Pattern Recognition 44 (2011) 607–614

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Clipping algorithms for solving the nearest point problem over reduced convex hulls

Jorge López *, Álvaro Barbero, José R. Dorronsoro

Departamento de Ingeniería Informática and Instituto de Ingeniería del Conocimiento, Universidad Autónoma de Madrid, 28049 Madrid, Spain

ARTICLE INFO

Article history:
Received 8 February 2010
Received in revised form
1 July 2010
Accepted 27 August 2010

Keywords:
Support vector machines
Nearest point problem
Reduced convex hulls
GSK algorithm
MDM algorithm

ABSTRACT

The nearest point problem (NPP), i.e., finding the closest points between two disjoint convex hulls, has two classical solutions, the Gilbert–Schlesinger–Kozinec (GSK) and Mitchell–Dem'yanov–Malozemov (MDM) algorithms. When the convex hulls do intersect, NPP has to be stated in terms of reduced convex hulls (RCHs), made up of convex pattern combinations whose coefficients are bound by a $\mu < 1$ value and that are disjoint for suitable μ. The GSK and MDM methods have recently been extended to solve NPP for RCHs using the particular structure of the extreme points of a RCH. While effective, their reliance on extreme points may make them computationally costly, particularly when applied in a kernel setting. In this work we propose an alternative clipped extension of classical MDM that results in a simpler algorithm with the same classification accuracy than that of the extensions already mentioned, but also with a much faster numerical convergence.

1. Introduction

Given a linearly separable two class sample $\mathcal{S} = \{(X_i, y_i) : y_i = \pm 1, 1 \le i \le N\}$, the nearest point problem (NPP) is to find the two nearest points in the convex hulls $C(\mathcal{S}_\pm)$ of the positive $\mathcal{S}_+ = \{X_i : y_i = 1\}$ and negative $\mathcal{S}_- = \{X_i : y_i = -1\}$ subsamples. Since the difference $W = W_+ - W_-$ between two points $W_+ \in C(\mathcal{S}_+)$ and $W_- \in C(\mathcal{S}_-)$ can be written as $W = \sum_1^N \alpha_i y_i X_i$, with $\sum_{y_i = 1} \alpha_i = \sum_{y_j = -1} \alpha_j = 1$, NPP can be formulated as

$$\min \quad \frac{1}{2} \|W_+ - W_-\|^2 = \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j X_i \cdot X_j$$
$$\text{s.t.} \quad 0 \le \alpha_i \le 1, \; \sum_i \alpha_i y_i = 0, \; \sum_i \alpha_i = 2. \tag{1}$$

When the sample is not linearly separable the convex hulls do intersect and the relevant problem is then to find the nearest points between the μ–reduced convex hulls (μ–RCH), i.e.,

$$C_\mu(\mathcal{S}_\pm) = \left\{ \sum_{y_i = \pm 1} \alpha_i X_i : \sum_i \alpha_i = 1, 0 \le \alpha_i \le \mu \right\}$$

for an appropriate μ that makes disjoint the corresponding hulls and such that $1/\nu \le \mu < 1$ with $\nu = \min\{N_+, N_-\}$, $N_\pm = |\mathcal{S}_\pm|$; observe that unless the barycenters of the $\mathcal{S}_\pm$ subsets coincide, the reduced hulls $C_\mu(\mathcal{S}_\pm)$ are disjoint for some μ. We will call this problem μ–NPP and, as done before, we can now formulate it as

$$\min \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j X_i \cdot X_j \quad \text{s.t. } 0 \le \alpha_i \le \mu, \; \sum_i \alpha_i y_i = 0, \; \sum_i \alpha_i = 2. \tag{2}$$

When the second class reduces to a single point, that we may assume to be 0, NPP becomes the minimum norm problem (MNP), as its solution is the point in $C(\mathcal{S})$ closest to the origin. Similarly, μ–NPP becomes μ–MNP as now the solution is the point in $C_\mu(\mathcal{S})$ closest to 0.

There are two classical procedures to solve MNP, namely the Gilbert–Schlesinger–Kozinec (GSK; see [1,2]) algorithm and the Mitchell–Dem'yanov–Malozemov (MDM; see [3]) algorithm. While sometimes discussed as different methods, the algorithms of Gilbert on the one hand and Schlesinger–Kozinec on the other, essentially coincide and we will make no distinctions between them. GSK iteratively updates the current vector W as $W' = (1 - \lambda) W + \lambda X_L$ with X_L such that $W \cdot X_L \le W \cdot X_i$ for any other vector in $\mathcal{S}$. The rationale for this is provided by the bound $W \cdot X_L / \|W\| \le \|W^*\| \le \|W\|$, with W^* being the solution, for it can be shown [2] that the GSK update decreases the difference $\|W\|^2 - W \cdot X_L$. The MDM updates are $W' = W + \lambda (X_L - X_U)$ with X_L as before and X_U such that $W \cdot X_U \ge W \cdot X_i$ for all i for which $\alpha_i > 0$; we have now $W \cdot X_L \le \|W\|^2 \le W \cdot X_U$, and it can be shown [3] that the MDM updates decrease $W \cdot X_U - W \cdot X_L$. It is well known [4] that, in practice, MDM has a faster convergence than GSK.

[*] Corresponding author. Instituto de Ingeniería del Conocimiento; C/ Francisco Tomás y Valiente, 11; E.P.S., Edificio B; UAM-Cantoblanco, 28049 Madrid, Spain. Tel.: +34 914972352; fax: +34 914972334.

E-mail addresses: j.lopez@uam.es, jorge.lopez@iic.uam.es (J. López), alvaro.barbero@iic.uam.es (Á. Barbero), jose.r.dorronsoro@iic.uam.es (J.R. Dorronsoro).

0031-3203/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2010.08.030

The classical GSK and MDM algorithms can be easily extended to NPP ([2,4]; see also Section 2), but the situation for $\mu-$NPP or even $\mu-$MNP is more complicated. To begin with, the previous updates no longer guarantee that W' be in $C_\mu(\mathcal{S})$ even if W is. The most natural path to follow is to observe that the previous X_L and X_U choices are extreme points of the sample's convex hulls and to replace them in a RCH setting with extreme points of the reduced hulls. Taking advantage of the concrete structure of these extreme points, Mavroforakis et al. [5] and Tao et al. [6] have recently proposed for GSK to replace X_L by

$$\hat{X} = \mu \sum_{k=1}^{K} X_{i_k} + \sigma X_{i_{K+1}},$$

where we have $K = \lfloor 1/\mu \rfloor$, $\sigma = 1-K\mu$ and X_{i_k}, $1 \le k \le K+1$, are the sample patterns with smallest $W \cdot X_j$ values. This guarantees that $\hat{X}$ is an extreme point of $C_\mu(\mathcal{S})$ such that $W \cdot \hat{X} \le W \cdot Z$ for any other extreme point Z and, moreover, $W' = (1-\lambda)W + \lambda \hat{X} \in C_\mu(\mathcal{S})$ if W is already there.

To extend MDM to $\mu-$MNP, Tao et al. [7] propose the updates $W' = W + \lambda(\hat{X} - X_U)$ with $\hat{X}$ as in extended GSK and X_U as in standard MDM. Under some extra conditions it also turns out that $W' \in C_\mu(\mathcal{S})$ if we already have $W \in C_\mu(\mathcal{S})$.

While effectively solving $\mu-$NNP, these procedures have a clear drawback when applied in a kernel setting as they require K+1 dot products (and, hence, kernel operations) per pattern for the GSK extension and K+2 products for the MDM one. This may become quite costly even when the kernel operations can be cached. An alternative viewpoint is suggested by the relationship between $\mu-$NPP and linear penalty support vector machine (SVM) training [8,9], which in turn builds upon the relationship between NPP and hard-margin SVM training [8,10]. In fact, the dual SVM problem must deal with vectors $W = \sum_i \alpha_i y_i X_i$ subject to restrictions $0 \le \alpha_i \le C$ for some penalty factor C and the standard SMO algorithm for SVM training simply clips adequately the updated α'_i coefficients so that these bounds hold. This suggests to extend to $\mu-$NPP the basic GSK and MDM algorithms for NPP by also clipping the updated coefficients so that we still have $0 \le \alpha'_i \le \mu$. Notice that this procedure reduces the dot products required for cache maintenance to just one per pattern for GSK and two for MDM. However, and as we shall see, this does not work for clipped GSK, as the procedure may get stuck at a vector W far away from the optimal W^*. On the other hand, we will prove that the clipped MDM algorithm we propose will not suffer from such a problem and, as our experiments will illustrate, it does indeed converge to optimal vectors at a much lower computational cost than needed for the other GSK and MDM extensions.

The rest of the paper is organized as follows. In Section 2 we briefly recall the GSK and MDM algorithms for standard NPP using a new viewpoint; while the algorithms are already known, our alternative presentation establishes the basic facts for our subsequent proposals and may also have an interest of its own, as it gives a compact and slightly alternative treatment of the basic algorithms. In Section 3 we briefly recall the GSK extension to $\mu-$NPP by Mavroforakis et al. and the MDM one by Tao et al., considering also their complexity and pointing out some difficulties that may arise with the latter. We present our clipped algorithms in Section 4, giving their details in its first subsection and provide in Section 4.2 an example showing how clipped GSK may get stuck far away from an optimal weight but arguing that this will not be the case for clipped MDM. In Section 4.3 we briefly recall how $\mu-$NPP can be cast as the dual problem of a certain primal one and derive from its Karush–Kuhn–Tucker (KKT) conditions an optimality criterion for $\mu-$NPP which we use to show that our clipped MDM procedure will not get stuck at a non-optimal vector W. Section 5 describes some numerical experiments that confirm our previous analysis. The paper ends with a brief discussion.

2. The GSK and MDM algorithms for NPP

In this section we will briefly review the standard GSK and MDM algorithms, deriving them using a viewpoint alternative to those usually found in the literature and that may be of interest on its own. In our experiments we shall work with kernel versions of both algorithms; thus, we also discuss briefly how to kernelize them and the resulting complexity. There are several good expositions on both algorithms in the literature; for more details on them we refer to [2,5,6] for GSK and to [3,4,11] for MDM.

2.1. The GSK algorithm

The standard GSK algorithm [2] uses a single pattern X_L to update one of the components $W_\pm$ of the current weight vector $W = W_+ - W_-$ to a new one $W'_\pm = (1-\lambda)W_\pm + \lambda X_L$, where X_L is in the same hull of the $W_\pm$ vector being updated. To simplify the notation we will write W_y to indicate W_+ when y=1 and W_- when $y = -1$. Then it is easy to see that the new W vector can be written as $W' = W + \lambda y_L (X_L - W_{y_L}) = W + \lambda y_L Z_L$, with $Z_L = X_L - W_{y_L}$. For instance, if $X_L \in \mathcal{S}_-$, $y_L = -1$ and we update W_- as $W'_- = (1-\lambda)W_- + \lambda X_L$, we have then

$$\begin{aligned} W' &= W_+ - W'_- = W_+ - (1-\lambda)W_- - \lambda X_L = W_+ - W_- - \lambda(X_L - W_-) \\ &= W - \lambda(X_L - W_{y_L}) = W + \lambda y_L (X_L - W_{y_L}) = W + \lambda y_L Z_L; \end{aligned}$$

a similar argument can be used if $X_L \in \mathcal{S}_+$.

Keeping in mind that we want to solve NPP, a natural way to select the index L is to choose it so that the decrease in $\|W'\|^2$ is largest. Notice that $\|W'\|^2$ is a function

$$\Phi(\lambda) = \|W + \lambda y_L Z_L\|^2 = \|W\|^2 + 2\lambda y_L W \cdot Z_L + \lambda^2 \|Z_L\|^2$$

of the dilation factor λ. Now, since $\Phi'(\lambda) = 2(y_L W \cdot Z_L + \lambda \|Z_L\|^2)$, Z_L will give a descent direction provided we have $\Phi'(0) = 2y_L W \cdot Z_L = 2y_L W \cdot (X_L - W_{y_L}) < 0$; that is, provided $y_L W \cdot X_L < y_L W \cdot W_{y_L}$. If this is the case, the λ value that minimizes Φ is obtained solving $\Phi'(\lambda) = 0$ and is given by

$$\lambda = -\frac{y_L W \cdot Z_L}{\|Z_L\|^2} = -y_L \frac{\Delta_L}{\|Z_L\|^2}, \tag{3}$$

where we write $\Delta_j = W \cdot (X_j - W_{y_j})$; notice that $\lambda > 0$ if Z^L is a descent direction. Then, the norm $\|W'\|^2 = \Phi(\lambda)$ of the updated vector becomes

$$\Phi(\lambda) = \|W\|^2 - 2\frac{(y_L W \cdot Z_L)^2}{\|Z_L\|^2} + \frac{(y_L W \cdot Z_L)^2}{\|Z_L\|^2} = \|W\|^2 - \frac{\Delta_L^2}{\|Z_L\|^2}, \tag{4}$$

and if we ignore the $\|Z_L\|^2$ denominator, an approximately maximum decrease in $\|W'\|^2$ will be obtained if we take $|\Delta_L|$ as large as possible. Since we want $y_L \Delta_L < 0$ to hold so that we have a descent direction, we also have to make sure that $\Delta_L < 0$ if $y_L = 1$ and that $\Delta_L > 0$ if $y_L = -1$. We can achieve this if we select first

$$L^+ = \text{argmin}\{W \cdot X_j : y_j = 1\}, \quad L^- = \text{argmax}\{W \cdot X_j : y_j = -1\},$$

compute then $\Delta_{L^+} = W \cdot (X_{L^+} - W_+)$ and $\Delta_{L^-} = W \cdot (X_{L^-} - W_-)$, and select finally $L = L^+$ if $|\Delta_{L^+}| > \Delta_{L^-}$ and $L = L^-$ otherwise; as we shall argue in Section 4.3, $|\Delta_L| > 0$ will hold at any non-optimal W. In any case, notice that, in order to have a convex combination, we have to take the optimum λ^* as $\lambda^* = \min(1,\lambda)$, with λ given by (3) and the new coefficients of W' are

$$\alpha'_i = (1-\lambda^*)\alpha_i + \lambda^* \delta_{iL} \quad \text{if } y_i = y_L; \ \alpha'_i = \alpha_i \ \text{ otherwise},$$

with δ_{jk} denoting Kronecker's delta.

To obtain a kernel version of the preceding, we observe that the previous computations can be expressed in terms of dot products. In fact, let us introduce as in [2,5] the following notation: $A = W_+ \cdot W_+$, $B = W_- \cdot W_-$, $C = W_+ \cdot W_-$, $D_j = W_+ \cdot X_j$, and $E_j = W_- \cdot X_j$, $1 \le j \le N$. We can clearly compute the $L^{\pm}$ indices from the $W \cdot X_j = D_j - E_j$ values and, for instance, we have

$$\Delta_{L^+} = D_{L^+} - E_{L^+} - A + C,$$

$$\|Z_{L^+}\|^2 = \|W_+ - X_{L^+}\|^2 = A - 2D_{L^+} + \|X_{L^+}\|^2,$$

with similar formulae holding for the corresponding L^- values. Moreover, if we take, say, $L = L^+$, then $W'_- = W_-$ and, therefore, $E'_j = E_j$ and $B' = B$; for the other values we have

$$D'_j = W'_+ \cdot X_j = (1-\lambda^*)W_+ \cdot X_j + \lambda^* X_{L^+} \cdot X_j = (1-\lambda^*)D_j + \lambda^* X_{L^+} \cdot X_j,$$

$$\begin{aligned} A' = \|W'_+\|^2 &= (1-\lambda^*)^2\|W_+\|^2 + 2\lambda^*(1-\lambda^*)W_+ \cdot X_{L^+} + (\lambda^*)^2\|X_{L^+}\|^2 \\ &= (1-\lambda^*)^2 A + 2\lambda^*(1-\lambda^*)D_{L^+} + (\lambda^*)^2\|X_{L^+}\|^2, \end{aligned}$$

$$C' = W'_+ \cdot W'_- = (1-\lambda^*)W_+ \cdot W_- + \lambda^* X_{L^+} \cdot W_- = (1-\lambda^*)C + \lambda^* E_{L^+}.$$

Similar formulae hold when $L = L^-$ (see [5] for more details) and it is clear that everything can be extended to a kernel setting where a dot product $X \cdot X'$ is replaced by a kernel operation $k(x,x')$. Moreover, each iteration of the kernel GSK algorithm will require N kernel operations to update either the D or E caches, another one to update either A or B and, finally, another one to compute the denominator in λ.

Although simple and easy to implement, the standard GSK algorithm may have a rather slow convergence. In Ref. [12] a Minkowski version of GSK that updates the current W using both $X_{L^{\pm}}$ patterns, alleviates this drawback, although essentially doubling the number of kernel operations per iteration, as then both the D_j and E_j coefficients have to be updated. However, a still better option is the MDM algorithm that we describe next.

2.2. The MDM algorithm

The standard MDM algorithm [4,11] uses two pattern vectors X_U, X_L to update the current weight W to another one of the form $W' = W + \lambda_L y_L X_L + \lambda_U y_U X_U$. The restrictions on the α_i imply

$$2 = \sum_i \alpha'_i = \sum_i \alpha_i + \lambda_U + \lambda_L = 2 + \lambda_U + \lambda_L,$$

$$0 = \sum_i y_i \alpha'_i = \sum_i y_i \alpha_i + y_U \lambda_U + y_L \lambda_L = y_U \lambda_U + y_L \lambda_L.$$

It follows from the second equation that $y_U \lambda_U = -y_L \lambda_L$ and since the first one gives $\lambda_U = -\lambda_L$, we must also have $y_U = y_L$. As a consequence, the update becomes

$$W' = W + \lambda_L y_L (X_L - X_U) = W + \lambda_L y_L Z_{L,U},$$

where now we set $Z_{i,j} = X_i - X_j$. Thus, writing $\lambda = \lambda_L$, $\|W'\|^2$ is again a function Φ of λ and it follows that

$$\Phi(\lambda) = \|W'\|^2 = \|W\|^2 + 2\lambda y_L W \cdot Z_{L,U} + \lambda^2 \|Z_{L,U}\|^2,$$

$$\Phi'(\lambda) = 2(y_L W \cdot Z_{L,U} + \lambda \|Z_{L,U}\|^2);$$

in particular, writing now $\Delta = W \cdot Z_{L,U} = W \cdot X_L - W \cdot X_U$, we have $\Phi'(0) = 2y_L\Delta$ and $\|W'\|^2$ will decrease provided we find two vectors X_L, X_U such that $y_L \Delta < 0$. As done before, solving $\Phi'(\lambda) = 0$ gives the optimal values

$$\lambda_L = \lambda = -y_L \frac{\Delta}{\|Z_{L,U}\|^2}, \quad \lambda_U = -\lambda_L = y_L \frac{\Delta}{\|Z_{L,U}\|^2}; \tag{5}$$

notice that $\lambda > 0$ when $Z_{L,U}$ defines a descent direction and it follows easily that

$$\|W'\|^2 = \|W\|^2 - \frac{\Delta^2}{\|Z_{L,U}\|^2}. \tag{6}$$

Ignoring the $\|Z_{L,U}\|^2$ denominator, the norm decrease will now be approximately largest if we maximize $|\Delta|$. Since we want $y_L\Delta < 0$ to hold, we have to make sure that $\Delta < 0$ if $y_L = 1$ and that $\Delta > 0$ if $y_L = -1$. Writing $I_{\pm} = \{i : y_i = \pm 1\}$, we may meet these requirements by simply selecting

$$\Delta_+ = \min_{I_+}\{W \cdot X_j\} - \max_{I_+}\{W \cdot X_j\}, \quad \Delta_- = \max_{I_-}\{W \cdot X_j\} - \min_{I_-}\{W \cdot X_j\},$$

which we achieve in principle by choosing the $L^{\pm}, U^{\pm}$ indices as

$$L^+ = \operatorname*{argmin}_{I_+}\{W \cdot X_j\}, \quad U^+ = \operatorname*{argmax}_{I_+}\{W \cdot X_j\},$$

$$L^- = \operatorname*{argmax}_{I_-}\{W \cdot X_j\}, \quad U^- = \operatorname*{argmin}_{I_-}\{W \cdot X_j\}.$$

It is clear now that $\Delta_+ \le 0$ and $\Delta_- \ge 0$ but, as we shall argue in Section 4.3, at least one of them will be non-zero if we have not arrived at an optimal W and, as a consequence, our $L^{\pm}, U^{\pm}$ choices will guarantee descent directions. We finally select $L = L^+, U = U^+$ and $\Delta = \Delta_+$ if $|\Delta_+| > \Delta_-$ and $L = L^-, U = U^-$ and $\Delta = \Delta_-$ otherwise.

However, these choices still must be slightly refined. Notice that since the W' updates can be written as $W' = W + \lambda y_L X_L - \lambda y_U X_U$, the new multipliers are

$$\alpha'_L = \alpha_L + \lambda, \quad \alpha'_U = \alpha_U - \lambda, \quad \alpha'_j = \alpha_j \text{ otherwise.}$$

In particular, $\alpha'_L > \alpha_L$ and $\alpha'_U < \alpha_U$. Thus α_U must be > 0 to begin with, and we must refine the previous $U^{\pm}$ choices to

$$U^+ = \operatorname*{argmax}_{I_+}\{W \cdot X_j : \alpha_j > 0\}, \quad U^- = \operatorname*{argmin}_{I_-}\{W \cdot X_j : \alpha_j > 0\}. \tag{7}$$

Moreover, the λ value in (5) must be adequately clipped so that $\alpha'_L \le 1$ and $\alpha'_U \ge 0$; this implies that we must take a λ^* such that

$$\lambda^* = \min\{\lambda, 1-\alpha_L, \alpha_U\}. \tag{8}$$

It is clear that the MDM algorithm can also be expressed in terms of dot products and, thus, can be easily extended to a kernel setting. For an efficient implementation we have to cache now a vector D with the inner products $D_j = W \cdot X_j$ from which we can easily compute the required values of Δ and λ. The D vector is updated as

$$D'_j = W' \cdot X_j = (W + y_L\lambda^*(X_L - X_U)) \cdot X_j = D_j + y_L\lambda^*(X_L \cdot X_j - X_U \cdot X_j),$$

with a cost of $2N$ kernel operations per iteration. While being twice the cost of a GSK iteration, this is more than compensated since MDM requires a much lower total number of iterations.

Once more, notice that in the preceding we compute four indices $L^{\pm}$, $U^{\pm}$ but use only two of them. As it was the case with the GSK algorithm, this may suggest to replace the standard MDM update by a Minkowski-like one in which all the four vectors $X_{L^{\pm}}$, $X_{U^{\pm}}$ are used. However, this also doubles the number of kernel operations to be done at each iteration and, at least in our experiments, there is no advantage over the standard MDM algorithm. We turn now our attention to NPP algorithms for reduced convex hulls.

3. Algorithms for NPP over reduced convex hulls

For $0 < \mu \le 1$ we shall denote by $C_\mu(\mathcal{A})$ the μ–reduced convex hull (RCH) of a set $\mathcal{A}$, i.e.,

$$C_\mu(\mathcal{A}) = \left\{ \sum_j \alpha_j X_j : X_j \in \mathcal{A}, 0 \le \alpha_j \le \mu, \sum_j \alpha_j = 1 \right\}.$$

It is clear that the GSK updates will keep a weight vector $W_{\pm}$ in $C_{\mu}(\mathcal{S}_{\pm})$ provided we make sure the updating patterns $X_{L^{\pm}}$ are also in $C_{\mu}(\mathcal{S}_{\pm})$. However, the choices of the $L^{\pm}$ indices given in the preceding section make this unlikely. To overcome this, we observe that the GSK $X_{L^{\pm}}$ patterns are in fact extreme points of the positive and negative class convex hulls. The papers [5] by Mavroforakis et al. and [6] by Tao et al. suggest to replace these standard GSK updating patterns by extreme sample vectors $\hat{X}_{L^{\pm}}$ of the $\mu-$reduced convex hulls giving the largest norm decreases, which here means selecting

$$L^{+}=\operatorname{argmin}\{W\cdot\hat{X}:\hat{X}\in\mathcal{E}(\mathcal{S}_{+})\},\quad L^{-}=\operatorname{argmax}\{W\cdot\hat{X}:\hat{X}\in\mathcal{E}(\mathcal{S}_{-})\},$$

where $\mathcal{E}(\mathcal{S}_{\pm})$ denote the extreme points of the $\mu-$reduced convex hulls.

To avoid a costly combinatorial characterization of these extreme points, in [5] the authors give a thorough discussion of the geometrical properties of reduced convex hulls $C_{\mu}(\mathcal{S}_{\pm})$, showing that the extreme points are precisely of the form $\hat{X}=\mu\sum_{k=1}^{K}\tilde{X}_{p_k}+\sigma\tilde{X}_{p_{K+1}}$, with $K=\lfloor 1/\mu\rfloor$, $\sigma=1-K\mu$, and proposing a simple way to select the appropriate $\hat{X}_{L^{\pm}}$. In it, one first separately sorts the subsets $\mathcal{D}_{+}=\{D_j=W\cdot X_j:y_j=1\}$ in ascending order and $\mathcal{D}_{-}=\{D_j=W\cdot X_j:y_j=-1\}$ in descending order and then renumber them according to these sortings, denoting by $\mathcal{I}_{\pm}$ the associated index sets and by $\tilde{X}_j$ the patterns thus renumbered; i.e., $\tilde{X}_1$ with $1\in\mathcal{I}_{+}$ corresponds to the pattern with the smallest dot product $W\cdot\tilde{X}_j, y_j=1$ and $\tilde{X}_1$ with 1 now in $\mathcal{I}_{-}$ corresponds to the pattern with the largest dot product $W\cdot\tilde{X}_j$, $y_j=-1$. We consider next two possible updating vectors

$$\hat{X}_{+}=\mu\sum_{k=1}^{K}\tilde{X}_{k}+\sigma\tilde{X}_{K+1},\quad \hat{X}_{-}=\mu\sum_{k'=1}^{K}\tilde{X}_{k'}+\sigma\tilde{X}_{K+1},$$

where the indices for $\hat{X}_{+}$ are taken from $\mathcal{I}_{+}$, those for $\hat{X}_{-}$ from $\mathcal{I}_{-}$ and, as before, we have $K=\lfloor 1/\mu\rfloor$ and $\sigma=1-K\mu$. Clearly $\hat{X}_{\pm}\in C_{\mu}(\mathcal{S}_{\pm})$ and, as shown in [5], they are the extreme points giving a largest norm decrease. As in standard GSK, we compute then $\Delta_{+}=W\cdot(\hat{X}_{+}-W_{+})$, $\Delta_{-}=W\cdot(\hat{X}_{-}-W_{-})$. If, in analogy to what was done before, we use the notations $y_L=+1$, $\Delta_L=\Delta_{+}$ and $\hat{X}_L=\hat{X}_{+}$ when $|\Delta_{+}|>\Delta_{-}$, and $y_L=-1$, $\Delta_L=\Delta_{-}$ and $\hat{X}_L=\hat{X}_{-}$ when $|\Delta_{+}|<\Delta_{-}$, the W update can be written as $W'=W+\lambda^{*}y_L(\hat{X}_L-W_{y_L})$ with the optimal λ^{*} being given by

$$\lambda^{*}=\min\left\{1,-y_L\frac{\Delta_L}{\|W_{y_L}-\hat{X}_L\|^{2}}\right\}. \tag{9}$$

We shall argue in Section 4.3 that the $\hat{X}_{\pm}$ choices ensure that $\Delta_{+}<0$ or $\Delta_{-}>0$. Thus, we also have here $\lambda^{*}>0$ and obtain a descent direction.

The resulting algorithm, which we will call $\mu-$GSK from now on, is shown in [5] to converge to good models and, as illustrated in [12], its Minkowski variant, where both $\hat{X}_{\pm}$ are used, has a faster convergence. On the other hand, the required sorting of the dot products adds typically an $O(N\log N)$ complexity to that of standard GSK. Moreover, in a kernel setting we have also to consider the cost of each updating step. In fact, assume that we update W with $\hat{X}_{+}$. Then, as in standard GSK, we update the D_j cache as

$$\begin{aligned}D'_j=W'_{+}\cdot\tilde{X}_j&=(1-\lambda^{*})W_{+}\cdot\tilde{X}_j+\lambda^{*}\hat{X}_{+}\cdot\tilde{X}_j\\&=(1-\lambda^{*})D_j+\lambda^{*}\mu\sum_{k=1}^{K}\tilde{X}_k\cdot\tilde{X}_j+\lambda^{*}\sigma\tilde{X}_{K+1}\cdot\tilde{X}_j\end{aligned}$$

and the E_j one similarly. It is thus clear that these updates will require at least KN kernel operations per iteration, i.e., about K times the cost of the previous GSK update. If K is large (i.e., μ small), this may be prohibitive when the kernel matrix cannot be cached and still very costly even if it is so.

Turning our attention to the extension of the MDM algorithm to the RCH setting, a first option along the previous lines would be to choose the $U_{\pm}$ indices as

$$\begin{aligned}U^{+}&=\operatorname{argmax}\{W\cdot\hat{X}_j:\hat{X}_j\in\mathcal{E}(\mathcal{S}_{+}),\hat{\alpha}_j>0\},\\U^{-}&=\operatorname{argmin}\{W\cdot\hat{X}_j:\hat{X}_j\in\mathcal{E}(\mathcal{S}_{-}),\hat{\alpha}_j>0\},\end{aligned}$$

where now $\hat{\alpha}_j$ denote the $\hat{X}_j$ coefficient in the W expansion. However, this last condition is hard to check as there is no simple way of keeping track of the $\hat{\alpha}_j$ coefficients. To overcome this, Tao et al. propose in [7] an MDM extension to $\mu-$NPP where we choose $\hat{X}_{L^{\pm}}$ as in $\mu-$GSK and $\tilde{X}_{U^{\pm}}$ as in standard MDM, and compute $\Delta_{\pm}=W\cdot(\hat{X}_{\pm}-\tilde{X}_{U^{\pm}})$. Using again the notations $y_L=+1$, $U=U^{+}$, $\Delta_L=\Delta_{+}$ and $\hat{X}_L=\hat{X}_{+}$ when $|\Delta_{+}|>\Delta_{-}$ and $y_L=-1$, $U=U^{-}$, $\Delta_L=\Delta_{-}$ and $\hat{X}_L=\hat{X}_{-}$ when $|\Delta_{+}|<\Delta_{-}$, the W updates are now $W'=W+\lambda y_L(\hat{X}_L-\tilde{X}_U)$, with the optimal λ being, in principle, given by

$$\lambda=-y_L\frac{\Delta_L}{\|\hat{X}_L-\tilde{X}_U\|^{2}}. \tag{10}$$

The new multipliers become

$$\alpha'_k=\alpha_k+\lambda\mu,\quad 1\le k\le K,$$

$$\alpha'_{K+1}=\alpha_{K+1}+\lambda\sigma,$$

$$\alpha'_U=\alpha_U-\lambda,$$

$$\alpha'_j=\alpha_j\quad\text{otherwise},$$

and some α_j selection and λ clipping may therefore have to be applied.

However, this choice is not problem free, as Fig. 1 shows, where we are taking $\mu=\frac{1}{2}$, assume the negative hull to be reduced to the 0 point and have for some α:

$$W=\frac{\alpha}{2}(X_1+X_2)+\frac{1-\alpha}{2}(X_2+X_3)=\frac{1}{2}X_2+\frac{\alpha}{2}X_1+\frac{1-\alpha}{2}X_3.$$

It can be easily checked that $\hat{X}_L=\frac{1}{2}(X_1+X_2)$ and $\tilde{X}_U=X_3$. The algorithm's update would then be

$$W'=W+\lambda(\hat{X}_L-\tilde{X}_U)$$

for some $\lambda>0$, which is clearly not feasible, as we would have then $\alpha'_2=\frac{1}{2}+\lambda$ for some positive λ, i.e., $\alpha'_2>\frac{1}{2}$. In other words, the optimal point $\frac{1}{2}(X_1+X_2)$ would not be attained since the algorithm gets stuck in the current estimate W. Note also that, however, GSK would indeed progress from W to $\hat{X}_L$ in a single iteration.

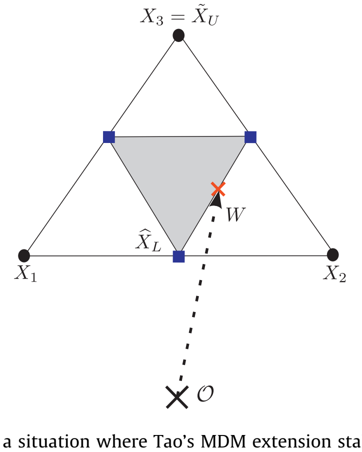

Fig. 1. Example of a situation where Tao's MDM extension stalls without finding the optimum (the lowest vertex of the reduced hull with $\mu=\frac{1}{2}$). The direction chosen by the algorithm $(\hat{X}_L-\tilde{X}_U)$ is clearly infeasible for the point W, which is a suboptimal estimate of $W^{*}=\hat{X}_L$.

Even if this last situation does not arise in a concrete experiment, it is again clear that, besides the cost of the sortings that are also required here, the updates of the algorithm in [7], that we will call $\mu-$MDM, will require at least $(K+1)N$ kernel operations per iteration, i.e., $(K+1)/2$ times that of the previous MDM update. Once again, this may be prohibitive for large K. To alleviate this we consider next two simple extensions to $\mu-$NPP of the standard GSK and MDM algorithms in which we simply clip the updating factor λ^* so that the new coefficients remain $\leq \mu$. While not effective for the GSK algorithm, we shall see that the resulting "clipped" MDM algorithm solves $\mu-$NPP much more efficiently than either the $\mu-$GSK or the $\mu-$MDM algorithms just discussed.

4. Clipped algorithms for $\mu-$NPP

4.1. Clipped GSK and MDM algorithms

We first notice that the bounds $0 \leq \alpha_i \leq \mu$ that $\mu-$NPP places on the α_i multipliers are formally very similar to the constraints $0 \leq \alpha_i \leq C$ that linear penalty SVM with coefficient C places on the feasible solutions of its dual problem. The relationship between $\mu-$NPP and SVM is well known (see for instance [8] or [9]) and there is a clear formal similarity between MDM and the well known SMO algorithm for SVM training. However, the various SMO implementations for linear penalty SVM just clip adequately the α updates of the penalty-free case so that the α bounds are met and it is thus natural to consider whether simply clipping the standard GSK and MDM updates would result in effective algorithms for $\mu-$NPP.

We begin the discussion with the GSK algorithm. Let X_L be the updating vector chosen in Section 2. Recall that the resulting coefficient updates are $\alpha_i' = (1-\lambda)\alpha_i + \lambda\delta_{iL}$ for $y_i = y_L$ and $\alpha_i' = \alpha_i$ otherwise, and we clearly have $0 \leq \alpha_i' \leq \alpha_i \leq \mu$ if $i \neq L$. However, if $i = L$ the update becomes $\alpha_L' = (1-\lambda)\alpha_L + \lambda = \alpha_L + (1-\alpha_L)\lambda$ and for α_L' to be $\leq \mu$, we must use a λ^* value such that

$$\lambda^* = \min\left\{\lambda, \frac{\mu-\alpha_L}{1-\alpha_L}\right\}, \tag{11}$$

with λ given by (3). Moreover, it is also clear that we cannot consider X_i patterns with $\alpha_i = \mu$ as updating candidates. Thus, we must also refine the previous $L_{\pm}$ choices as

$$L^+ = \text{argmin}\{W \cdot X_i : y_i = 1, W \cdot X_i < W \cdot W_+, \alpha_i < \mu\},$$

$$L^- = \text{argmax}\{W \cdot X_j : y_j = -1, W \cdot X_j > W \cdot W_-, \alpha_j < \mu\}; \tag{12}$$

recall that the $W \cdot X_i < W \cdot W_+$ and $W \cdot X_j > W \cdot W_-$ conditions are needed so that we obtain descent directions. We will call the resulting algorithm $\mu-$clipGSK. The algorithm clearly stays in the reduced convex hulls if it starts there and the cost in kernel operations of a $\mu-$clipGSK update is now N, the same of standard GSK and much smaller than that of the $\mu-$GSK, as discussed in the previous section.

Turning our attention to the MDM algorithm, since it only increases the α_{L^+} and α_{L^-} coefficients, in a $\mu-$RCH setting we can simply keep our previous $U^{\pm}$ choices in (7) and refine the $L^{\pm}$ choices as

$$L^+ = \underset{j}{\text{argmin}}\{W \cdot X_j : y_j = 1, \alpha_j < \mu\},$$

$$L^- = \underset{j}{\text{argmax}}\{W \cdot X_j : y_j = -1, \alpha_j < \mu\}. \tag{13}$$

Now we make our final L, U and Δ choices as done for standard MDM. Moreover, since we must have $\alpha'_L = \alpha_L + \lambda^* \leq \mu$ and $\alpha'_U = \alpha_U - \lambda^* \geq 0$ after the optimal L, U have been chosen, we must select the final λ^* value as

$$\lambda^* = \min\left(-y_L \frac{\Delta}{\|Z_{L,U}\|^2}, \mu-\alpha_L, \alpha_U\right). \tag{14}$$

Again, the algorithm, which we call $\mu-$clipMDM, clearly stays in the reduced convex hulls if it starts there and the cost in kernel operations of a $\mu-$ClipMDM update is now $2N$, the same of standard MDM and much smaller than that of the $\mu-$MDM variant of the previous section.

It is therefore clear that $\mu-$ClipGSK and $\mu-$ClipMDM offer simple algorithms that potentially may solve $\mu-$NPP. However, as we shall see next over a concrete example, this is not the case for $\mu-$ClipGSK, whose training may get stuck far away from an optimal weight. On the other hand, we will show that $\mu-$ClipMDM is not affected by these problems and, as we shall see, is much faster than the $\mu-$GSK and $\mu-$MDM alternatives.

4.2. A concrete example

When applying the index L selection in the $\mu-$clipGSK algorithm, there is the obvious risk that the right-hand side index sets in (12) be empty. How this may happen is shown in Fig. 2, where a convex hull formed by four vectors is depicted. We assume $\mu = \frac{1}{2}$, $W_- = (0,0)$, $W = W_+ = \frac{1}{2}(X_1 + X_4)$ to be the point at the center of the convex hull (which obviously also lies inside the reduced hull) and the α_i coefficients of the four X_i points being the ones shown in the figure. It is clear that the only possible choices for X_L would be the points X_2 and X_3 with a zero coefficient, but for them $W \cdot X_i = W \cdot W_+$ and the choice of λ^* in (11) results in $\lambda^* = 0$ and $W' = W$. Thus, no decrease in $\|W\|^2$ is achieved, no updating vector could then be found and $\mu-$clipGSK would be stuck at a clearly wrong W weight, as the optimal one is given by $\frac{1}{2}X_4 + \frac{1}{4}(X_2 + X_3)$. We performed some experiments and while this condition never happened numerically, the convergence of $\mu-$clipGSK was quite slow and sometimes led to poor models.

On the other hand, we observe that the previously described situation does not hamper the progress of the $\mu-$clipMDM algorithm. In fact, notice that under the assumptions made before, we have $U^+ = 1$ but we cannot choose $L^+ = 4$ since $\alpha_4 = \frac{1}{2} = \mu$. Thus we take now, say, $L^+ = 2$ to arrive at $W' = W + \lambda(X_2 - X_1)$; moreover, writing $P = X_1 - X_2$ we have

$$\lambda = -\frac{W \cdot (X_2 - X_1)}{\|X_2 - X_1\|^2} = \frac{W \cdot P}{\|P\|^2} = \frac{\|W\|}{\|P\|}\cos\phi,$$

with ϕ the angle between W and P. It follows from (14) that

$$\lambda^* = \min\left\{\frac{\|W\|}{\|P\|}\cos\phi, \mu-\alpha_2, \alpha_1\right\} = \min\left\{\frac{\|W\|}{\|P\|}\cos\phi, \frac{1}{2}, \frac{1}{2}\right\}.$$

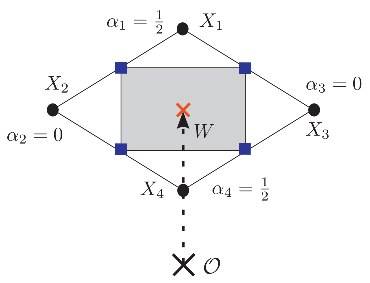

Fig. 2. Example of a situation where $\mu-$clipGSK stalls without finding the optimum, as any movement towards the two possible choices for X_L (either X_2 or X_3) would produce an increase of the norm $\|W\|$.

In this case we would like $\lambda^* = \frac{1}{2}$, i.e., $\lambda \geq \frac{1}{2}$, so that we get rid of X_1. This will be now the case if $\|W\|\cos\phi \geq \|P\|/2$. If, however, $\lambda < \frac{1}{2}$, here again X_1 will stay and the convergence will go on although perhaps more slowly. In any case, contrary to the GSK situation, the procedure does not stop. In fact, and as a consequence of our discussion in the following section, it will turn out that the clipped MDM algorithm will not get stuck at non-optimal W vectors.

4.3. *An optimality condition for* $\mu-$RCH *NPP*

It is well known [8] that NPP for $\mu-$RCH (2) can be written as the dual of a particular primal problem. More precisely, consider for a sample $\mathcal{S} = \{(X_i, y_i) : y_i = \pm 1, 1 \leq i \leq N\}$ and a fixed μ the following quadratic programming problem:

$$\min_{W,\gamma,\rho,\xi} \frac{1}{2}\|W\|^2 - \gamma + \rho + \mu\sum_i \xi_i \tag{15}$$

subject to the restrictions $\xi_i \geq 0, W \cdot X_i \geq \gamma - \xi_i$ when $y_i = 1$ and $W \cdot X_i \leq \rho + \xi_i$ when $y_i = -1$. Then standard Lagrangian computations can be applied to show that (2) is the dual of (15) (see [8] for more details). Moreover, the KKT conditions for (2) at optimum values $W^*, \gamma^*, \rho^*, \xi^*, \alpha^*$ are

$$\alpha_i^*(W^* \cdot X_i - \gamma^* + \xi_i^*) = 0 \quad \text{when } y_i = 1, \tag{16}$$

$$\alpha_i^*(W^* \cdot X_i - \rho^* - \xi_i^*) = 0 \quad \text{when } y_i = -1, \tag{17}$$

$$(\mu - \alpha_i^*)\xi_i^* = 0. \tag{18}$$

Assume $y_i = 1$; if $\alpha_i^* < \mu$ it follows from (18) that $\xi_i^* = 0$ and, hence, from the feasibility conditions on W^*, that $W^* \cdot X_i \geq \gamma^*$. On the other hand, if $\alpha_i^* > 0$, then (16) gives $W^* \cdot X_i = \gamma^* - \xi_i^* \leq \gamma^*$. In other words, if we define

$$b_U^+(W) = \max\{W \cdot X_i : y_i = 1, \alpha_i > 0\},$$
$$b_L^+(W) = \min\{W \cdot X_i : y_i = 1, \alpha_i < \mu\}$$

at an optimum W^* we must have $b_U^+(W^*) \leq b_L^+(W^*)$. When $y_i = -1$, a similar reasoning shows that $W^* \cdot X_i \leq \rho^*$ when $\alpha_i^* < \mu$, and $W^* \cdot X_i \geq \rho^*$ when $\alpha_i^* > 0$. Thus, defining now

$$b_L^-(W) = \max\{W \cdot X_i : y_i = -1, \alpha_i < \mu\},$$
$$b_U^-(W) = \min\{W \cdot X_i : y_i = -1, \alpha_i > 0\},$$

we must have now $b_L^-(W^*) \leq b_U^-(W^*)$.

On the other hand, if the current W is not optimal, either $b_U^+(W) > b_L^+(W)$ or $b_L^-(W) > b_U^-(W)$ or both must hold. Notice that the $L^\pm$ and $U^\pm$ choices in (7) and (13) imply

$$b_L^+(W) = W \cdot X_{L^+}, \quad b_L^-(W) = W \cdot X_{L^-},$$

$$b_U^+(W) = W \cdot X_{U^+}, \quad b_U^-(W) = W \cdot X_{U^-},$$

and, moreover, $\Delta_+ = b_L^+ - b_U^+$ and $\Delta_- = b_L^- - b_U^-$. Thus, if W is not optimal, the L, U choices of clipped MDM guarantee that we will have $y_L\Delta < 0$ for the final choice of Δ and, hence, that they provide a descent direction. In other words, contrary to what the previous example shows for clipped GSK, clipped MDM will not get stuck at a non-optimal W. Moreover, this same reasoning can be used to show that the index choices in the standard GSK and MDM algorithms provide descent directions and the same is true for $\mu-$GSK, since it is equivalent to standard GSK in the reduced hull.

5. Numerical experiments

In this section we will compare the performance of the $\mu-$GSK and $\mu-$ClipMDM algorithms on the 13 G. Rätsch's benchmark datasets [13]: *Titanic* (T), *Heart* (H), *Diabetes* (D), *Breast Cancer* (BC), *Thyroid* (Th), *Flare* (F), *Splice* (S), *Image* (I), *German* (G), *Banana* (B), *Twonorm* (Tw), *Ringnorm* (R) and *Waveform* (W). In particular, we have used in each problem the train-test splits given in that reference (100 for each dataset except for *Image* and *Splice*, where only 20 splits are given). Table 1 contains for each dataset the pattern dimensions, the training and test data sizes and the test accuracies reported in [13] and computed on the corresponding splits with a RBF kernel and the C and σ values also shown in Table 1.

In our RCH experiments we have kept the table's σ^2 values and derived the μ values from the SVM C parameters as follows. In [14] it is shown that if W^o and ξ_i^o are the optimal values for a lineal penalty SVM problem with a C parameter, then $W^* = 2W^o/\rho$ is the optimal solution for a $\mu-$RCH NPP problem, with ρ and μ given as

$$\rho = \|W^o\|^2 + C\sum \xi_i^o, \quad \mu = \frac{2C}{\rho}. \tag{19}$$

Note that for a given C, the corresponding μ depends also on the optimal solution and, ultimately, on the partition used. Thus, a different μ value would be obtained for each one of the train–test splits. To arrive at a single μ value for a given problem, we have applied the above equality over all the training splits, after the SMO algorithm has been executed using the parameters in Table 1 and stopped as soon as the KKT maximal violation is less than 10^{-6} (see [14]).

The μ value finally selected is the minimum of the μ values given by (19) over each training split of a given dataset. When this minimum value is too small for a given split, that is, when $\mu < \mu_{\min} = \min\{1/N^+, 1/N^-\}$, it is set to $\mu_{\min}$. Similarly, if this minimum value is greater than 1, i.e., if the training splits are actually linearly separable, it is then set to $\mu = 1$; this is the case of the *Ringnorm* dataset for which a quite large $C = 10^9$ value is given in Table 1. The values finally obtained for the parameter μ are also reported in Table 1.

In all our experiments we have started the algorithms at the barycenter of the training hulls and used as stopping criterion the same than in [5] for GSK, that is, when

$$\|W\|^2 - \min_{i \in I^\pm}\{y_i W \cdot (X_i - W^\mp)\} \leq \varepsilon\|W\|,$$

where we use an $\varepsilon = 10^{-5}$ tolerance for all datasets. Since $\mu-$GSK is a rather costly algorithm, we also establish a maximum number of 10^5 iterations to be done if the above criterion is not met.

Table 1
Number of dimensions, training and test patterns, together with the accuracies (in %) reported in G. Rätsch's benchmark for an SVM with the RBF kernel and the reported hyperparameters.

Set	Dim	N_{Tr}	N_{Te}	SVM err.	C	σ^2	RCH μ
T	3	150	2051	22.4 ± 1.0	5.0	2.0	0.0222
H	13	170	100	16.0 ± 3.3	3.2	120.0	0.0232
D	8	468	300	23.5 ± 1.7	1.0	20.0	0.0074
BC	9	200	77	26.0 ± 4.7	15.2	50.0	0.0181
Th	5	140	75	4.8 ± 2.2	10.0	3.0	0.1456
F	9	666	400	32.4 ± 1.8	1.0	30.0	0.0041
S	60	1000	2175	10.9 ± 0.7	1000.0	70.0	0.7029
I	18	1300	1010	3.0 ± 0.6	500.0	30.0	0.0214
G	20	700	300	23.6 ± 2.1	3.2	55.0	0.0053
B	2	400	4900	11.5 ± 0.7	316.2	1.0	0.0215
Tw	20	400	7000	3.0 ± 0.2	3.2	40.0	0.0416
R	20	400	7000	1.7 ± 1.1	10^9	10.0	1.0000
W	21	400	4600	9.9 ± 0.4	1.0	20.0	0.0138

The μ values used for the RCH experiments are also shown.

We have compared the resulting models over four different performance measures. The first two ones are the accuracy of the final classifiers obtained and the number of support vectors (SVs), i.e., sample patterns X_i for which the optimal α_i^* coefficients are not zero. Recall that a large number of SVs may make quite slow the application of the classifier to new unseen patterns. The second pair of performance values are the number of iterations and of kernel operations required to meet the above stopping criterion.

Table 2 gives the average and standard deviations of the errors and the number of support vectors of the final models obtained by the two algorithms. As it can be seen, the classification errors are in general very similar and a Wilcoxon rank-sum test fails to find any significant differences at the 10% level for the $\mu-$GSK or $\mu-$ClipMDM methods. Regarding support vectors, in general, $\mu-$ClipMDM arrives to models with less support vectors than $\mu-$GSK. This is natural since $\mu-$ClipMDM can get rid of unnecessary support vectors in a single iteration by making $\alpha_U = 0$. This is not possible for $\mu-$GSK, where unnecessary support vectors can only be asymptotically removed.

Regarding the computational burden, Table 3 reports the number of iterations and kernel operations required to meet the stopping criterion. It can be seen that $\mu-$ClipMDM requires less iterations for all datasets except *Heart* and *Diabetes*. However, when kernel operations are considered, $\mu-$ClipMDM is clearly superior, something to be expected, as one $\mu-$GSK iteration requires about $1/\mu$ kernel operations, while a $\mu-$ClipMDM one requires basically 2. As mentioned, an improvement for standard $\mu-$GSK is given in [12] and called Minkowski GSK, as it simultaneously updates the W_+ and $W-$. Standard and Minkowski GSK performances are compared in [12] for the *Diabetes, Flare, German, Banana, Heart* and *Waveform* datasets. On the first two problems Minkowski GSK requires more kernel operations than standard GSK but it has a better behavior in the other datasets, particularly on the *Heart* problem, where it needs about 150 times less kernel operations. On the other datasets it needs between 1.34 and 8.15 less operations. On the other hand, and it can be seen in Table 3, $\mu-$ClipMDM requires between 20 and 390 less kernel operations than $\mu-$GSK on these datasets, clearly an improvement superior than that of Minkowski $\mu-$GSK for all datasets except *Heart*. In short, these results show that $\mu-$ClipMDM is in general a more efficient algorithm to solve the NPP problem for $\mu-$RCHs, as it gives sparser models, usually requires less iterations and performs a much lower number of kernel operations.

We finally point out that we also performed the same experiments above using the $\mu-$MDM algorithm. In computational terms, its behavior lies between that of $\mu-$ClipMDM and of $\mu-$GSK, something to be expected, as it can get rid of wrong support vectors more efficiently than GSK (and, thus, converge faster) but also requires about $1/\mu$ kernel operations per iteration, much more in general than $\mu-$ClipMDM. However, the $\mu-$MDM test accuracies were significantly worse on the *Splice*, *Image*, *Banana* and *Ringnorm* datasets, where $\mu-$MDM convergence seems to stall at suboptimal points.

Table 2
Averages and standard deviations of the test errors (in %) and number of support vectors for the $\mu-$GSK and $\mu-$ClipMDM algorithms.

Set	$\mu-$GSK	$\mu-$ClipMDM	$\mu-$GSK	$\mu-$ClipMDM
T	22.7 ± 0.6	22.7 ± 0.7	135.5 ± 12.3	124.2 ± 5.9
H	15.9 ± 3.2	15.9 ± 3.2	104.8 ± 6.5	91.8 ± 1.5
D	23.7 ± 1.8	23.6 ± 1.8	303.3 ± 7.4	281.4 ± 2.4
BC	27.7 ± 4.7	27.6 ± 4.9	157.1 ± 9.5	121.9 ± 5.0
Th	4.5 ± 2.0	4.5 ± 2.0	42.3 ± 7.4	30.2 ± 5.1
F	32.8 ± 1.6	32.9 ± 1.6	608.1 ± 26.9	529.1 ± 6.8
S	10.7 ± 0.7	10.8 ± 0.6	989.4 ± 46.2	629.2 ± 13.7
I	3.1 ± 0.7	3.2 ± 0.6	363.3 ± 14.2	179.3 ± 8.8
G	23.9 ± 2.0	23.8 ± 2.1	465.4 ± 14.3	414.5 ± 7.7
B	10.7 ± 0.5	10.7 ± 0.5	151.5 ± 7.4	109.3 ± 3.2
Tw	2.9 ± 0.2	2.9 ± 0.2	103.4 ± 5.9	78.5 ± 4.4
R	1.7 ± 0.1	1.7 ± 0.1	400.0 ± 0.0	152.4 ± 11.0
W	9.8 ± 0.4	9.8 ± 0.4	209.6 ± 10.3	173.3 ± 4.6

Table 3
Averages and standard deviations of the number of iterations and kernel operations (the former in thousands and the latter in millions) for the $\mu-$GSK and $\mu-$ClipMDM algorithms.

Set	$\mu-$GSK	$\mu-$ClipMDM	$\mu-$GSK	$\mu-$ClipMDM
T	18.1 ± 19.9	0.09 ± 0.02	184.5 ± 198.6	0.027 ± 0.007
H	0.18 ± 0.09	0.19 ± 0.01	2.0 ± 1.1	0.064 ± 0.004
D	0.17 ± 0.08	0.49 ± 0.02	17.4 ± 7.7	0.46 ± 0.02
BC	1.0 ± 0.7	0.3 ± 0.1	17.6 ± 12.6	0.13 ± 0.04
Th	13.3 ± 4.2	0.26 ± 0.06	14.2 ± 4.5	0.073 ± 0.002
F	2.0 ± 0.9	0.6 ± 0.06	578.3 ± 247.2	0.85 ± 0.09
S	100.0 ± 0.0	2.5 ± 0.2	200.6 ± 0.0	5.1 ± 3.3
I	70.8 ± 7.9	22.2 ± 4.6	4635.3 ± 515.6	57.8 ± 11.9
G	1.0 ± 0.4	0.85 ± 0.08	198.4 ± 76.7	1.2 ± 0.1
B	8.9 ± 12.2	1.9 ± 2.1	205.2 ± 283.0	1.4 ± 1.7
Tw	5.6 ± 1.3	0.5 ± 0.03	63.0 ± 14.6	0.41 ± 0.02
R	100.0 ± 0.0	0.54 ± 0.02	40.2 ± 0.0	0.43 ± 0.02
W	1.1 ± 0.4	0.45 ± 0.02	45.0 ± 14.1	0.36 ± 0.02

6. Discussion and conclusions

The classical GSK and MDM algorithms for the nearest point problem (NPP) have been extended by Mavroforakis et al. and Tao et al., respectively, to $\mu-$NPP, i.e., NPP over $\mu-$reduced convex hulls (RCHs). Their key idea is to work with the extreme points of the RCHs that can be written as concrete convex combinations of the original sample patterns. While solving $\mu-$NPP, these extensions require first to sort the cache of the dot products between the current vector W and the sample patterns as well as a potentially quite high number of kernel operations to update the cache in a kernel setting. On the other hand, the MDM algorithm is closely related to SMO, and linear penalty SVMs for non-linearly separable samples can be shown to be equivalent to $\mu-$NPP over suitable RCHs. This suggests to explore the application to this problem of simple clipped versions of the original GSK and MDM algorithms that would have the obvious advantages of not needing any sorting step and of requiring many less kernel operations for cache updates.

We have done so in this paper, describing how a clipped GSK algorithm may get stuck at non-optimal vectors and, thus, fail to solve $\mu-$NPP; this difficulty may also affect Tao et al.'s MDM extension. However, using the KKT conditions for $\mu-$NPP, we have also shown that clipped MDM is not affected by this problem and provides a simpler and easier to implement procedure. Moreover, we have shown it to have, for similar accuracy values, a better computational performance than that of the other GSK and MDM extensions as it gives sparser models, usually requires less iterations and performs a much lower number of kernel operations. In short, our results show that $\mu-$ClipMDM is the most efficient up-to-date algorithm to solve the NPP problem for $\mu-$RCHs.

On the other hand, we have shown in [14] that the MDM algorithm can be seen as a version of the SMO algorithm in which the two updating patterns must be taken in the same class. This implies that SMO is a more general algorithm than MDM, and thus it is bound to be faster when solving SVM classification, something which is borne out experimentally in [14]. However, devising efficient algorithms to solve $\mu-$NPP has an interest of its own. First, this problem arises naturally in fields such as Robotics or Computational Geometry [15] where an SVM formulation is not appropriate. Moreover, the geometric component inherent to $\mu-$NPP sheds light on the behavior of MDM and, by extension, SMO. For instance, this perspective suggests new ways to accelerate both algorithms; as an example, such an improvement based on zigzagging detection is proposed in [16].

We conclude with some pointers to further research. First we observe that the clipping strategy followed for RCH-NPP might be applicable in other, similar contexts. Bi and Bennett [17] present a framework in which support vector regression [18] can be solved through NPP and a clipping approach could also be useful in that problem. Another question of interest is to prove the convergence of clipped MDM. In [19] the authors have given a unified framework to prove convergence of GSK, MDM and SMO for linearly separable samples. Notice that reduced convex hulls are, in fact, linearly separable and the convergence of the GSK extension of Mavroforakis et al. follows from the results in [19]; they can also be seen as a first step towards proving the convergence of clipped MDM. Moreover, once convergence has been established, another point of interest is the study of convergence rates for clipped MDM. We are currently working on these and other related topics.

Acknowledgments

This research work was carried out at the Instituto de Ingeniería del Conocimiento, Universidad Autónoma de Madrid, and partially supported by Spain's TIN 2007-66862 and by Cátedra UAM-IIC en Modelado y Predicción. Jorge López is a doctoral researcher kindly supported by an FPU (Formación de Profesorado Universitario) Grant from the Spanish Ministry of Education, reference AP2007-00142. Álvaro Barbero is a doctoral researcher kindly supported by another FPU Grant from the Spanish Ministry of Education, reference AP2006-02285.

References

[1] E. Gilbert, Minimizing the quadratic form on a convex set, SIAM Journal on Control 4 (1966) 61–79.
[2] V. Franc, V. Hlavăc, An iterative algorithm learning the maximal margin classifier, Pattern Recognition 36 (2003) 1985–1996.
[3] B. Mitchell, V. Dem'yanov, V. Malozemov, Finding the point of a polyhedron closest to the origin, SIAM Journal on Control 12 (1974) 19–26.
[4] S.S. Keerthi, S.K. Shevade, C. Bhattacharyya, K.R.K. Murthy, A fast iterative nearest point algorithm for support vector machine classifier design, IEEE Transactions on Neural Networks 11 (1) (2000) 124–136.
[5] M. Mavroforakis, S. Theodoridis, A geometric approach to support vector machine (SVM) classification, IEEE Transactions on Neural Networks 17 (3) (2006) 671–682.
[6] Q. Tao, G.W. Wu, J. Wang, A generalized S–K algorithm for learning $\nu-$SVM classifiers, Pattern Recognition Letters 25 (10) (2004) 1165–1171.
[7] Q. Tao, G.W. Wu, J. Wang, A general soft method for learning SVM classifiers with L1-norm penalty, Pattern Recognition 41 (2008) 939–948.
[8] K. Bennett, E. Bredensteiner, Duality and geometry in SVM classifiers, in: Proceedings of the 17th International Conference on Machine Learning, 2000, pp. 57–64.
[9] L. González, C. Angulo, F. Velasco, A. Català, Dual unification of bi-class support vector machine formulations, Pattern Recognition 39 (7) (2006) 1325–1332.
[10] L. González, C. Angulo, F. Velasco, A. Català, Unified dual for bi-class SVM approaches, Pattern Recognition 38 (10) (2005) 1772–1774.
[11] V. Franc, V. Hlavăc, Simple solvers for large quadratic programming tasks, in: Proceedings of the 27th DAGM Symposium, Springer, Vienna, Austria, 2005, pp. 75–84.
[12] M. Mavroforakis, M. Sdralis, S. Theodoridis, A geometric nearest point algorithm for the efficient solution of the SVM classification task, IEEE Transactions on Neural Networks 18 (5) (2007) 1545–1549.
[13] G. Rätsch, Benchmark repository, datasets available at 〈http://ida.first.fhg.de/projects/bench/benchmarks.htm〉, 2000.
[14] J. López, Á. Barbero, J.R. Dorronsoro, On the equivalence of the SMO and MDM algorithms for SVM training, in: Lecture Notes in Artificial Intelligence: Machine Learning and Knowledge Discovery in Databases-ECML 2008 Proceedings, Part II, Springer, 2008.
[15] N.K. Sancheti, S.S. Keerthi, Computation of certain measures of proximity between convex polytopes: a complexity viewpoint, in: Proceedings of the 1992 IEEE International Conference on Robotics and Automation, 1992.
[16] Á. Barbero, J. López, J.R. Dorronsoro, Cycle-breaking acceleration of SVM training, Neurocomputing 72 (7–9) (2009) 1398–1406.
[17] J. Bi, K. Bennett, A geometric approach to support vector regression, Neurocomputing 55 (2003) 79–108.
[18] A.J. Smola, B. Schölkopf, A tutorial on support vector regression, Statistics and Computing 48 (2003) 199–222.
[19] J. López, J.R. Dorronsoro, A common framework for the convergence of the GSK, MDM and SMO algorithms, in: Lecture Notes in Computer Science, vol. 6353, Springer, 2010, pp. 82–87. doi:10.1007/978-3-642-15822-3_10.

Jorge López received his degree in Computer Engineering from the Universidad Autónoma de Madrid in 2006, where he got an honorific mention as the best student, and his MSc in Computer Science and Telecommunications in 2008 from the same university. Currently he is a doctoral researcher in its Computer Science Department, in collaboration with the Instituto de Ingeniería del Conocimiento. His research interests concentrate on support vector machines, but also cover additional machine learning and pattern recognition paradigms.

Álvaro Barbero received his degree in Computer Engineering from the Universidad Autónoma de Madrid in 2006 and his MSc in Computer Science and Telecommunications in 2008 from the same university. Currently he is a doctoral researcher in its Computer Science Department, in collaboration with the Instituto de Ingeniería del Conocimiento. His research interests concentrate on pattern recognition, kernel methods and wind power forecasting.

José R. Dorronsoro received his degree in Mathematics from the Universidad Complutense de Madrid in 1977 and his PhD degree in Mathematics from Washington University in St. Louis in 1981. Currently he is a full professor in the Computer Engineering Department of the Universidad Autonoma de Madrid, of which he was head from 1993 to 1996. His research interests concentrate on neural networks, image processing and pattern recognition.