k < 0,\\ [-1,1] & \text{if} \quad W_k = 0.\end{cases}$$

That is to say, for values of W with entries at 0 several possible subgradients appear. Nevertheless, since if $W_k = 0$ then $|W_k|^p = 0$ (except for $p = 1$, see below), the particular choice of μ_k is irrelevant, and we end up at

$$\left[\frac{\partial||W||_p^p}{\partial W}\right]_k = p\,|W_k|^{p-1}\,\mathrm{sign}(W_k).$$

as shown in Eq. (23).

The cases $p = 1$ and $p = \infty$, which are of special relevance for their known sparsity/uniformity inducing properties, require some further attention. First, for $p = 1$ we have

$$\left[\frac{\partial||W||_1}{\partial W}\right]_k = \frac{\partial}{\partial W_k}\sum_i |W_i| = \mu_k,$$

and a similar situation to that of the general p arises, though this time the particular choice of μ_k does produce different subgradients. This is not surprising, since the ℓ_1–norm is non–smooth. To address this issue, in this paper we take the simplest of the available subgradients, taking $\mu_k = 0$ whenever $W_k = 0$, resulting in

$$\left[\frac{\partial||W||_1}{\partial W}\right]_k = \mathrm{sign}(W_k).$$

It must be noted, however, that by making this simplification we might be failing to identify the correct updating directions in our algorithm when standing on a W point where the norm is not differentiable. This, however, poses no problems to our method in practice, but for very specifically tailored cases unlikely to arise in practice. Even in those cases the solution of the ERCH with norm ℓ_1 can be safely approximated by a norm choice like $\ell_{1.001}$, which is smooth.

Now for $p = \infty$ the derivative is, in principle, not separable, since we have

$$\frac{\partial||W||_\infty}{\partial W} = \frac{\partial}{\partial W}\max\{|W_i|\}\,.$$

Nevertheless we can rewrite this as

$$\frac{\partial||W||_\infty}{\partial W} = \frac{\partial}{\partial W}\max\{W_1, -W_1, \dots W_n, -W_n\},$$

and invoke again the property that the subdifferential of the maximum of a set of convex functions (linear, in this case) at a given point is the convex hull of the subdifferentials of the functions attaining such maximum at that point (Boyd and Vandenberghe, 2007). With this, we obtain that

$$\left[\frac{\partial||W||_\infty}{\partial W}\right]_k = \left\{\begin{array}{lll} 0 & \text{if} & |W_k| < \max_j\{|W_j|\}, \\ \tau_i & \text{if} & W_k = \max_j\{|W_j|\}, \\ -\tau_i & \text{if} & -W_k = \max_j\{|W_j|\}, \end{array}\right.$$

with τ_i the convex hull coefficients, i.e.,

$$\sum_{i\in I}\tau_i = 1, \;\; I \equiv \left\{i : |W_i| = \max_j\{|W_j|\}\right\}.$$

Now, since the scale of $\frac{\partial||W||_\infty}{\partial W}$ is not relevant (only its orientation) and by picking only the most convenient subgradient we arrive at

$$\left[\frac{\partial||W||_\infty}{\partial W}\right]_k = \left\{\begin{array}{ll} 0 & \text{if} \quad |W_k| < \max_j\{|W_j|\}, \\ \text{sign}(W_i) & \text{if} \quad |W_k| = \max_j\{|W_j|\}. \end{array}\right.$$

The same comments than those for norm ℓ_1 apply here; if needed, the ℓ_∞ norm can be approximated by a large norm such as ℓ_{100}.

Appendix D. General $\ell_{p\geq 1}$ RCH-NPP Solver

The generalized $\ell_{p\geq 1}$ RCH-NPP problem takes the form

$$\min_{X_+\in\mathcal{U}_+, X_-\in\mathcal{U}_-} \|X_+ - X_-\|_p, \tag{31}$$

for $p \geq 1$ and sets $\mathcal{U}_\pm$ defined as in Proposition 2. Such problem is an instance of a common family of problems arising in machine learning in the form

$$\min_x f(x) + r(x),$$

for f convex and differentiable, r convex and lower semicontinuous, but not necessarily differentiable. Such problems are addressed efficiently by making use of a proximal method (see Combettes and Pesquet 2009 for a thorough review), as long as two basic ingredients are provided: a subroutine to compute the gradient of f and an efficient method to solve the proximity operator of r, an optimization subproblem taking the form

$$\text{prox}_r(y) \;\equiv\; \min_x \frac{1}{2}||x-y||_2^2 + r(x).$$

Problem 31 can be written in $\min_x f(x) + r(x)$ form by defining

$$\begin{array}{rcl} x & = & \left[\begin{array}{c} X_+ \\ X_- \end{array}\right], \\ f(x) & = & \|X_+ - X_-\|_p, \\ r(x) & = & \iota_{\mathcal{U}_+}(X_+) + \iota_{\mathcal{U}_-}(X_-), \end{array}$$

where $\iota_{\mathcal{C}}(x)$ is an indicator function valued 0 if $x \in \mathcal{C}$, $+\infty$ else. Using the results of the previous appendix the gradient of f can be shown to take the form

$$\nabla f(x) = \begin{bmatrix} \left(\frac{|X_+ - X_-|}{||X_+ - X_-||_q}\right)^{q-1} \operatorname{sign}(X_+ - X_-) \\ -\left(\frac{|X_+ - X_-|}{||X_+ - X_-||_q}\right)^{q-1} \operatorname{sign}(X_+ - X_-) \end{bmatrix},$$

while the proximity operator of r is

$$\begin{aligned} \operatorname{prox}_r(y) &\equiv \min_x \frac{1}{2}||x-y||_2^2 + \iota_{\mathcal{U}_+}(X_+) + \iota_{\mathcal{U}_-}(X_-), \\ &= \left\{\min_{X_+} \frac{1}{2}||X_+ - Y_+||_2^2 + \iota_{\mathcal{U}_+}(X_+)\right\} + \left\{\min_{X_-} \frac{1}{2}||X_- - Y_-||_2^2 + \iota_{\mathcal{U}_-}(X_-)\right\}, \\ &= \left\{\min_{X_+ \in \mathcal{U}_+} \frac{1}{2}||X_+ - Y_+||_2^2\right\} + \left\{\min_{X_- \in \mathcal{U}_-} \frac{1}{2}||X_- - Y_-||_2^2\right\}, \end{aligned}$$

where y has also been decomposed in two parts Y_+ and Y_-. It is evident now that the proximity operator can be computed by solving two independent subproblems, which turn out to be instances of the classic RCH–NPP where one of the hulls is a singleton $Y_\pm$. Such problem is solved through trivial modifications of a standard RCH–NPP solver.

In our RapMinos implementation we make use of the FISTA proximal algorithm (Beck and Teboulle, 2009), which by the inclusion of the aforementioned gradient and proximity subroutines results in an effective $\ell_{p\geq 1}$ RCH-NPP solver.

It is also worth pointing out that for the extreme ℓ_1 and ℓ_∞ cases problem (31) becomes non–differentiable, preventing the use of the presented approach. Still, a solution is easily attainable by realizing that in these two cases the minimization of the norm function can be rewritten as a set of linear constraints, as

$$\begin{aligned} \min_x ||x||_1 &= \min_x \sum_i \max\{x_i, -x_i\} = \min_{x,z} \sum_i z_i \quad \text{s.t.} \quad z_i \geq x_i, -x_i \; \forall\, i, \\ \min_x ||x||_\infty &= \min_x \max\{|x_1|, \ldots, |x_d|\} = \min_{x,z} z \quad \text{s.t.} \quad z \geq x_i, -x_i \; \forall\, i. \end{aligned}$$

Hence, the whole problem is rewritten as a Linear Program, which we solve by making use of Matlab's internal LP solver routine *linprog*.

Appendix E. Bias Computation in ERCH–NPP

When no reduction of the hulls is applied in RCH–NPP the usual procedure to compute the bias is to take it in such a way that the classification hyperplane lies at the middle of the extreme points in the convex–hulls ($b = -\frac{1}{2}W \cdot (X_+ + X_-)$ for the optimal solution X_+ and X_- of Eq. 7). However, such bias value is not necessarily equivalent to the one obtained when solving ν–SVM, as already pointed out by Crisp and Burges (2000). The same situation holds for Eν–SVM, and so we show here how to compute the correct value of b.

The KKT complementary slackness conditions of the inner minimization problem in ERCH–Margin (Eq. 5) are the following

$$\begin{aligned} \lambda_i(W \cdot X_i - \alpha + \xi_i) = 0 \qquad & \forall\, i \in M_+, \\ \lambda_i(W \cdot X_i - \beta - \xi_i) = 0 \qquad & \forall\, i \in M_-, \\ \xi_i \mu_i = 0 \qquad & \forall\, i, \end{aligned}$$

from which, together with the relationships obtained from the derivatives of the Lagrangian (Eq. 8) the following statements can be derived

- If $i \in M_+, \lambda_i > 0 \implies W \cdot X_i - \alpha + \xi_i = 0$.
- If $i \in M_-, \lambda_i > 0 \implies W \cdot X_i - \beta - \xi_i = 0$.
- If $\lambda_i < \eta \implies \mu_i > 0 \implies \xi_i = 0$.

Joining these three facts we can compute α by finding an $i \in M_+$ s.t. $0 < \lambda_i < \eta$, as for this case $W \cdot X_i - \alpha = 0$, and similarly for β, obtaining

$$\begin{aligned} \alpha = W \cdot X_i \qquad & \text{for some } i \in M_+, 0 < \lambda_i < \eta, \\ \beta = W \cdot X_i \qquad & \text{for some } i \in M_-, 0 < \lambda_i < \eta. \end{aligned}$$

Once α and β are known the bias can be computed through the definitions of these two terms (see the proof for Proposition 1), as

$$b = -\frac{1}{2}(\alpha + \beta). \tag{32}$$

Therefore, for any given W in ERCH–Margin or ERCH–NPP its corresponding bias can be computed with the obtained formula. A similar derivation was already proposed in Chang and Lin (2001) for the ν–SVM, though the connection with RCH–Margin was not made.

It should be noted, however, that the presented bias computation requires the sets $i \in M_+, 0 < \lambda_i < \eta$ and $i \in M_-, 0 < \lambda_i < \eta$ to be non–empty. If one of them turns out to be empty, which is a not so uncommon situation in practice, the bias cannot be computed in closed form. In such cases lower and upper bounds on b can be derived from the KKT conditions, as done in Chang and Lin (2001). We follow such procedure to obtain bounds on b and pick some value in the admissible range. Another possible solution would be to determine the bias as the one maximizing classification accuracy over the training set, that is

$$b^* = \arg\max_b \sum_{i \in M} \text{sign}\left\{y_i(X_i \cdot W + b)\right\}.$$

Such problem is solvable in log–linear time by sorting all the $X_i \cdot W$ values and counting the number of correct labellings for each possible b between all couples of consecutive $X_i \cdot W$ values. Even though this procedure seems to be more solid than selecting b from some loose bounds, it is actually prone to overfitting. Only in settings where the training data presents low noise have we found this procedure to produce better test accuracies, and thus we recommend resorting instead to the bounds provided by the KKT conditions.

References

A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal of Imaging Sciences*, 2(1):183–202, 2009.

K.P. Bennett and E.J. Bredensteiner. Duality and geometry in SVM classifiers. In *Proceedings of the 17th International Conference on Machine Learning*, pages 57–64, 2000.

D.P. Bertsekas. *Nonlinear Programming*. Athena Scientific, 1995.

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

S. Boyd and L. Vandenberghe. Subgradients. Notes for EE364b, Stanford University, Winter 2006-07, January 2007.

C.-C. Chang and C.-J. Lin. *LIBSVM: a Library for Support Vector Machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

F. H. Clarke. *Optimization and Nonsmooth Analysis*. Classics in Applied Mathematics. SIAM, 1990.

P.L. Combettes and J.-C. Pesquet. Proximal splitting methods in signal processing. *arXiv:0912.3522*, 2009.

C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

D.J. Crisp and C.J.C. Burges. A geometric interpretation of ν–SVM classifiers. In *Advances in Neural Information Processing Systems*, volume 12, 2000.

R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification*. Wiley-Interscience. Wiley & Sons, New York, 2nd edition, 2001.

C. Ericson. The Gilbert-Johnson-Keerthi algorithm. Technical report, Sony Computer Entertainment America, 2005.

D.G. De Figueiredo and L.A. Karlovitz. On the radial projection in normed spaces. *Bulletin of the American Mathematical Society*, 1967.

K. Huang, D. Zheng, I. King, and M.R. Lyu. Arbitrary norm support vector machines. *Neural Computation*, 21(2):560–582, 2009.

S. S. Keerthi, S. K. Shevade, C. Bhattacharyya, and K. R. K. Murthy. A fast iterative nearest point algorithm for support vector machine classifier design. *IEEE Transactions on Neural Networks*, 11(1):124–136, 2000.

J. López, Á. Barbero, and J.R. Dorronsoro. On the equivalence of the SMO and MDM algorithms for SVM training. In *Lecture Notes in Computer Science: Machine Learning and Knowledge Discovery in Databases*, volume 5211, pages 288–300. Springer, 2008.

J. López, Á. Barbero, and J.R. Dorronsoro. Clipping algorithms for solving the nearest point problem over reduced convex hulls. *Pattern Recognition*, 44(3):607–614, 2011a.

J. López, K. De Brabanter, J.R. Dorronsoro, and JAK Suykens. Sparse LS-SVMs with ℓ_0–norm minimization. ESANN, 2011b.

D.G. Luenberger and Y. Ye. *Linear and Nonlinear Programming*. Springer, 2008.

M.E. Mavroforakis and S. Theodoridis. A geometric approach to support vector machine (SVM) classification. *IEEE Transactions on Neural Networks*, 17(3):671–682, 2006.

M.E. Mavroforakis, M. Sdralis, and S. Theodoridis. A geometric nearest point algorithm for the efficient solution of the SVM classification task. *IEEE Transactions on Neural Networks*, 18(5):1545–1549, 2007.

F. Pérez-Cruz, J. Weston, D.J.L. Hermann, and B. Schölkopf. Extension of the ν–SVM range for classification. In *Advances in Learning Theory: Methods, Models and Applications*, volume 190, pages 179–196, 2003.

G. Rätsch. *Benchmark Repository*, 2000. Datasets available at `http://www.raetschlab.org/Members/raetsch/benchmark`.

R.T. Rockafellar. *Convex Analysis*, volume 28 of *Princeton Mathematics Series*. Princeton University Press, 1970.

R.T. Rockafellar and S. Uryasev. Conditional value-at-risk for general loss distributions. *Journal of Banking & Finance*, 26(7):1443–1472, 2002.

B. Schölkopf, A.J. Smola, R.C. Williamson, and P.L. Bartlett. New support vector algorithms. *Neural Computation*, 12(5):1207–1245, 2000.

Y. Shi, Y. Tian, G. Kou, Y. Peng, and J. Li. Feature selection via ℓ_p–norm support vector machines. In *Optimization Based Data Mining: Theory and Applications*, pages 107–116. Springer, 2011.

A. Takeda and M. Sugiyama. ν–support vector machine as conditional value-at-risk minimization. In *Proceedings of the 25th International Conference on Machine Learning*, pages 1056–1063, 2008.

A. Takeda and M. Sugiyama. On generalization and non–convex optimization of extended ν–support vector machine. *New Generation Computing*, 27:259–279, 2009.

A. Takeda, H. Mitsugi, and T. Kanamori. A unified classification model based on robust optimization. *Neural Computation*, 25 (3):759–804, 2013.

J. Zhu, S. Rosset, T. Hastie, and R. Tibshirani. 1–norm support vector machines. In S. Thrun, L.K. Saul, and B. Schölkopf, editors, *Neural Information Processing Systems*. MIT Press, 2003.