On the Equivalence of the SMO and MDM Algorithms for SVM Training

Jorge López, Álvaro Barbero, and José R. Dorronsoro*

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

Abstract. SVM training is usually discussed under two different algorithmic points of view. The first one is provided by decomposition methods such as SMO and SVMLight while the second one encompasses geometric methods that try to solve a Nearest Point Problem (NPP), the Gilbert–Schlesinger–Kozinec (GSK) and Mitchell–Demyanov–Malozemov (MDM) algorithms being the most representative ones. In this work we will show that, indeed, both approaches are essentially coincident. More precisely, we will show that a slight modification of SMO in which at each iteration both updating multipliers correspond to patterns in the same class solves NPP and, moreover, that this modification coincides with an extended MDM algorithm. Besides this, we also propose a new way to apply the MDM algorithm for NPP problems over reduced convex hulls.

1 Introduction

Given a sample $\mathcal{S} = \{(X_i, y_i) : i = 1, \ldots, N\}$ with $y_i = \pm 1$, the standard formulation of SVM for linearly separable problems seeks [1,2] to maximize the margin of a separating hyperplane by solving the problem

$$\min \frac{1}{2}\|W\|^2 \;\text{ with }\; y_i(W \cdot X_i + b) \geq 1, i = 1, \ldots, N. \tag{1}$$

Any pair (W, b) verifying the restrictions in (1) is said to be in canonical form. In practice, however, the problem actually solved is the simpler dual problem of minimizing

$$W(\alpha) = \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j X_i \cdot X_j - \sum_i \alpha_i \;\text{ with }\; \alpha_i \geq 0, \;\sum_i \alpha_i y_i = 0. \tag{2}$$

The optimal weight W^o can be then written as $W^o = \sum \alpha_i^o y_i X_i$ and patterns for which $\alpha_i^o > 0$ are called support vectors (SV). There are quite a few proposals of algorithms to solve (2); many of them can be broadly classified into two categories that usually are discussed as independent procedures, decomposition

* All authors have been partially supported by Spain's TIN 2007–66862. The second author is kindly supported by the FPU–MEC grant reference AP2006–02285.

W. Daelemans et al. (Eds.): ECML PKDD 2008, Part I, LNAI 5211, pp. 288–300, 2008.

algorithms and geometrically inspired methods. Many decomposition algorithms can be traced to Platt's SMO [3] or Joachims's SVM–Light [4] algorithms. SMO, one of the most popular methods, proceeds iteratively, working at each step with a reduced set of only two multipliers, α_{i_1}, α_{i_2} and solving problem (2) exactly for them while keeping fixed all others. To stop training, SMO looks at the KKT conditions for the dual of (2). At the optimal $W^o = \sum \alpha_i^o y_i X_i$, they imply $\alpha_i^o y_i (W^o \cdot X_i + b^o - y_i) = 0$ and, thus, we have

$$\begin{aligned} \alpha_i^o > 0 &\Rightarrow y_i(W^o \cdot X_i + b^o - y_i) = 0, \\ \alpha_i^o = 0 &\Rightarrow y_i(W^o \cdot X_i + b^o - y_i) \geq 0. \end{aligned} \tag{3}$$

Hence, during training there might be two kinds of violations of these KKT conditions. The first one happens when $\alpha_i > 0$ but $y_i(W \cdot X_i + b - y_i) \neq 0$. The second one takes place if $\alpha_i = 0$ but $y_i(W \cdot X_i + b - y_i) < 0$. Platt's SMO algorithm essentially tries to choose i_2 as the index of the pattern X_i that somehow most violates these conditions for the current W and i_1 as the index that gives then a maximum decrease in $W(\alpha)$. However, and as pointed out in [5], this may lead to some difficulties as the KKT conditions only hold approximately during training. To avoid this Keerthi et al. propose in [5] two modifications to SMO and recommend the second one, Modification 2, as the most effective (see also [6], where it is shown to be equivalent to 2–vector SVM–Light); we will briefly describe it in section 2.

Turning our attention to geometric algorithms, they are usually motivated through another way of setting up SVM training, the Nearest Point Problem (NPP; see [7]) in which we want to find the nearest points W_+^* and W_-^* of the convex hulls $C(\mathcal{S}_\pm)$ of the positive $\mathcal{S}_+ = \{X_i : y_i = 1\}$ and negative $\mathcal{S}_- = \{X_i : y_i = -1\}$ sample subsets. The maximum margin hyperplane is then $W^* = W_+^* - W_-^*$ and the optimal margin is given by $\|W^*\|/2$. If we write a $W_+ \in C(\mathcal{S}_+)$ as $W_+ = \sum \alpha_p X_p$, with $\sum \alpha_p = 1$ and a $W_- \in C(\mathcal{S}_-)$ as $W_- = \sum \alpha_q X_q$, with $\sum \alpha_q = 1$ we have $W = W_+ - W_- = \sum \alpha_i y_i X_i$ with $X_i \in \mathcal{S} = \mathcal{S}_+ \bigcup \mathcal{S}_-$. We can thus state the NPP problem as follows:

$$\min \frac{1}{2}\|W\|^2 = \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j X_i \cdot X_j, \quad \text{with} \quad \alpha_i \geq 0, \sum_i \alpha_i y_i = 0, \sum_i \alpha_i = 2, \tag{4}$$

where we assume again a linearly separable training sample. In [8,9] specific algorithms have been proposed for NPP that originate in the more classical Gilbert–Schlesinger–Kozinec (GSK; [10,11]) and Mitchell–Demyanov–Malozemov (MDM; [12]) algorithms to find the minimum norm vector of a convex set. While the GSK algorithm can be very slow, the MDM algorithm and some improvements (see [8]) are quite efficient.

While, as mentioned before, decomposition and geometric algorithms are usually discussed as independent procedures, we shall give in section 2 a new derivation of the MDM algorithm and show that for linearly separable problems, it

essentially coincides with a slight variant of SMO in which we require that both updating vectors belong to the same class. Although SVM algorithms for linearly separable problems extend immediately to non separable ones if square penalties $C\sum \xi_i^2$ are applied to margin slacks ξ_i [13], a different set up has to be pursued if linear penalties $C\sum \xi_i$ are considered. For SVM training this implies a restriction $\alpha_i \leq C$ for the multipliers α_i while NPP has to be solved over the so–called μ–Reduced Convex Hulls, where an extra restriction $\alpha_i \leq \mu$ has to be added to those in (4). It is well known that both problems are equivalent [7], but in the Appendix we will give a new, short proof of this fact. In section 3 we will extend to these settings the equivalence between SMO and MDM already proved for linearly separable problems in section 2. We will briefly compare numerically the performance of basic versions of the SMO and MDM algorithms in section 4 and show that, for square penalties, the final models they arrive at are essentially the same, as they have similar test accuracies and numbers of support vectors. SMO, however, needs less iterations than MDM, something to be expected, as it has to meet less restrictions when iteratively looking for maximum gains. The comparison for linear penalties is somewhat more involved, but the faster convergence of SMO still holds. A brief discussion ends the paper.

2 The SMO–MDM Equivalence for Linearly Separable Problems

2.1 Keerthi et al.'s Modification 2

Writing $F_i^o = W^o \cdot X_i - y_i$, the KKT conditions (3) at the optimal W^o, b^o can be expressed as

$$y_i(F_i^o + b^o) = 0 \ \text{ if } \ \alpha_i^o > 0, \ \ y_i(F_i^o + b^o) \geq 0 \ \text{ if } \ \alpha_i^o = 0. \tag{5}$$

Thus, if we define first the index sets $I^+ = \{i : y_i = 1\}$, $I^- = \{i : y_i = -1\}$ and then $I_{nSV} = \{i : \alpha_i = 0\}$, $I_{SV} = \{i : \alpha_i > 0\}$ (I_0 in the notation of [5]), $I_{nSV}^+ = I^+ \bigcap I_{nSV}$ (I_1 in Keerthi's notation), $I_{nSV}^- = I^- \bigcap I_{nSV}$ (I_4 in Keerthi's notation), the preceding conditions can be written as

$$F_i^o + b^o \geq 0 \ \text{ for } \ i \in I_{SV} \bigcup I_{nSV}^+, \ \ F_i^o + b^o \leq 0 \ \text{ for } \ i \in I_{SV} \bigcup I_{nSV}^-.$$

In particular, we will have $F_i^o \geq -b^o$ for $i \in I_{SV} \bigcup I_{nSV}^+$ and $-b^o \geq F_j^o$ for $j \in I_{SV} \bigcup I_{nSV}^-$. Thus, if we write $F_i = W \cdot X_i - y_i$ and define

$$b_{low} = \max\{F_j : j \in I_{SV} \bigcup I_{nSV}^-\}, \quad b_{up} = \min\{F_i : i \in I_{SV} \bigcup I_{nSV}^+\},$$

we must have $b_{low} \leq -b^o \leq b_{up}$ at the optimum. In practice one has to relax these conditions to $b_{low} - \epsilon/2 \leq -b^o \leq b_{up} + \epsilon/2$ for some $\epsilon > 0$. These observations

motivate Keerthi et al.'s Modification 2 in [5]. More precisely, they define at each step two indices

$$\begin{aligned} i_{low} &= \arg\max \ \{F_j : j \in I_{SV} \bigcup I^-_{nSV}\}, \\ i_{up} &= \arg\min \ \{F_i : i \in I_{SV} \bigcup I^+_{nSV}\}, \end{aligned} \tag{6}$$

and propose to take $i_2 = i_{low}$ and $i_1 = i_{up}$ in SMO. We then have $b_{low} = F_{i_{low}}$, $b_{up} = F_{i_{up}}$ and training will continue while $b_{low} > b_{up} + \epsilon$ or, in other words, while the i_2, i_1 indices violate the KKT conditions. As the experiments reported in [5] illustrate, these choices can significantly speed up Platt's original algorithm.

2.2 An Alternative Motivation for Choosing i_2 and i_1

Keerthi's heuristics are motivated by an attempt to simplify Platt's original ones but we will show next how they also arise if we try to choose directly the updating indices i_2, i_1 so that they maximize the gain in the dual cost function $W(\alpha)$ (see also the Appendix A in [6] for another way to arrive at these selections). Notice first that for any such pair (i_2, i_1) the new multipliers α' to be considered are $\alpha'_{i_1} = \alpha_{i_1} + \delta_{i_1}$, $\alpha'_{i_2} = \alpha_{i_2} + \delta_{i_2}$ while $\alpha'_j = \alpha_j$ for all others. The new W' has thus the form $W' = W + \delta_{i_1} y_{i_1} X_{i_1} + \delta_{i_2} y_{i_2} X_{i_2}$. Taking into account the restriction $\sum_i \alpha_i y_i = 0$, we must have $y_{i_1}\delta_{i_1} + y_{i_2}\delta_{i_2} = 0$ and, therefore, $\delta_{i_1} = -y_{i_1} y_{i_2} \delta_{i_2}$ and

$$W' = W + \delta_{i_2} y_{i_2} (X_{i_2} - X_{i_1}) = W + \delta_{i_2} y_{i_2} Z_{i_2,i_1},$$

where $Z_{j,k} = X_j - X_k$. Thus, $W(\alpha') = \frac{1}{2}\|W'\|^2 - \sum \alpha'_i$ is just a function $\Phi(\delta_{i_2})$ of δ_{i_2}, and we have

$$\begin{aligned} \Phi(\delta_{i_2}) &= \frac{1}{2}\|W\|^2 + \delta_{i_2} y_{i_2} W \cdot Z_{i_2,i_1} + \frac{\delta^2_{i_2}}{2}\|Z_{i_2,i_1}\|^2 - \sum \alpha_i - \delta_{i_1} - \delta_{i_2} \\ &= W(\alpha) + \delta_{i_2} y_{i_2} W \cdot Z_{i_2,i_1} + \frac{\delta^2_{i_2}}{2}\|Z_{i_2,i_1}\|^2 - \delta_{i_2} y^2_{i_2} + y_{i_1} y_{i_2} \delta_{i_2} \\ &= W(\alpha) + \delta_{i_2} y_{i_2} \left(W \cdot Z_{i_2,i_1} - (y_{i_2} - y_{i_1})\right) + \frac{\delta^2_{i_2}}{2}\|Z_{i_2,i_1}\|^2. \end{aligned} \tag{7}$$

Solving $\Phi'(\delta^*_{i_2}) = 0$ to obtain the optimal $\delta^*_{i_2}$ yields

$$\delta^*_{i_2} = -\frac{y_{i_2}\left(W \cdot Z_{i_2,i_1} - (y_{i_2} - y_{i_1})\right)}{\|Z_{i_2,i_1}\|^2} = -y_{i_2}\frac{\Delta}{\|Z_{i_2,i_1}\|^2}, \tag{8}$$

where $\Delta = W \cdot Z_{i_2,i_1} - (y_{i_2} - y_{i_1})$, and, in turn, $\delta^*_{i_1} = -y_{i_1} y_{i_2} \delta^*_{i_2} = y_{i_1}\frac{\Delta}{\|Z_{i_2,i_1}\|^2}$. Moreover, we have

$$\Phi(\delta^*_{i_2}) = W(\alpha) - \frac{1}{2}\frac{\left[y_{i_2}\left(W \cdot Z_{i_2,i_1} - (y_{i_2} - y_{i_1})\right)\right]^2}{\|Z_{i_2,i_1}\|^2} = W(\alpha) - \frac{1}{2}\frac{\Delta^2}{\|Z_{i_2,i_1}\|^2}.$$

Now, to maximize the decrease in $W(\alpha')$ we should choose (i_2, i_1) so that

$$(i_2, i_1) = \arg\max_{i,j} \left\{ \frac{(W \cdot Z_{i,j} - (y_i - y_j))^2}{\|Z_{i,j}\|^2} \right\}.$$

Such a choice of i_2, i_1 is sometimes called a second order working set selection [14]. If we simply ignore the $\|Z_{i,j}\|^2$ denominator, we can choose instead

$$(i_2, i_1) = \arg\max_{i,j} \{|W \cdot Z_{i,j} - (y_i - y_j)|\}. \tag{9}$$

It is clear that the maximum in (9) is attained at

$$\max_i \{W \cdot X_i - y_i\} - \min_j \{W \cdot X_j - y_j\},$$

which tells us to choose in principle (i_2, i_1) as

$$i_2 = \arg\max_j \{W \cdot X_j - y_j\}, \;\; i_1 = \arg\min_i \{W \cdot X_i - y_i\}. \tag{10}$$

These choices imply $\Delta \geq 0$ and we note in passing that there is a gain in $W(\alpha)$ whenever $\Delta > 0$ or, stated equivalently, whenever there is a violating pair; this gives a new and simple derivation of a well known result of Hush and Scovel (see Theorem 3 in [15]). Now, notice that if $y_{i_2} = 1$, $\delta_{i_2} < 0$ and, hence, we must have $\alpha_{i_2} > 0$. On the other hand, if $y_{i_1} = -1$, $\delta_{i_1} < 0$ and, hence, we must have $\alpha_{i_1} > 0$. As a consequence, we must refine our previous choices of i_2 and i_1 in (10) to

$$i_1 = \arg\min_i \{F_i : i \in I^+ \bigcup I^-_{SV}\}, \;\; i_2 = \arg\max_j \{F_i : i \in I^- \bigcup I^+_{SV}\}. \tag{11}$$

with $I^{\pm}_{SV} = I^{\pm} \bigcap I_{SV}$ and $F_i = W \cdot X_j - y_j$ again. Now it can be easily seen that $I^+ \bigcup I^-_{SV} = I_{SV} \bigcup I^+_{nSV}$ and, similarly, $I^- \bigcup I^+_{SV} = I_{SV} \bigcup I^-_{nSV}$. It is thus clear that these are the same selections done in Modification 2 of [5] as given in (6).

2.3 Solving NPP a la SMO

As discussed in section 1 there are several procedures for the NPP problem that have their origin in the MDM algorithm. In its original formulation as a minimum norm problem, the MDM algorithm selects at each step updating indices $i_2 = \arg\min_j \{W \cdot X_j\}$, $i_1 = \arg\max_i \{W \cdot X_i : \alpha_i > 0\}$. While the algorithm's objective is to update the current weight W with the one in the line segment between W and $W + \alpha_{i_2}(X_{i_2} - X_{i_1})$ with minimum norm, it is clear that the i_2 and i_1 choices also maximize $\Delta^2 = (W \cdot (X_i - X_j))^2$ (the condition $\alpha_i > 0$ for i_1 candidates is needed, as the W update will decrease α_{i_1}). While the approach in [8] to NPP is closer to the original MDM one as given in [12], the one in [9] does in fact try to maximize Δ^2.

In any case the above index choices are clearly related to the previous discussion for SMO and their minimization of Δ suggests to solve NPP as just done in the preceding section, that is, to work at each step with just two multipliers α_{i_1} and α_{i_2} and update a given $W = \sum \alpha_i y_i X_i$ to another one of the form $W' = W + \delta_{i_1} y_{i_1} X_{i_1} + \delta_{i_2} y_{i_2} X_{i_2}$ so that the minimization in the norm $\|W'\|^2$ is largest. The restrictions in (4) imply $2 = \sum \alpha'_i = \sum \alpha_i + \delta_{i_1} + \delta_{i_2} = 2 + \delta_{i_1} + \delta_{i_2}$ and $0 = \sum y_i \alpha'_i = \sum y_i \alpha_i + y_{i_1}\delta_{i_1} + y_{i_2}\delta_{i_2} = y_{i_1}\delta_{i_1} + y_{i_2}\delta_{i_2}$. The second one implies that $y_{i_1}\delta_{i_1} = -y_{i_2}\delta_{i_2}$ and, since the first one gives $\delta_{i_1} = -\delta_{i_2}$, we must also have $y_{i_1} = y_{i_2}$. As a consequence, $W' = W + \delta_{i_2} y_{i_2} (X_{i_2} - X_{i_1}) = W + \delta_{i_2} y_{i_2} Z_{i_2,i_1}$, where again $Z_{i,j} = X_i - X_j$; thus, $\|W'\|^2$ is a function of δ_{i_2} and we have

$$\Phi(\delta_{i_2}) = \|W'\|^2 = \|W\|^2 + 2\delta_{i_2} y_{i_2} W \cdot Z_{i_2,i_1} + \delta_{i_2}^2 \|Z_{i_2,i_1}\|^2.$$

As done before, solving $\Phi'(\delta^*_{i_2}) = 0$ gives

$$\delta^*_{i_2} = -y_{i_2} \frac{\Delta}{\|Z_{i_2,i_1}\|^2}, \quad \delta^*_{i_1} = y_{i_2} \frac{\Delta}{\|Z_{i_2,i_1}\|^2},$$

where now $\Delta = W \cdot Z_{i_2,i_1}$ and, in turn,

$$\Phi(\delta^*_{i_2}) = \|W\|^2 - \frac{\Delta^2}{\|Z_{i_2,i_1}\|^2}. \tag{12}$$

Thus, just as before, if we ignore the $\|Z_{i_2,i_1}\|^2$ denominator, we can maximize the gain in Φ by selecting i_1 and i_2 so that Δ is maximized. We do so setting first

$$\begin{aligned} i_2^+ &= \arg\max_i \{W \cdot X_i : y_i = 1\}, \quad i_1^+ = \arg\min_j \{W \cdot X_j : y_j = 1\}, \\ i_2^- &= \arg\max_i \{W \cdot X_i : y_i = -1\}, \quad i_1^- = \arg\min_j \{W \cdot X_j : y_j = -1\}, \end{aligned} \tag{13}$$

and deciding next which one of the pairs $(i_2^\pm, i_1^\pm)$ to choose, for which we compute

$$\Delta^+ = W \cdot \left(X_{i_2^+} - X_{i_1^+}\right), \quad \Delta^- = W \cdot \left(X_{i_2^-} - X_{i_1^-}\right),$$

(notice that both are positive) and take $i_2 = i_2^+$, $i_1 = i_1^+$ if $\Delta^+ > \Delta^-$ and $i_2 = i_2^-$, $i_1 = i_1^-$ otherwise. We observe that the corresponding index choices in the extension of MDM to NPP are

$$\begin{aligned} i_2^+ &= \arg\max_i \{W \cdot (X_i - W_-) : y_i = 1\}, \\ i_1^+ &= \arg\min_j \{W \cdot (X_j - W_-) : y_j = 1\}, \\ i_2^- &= \arg\max_i \{W \cdot (X_i - W_+) : y_i = -1\}, \\ i_1^- &= \arg\min_j \{W \cdot (X_j - W_+) : y_j = -1\}, \end{aligned}$$

which are obviously equivalent to the previous ones.

In any case, and just as it was done for SMO, we must make sure that the updated coefficients remain positive. Just as before we have $\Delta^\pm > 0$. Thus, if $y_{i_2} = 1$, $\delta_{i_2^+} < 0$ and, hence, we must have $\alpha_{i_2^+} > 0$. On the other hand, if

$y_{i_1} = -1$, $\delta_{i_1^-} < 0$ and, hence, we must have $\alpha_{i_1^-} > 0$. As a consequence, we refine our previous choices of i_2^+ and i_1^- in (13) to

$$i_1^- = \arg\min_i\{W \cdot X_i : i \in I_{SV}^-\}, \;\; i_2^+ = \arg\max_j\{W \cdot X_i : i \in I_{SV}^+\}. \quad (14)$$

As we show next, these choices coincide with those made in a slight variant of SMO.

2.4 Enforcing $y_{i_1} = y_{i_2}$ in SMO

Although in standard SMO the y_{i_1} and y_{i_2} values do not have to be equal, let us discuss SMO's formulation when at each iteration we force $y_{i_1} = y_{i_2}$ (the use of updates where all patterns used belong to the same class has also been proposed for ν–SV training [16]). We then have $\delta_{i_1} = -y_{i_1}y_{i_2}\delta_{i_2} = -\delta_{i_2}$ and $W' = W + \delta_{i_2}y_{i_2}X_{i_2} - \delta_{i_2}y_{i_2}X_{i_1} = W + \delta_{i_2}y_{i_2}Z_{i_2,i_1}$. Furthermore, (7) becomes now

$$\Phi(\delta_{i_2}) = W(\alpha') = W(\alpha) + y_{i_2}W \cdot Z_{i_2,i_1}\delta_{i_2} + \frac{\delta_{i_2}^2}{2}\|Z_{i_2,i_1}\|^2, \quad (15)$$

equation (8) for the optimum $\delta_{i_2}^*$ becomes

$$\delta_{i_2}^* = -y_{i_2}\frac{W \cdot Z_{i_2,i_1}}{\|Z_{i_2,i_1}\|^2} = -y_{i_2}\frac{\Delta}{\|Z_{i_2,i_1}\|^2},$$

where here $\Delta = W \cdot Z_{i_2,i_1}$, and, again, we have

$$\Phi(\delta_{i_2}^*) = W(\alpha) - \frac{1}{2}\frac{\Delta^2}{\|Z_{i_2,i_1}\|^2},$$

which has the same form that (12). Ignoring once more the denominator $\|Z_{i_2,i_1}\|^2$, this also suggests to take i_2, i_1 so as to maximize $|\Delta|$, which leads to the same index choices as done for MDM in the previous section.

Moreover, enforcing $y_{i_2} = y_{i_1}$ implies that $\delta_{i_1} = -\delta_{i_2}$ and also that, after initialization, the multipliers' sum $\sum \alpha_i$ remains constant at each iteration. Thus, in this setting, minimizing the dual criterion $W(\alpha) = \|W\|^2/2 - \sum \alpha_i$ reduces to minimize just $\|W\|^2$ and if the α_i are initialized so that $\sum \alpha_i = 2$, the problem that this SMO variant solves coincides with NPP. Moreover, since the updating indices' choices are the same in both cases, we can conclude that after a proper initialization, enforcing $y_{i_2} = y_{i_1}$ in SMO is equivalent to using MDM to solve NPP.

3 SMO and MDM for Non–linearly Separable Problems

In the preceding discussion we have assumed that the original sample classes were linearly separable. This assumption must be relaxed in practice allowing

for margin slacks that are penalized using either a linear or a quadratic cost function. The theory for the linearly separable case extends easily to the quadratic cost setting [13], but for a linear penalty we want to solve now the quadratic minimization problem

$$\min_{W,b,\xi} \frac{1}{2}\|W\|^2 + C\sum_i \xi_i, \tag{16}$$

subject to the linear restrictions $y_i(W \cdot X_i + b) + \xi_i \geq 1, i = 1, \ldots, N$. Its Wolfe dual is now

$$W(\alpha) = \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j X_i \cdot X_j - \sum_i \alpha_i, \tag{17}$$

where $0 \leq \alpha_i \leq C$, $\sum_i \alpha_i y_i = 0$. If NPP is to be considered, the alternative to (16) would be to consider it for the so–called μ–Reduced Convex Hulls, defined as $C_\mu(\mathcal{S}_\pm) = \{\sum \alpha_i X_i : X_i \in \mathcal{S}_\pm, \sum \alpha_i = 1, 0 \leq \alpha_i \leq \mu\}$. We shall refer to this new problem as RCH$_\mu$–NPP (see [7] for more details).

Considering first SMO, the only difference with respect the discussion in section 2 is the restriction $\alpha_i \leq C$, which forces the δ_i increments to be positive only when $\alpha_i < C$. Thus, if $y_{i_2} = -1$ we must have $\alpha_{i_2} < C$ and if $y_{i_1} = 1$ we must have $\alpha_{i_1} < C$. As a consequence, in the non–linearly separable setting we must refine the i_2 and i_1 choices in (11) to

$$i_1 = \arg\min_i\{F_i : i \in I^+_{nB_C} \bigcup I^-_{SV}\}, \;\; i_2 = \arg\max_j\{F_i : i \in I^-_{nB_C} \bigcup I^+_{SV}\} \tag{18}$$

where now $I^\pm_{nB_C} = \{i : y_i = \pm 1, \alpha_i < C\}$. It can be easily checked that these are the same selections done in Modification 2 of [5]. Turning our attention to the MDM algorithm for RCH$_\mu$–NPP, the situation is quite similar to the one just discussed for SMO, as we have to make sure that when $\alpha = \mu$, decrementing α is then the only option. As a consequence, we must now refine our previous choices of i_2^- and i_1^+ in (13) to

$$i_1^+ = \arg\min_i\{W \cdot X_i : i \in I^+_{nB_\mu}\}, \;\; i_2^- = \arg\max_j\{W \cdot X_i : i \in I^-_{nB_\mu}\}, \tag{19}$$

where now $I^\pm_{nB_\mu} = \{i : y_i = \pm 1, \alpha_i < \mu\}$. Arguing as before, initializing the α_i and scaling C adequately, enforcing $y_{i_1} = y_{i_2}$ results in SMO solving RCH$_\mu$–NPP. We finally note that MDM–type algorithms for RCH$_\mu$–NPP have been recently proposed [17] but they are conceptually more involved and computationally costlier than our just explained proposal.

4 Numerical Experiments

We shall compare the performance of the most basic versions of the SMO and NPP algorithms over 10 of the datasets provided in G. Rätsch's Benchmark Repository [18]. We employed the same experimental set–up described in the data site; in particular we used the provided 100 partitions (with about 40% training and 60% test patterns) to compute the test accuracies and the number of final SVs and training iterations, as well as the corresponding standard

Table 1. Average test accuracies, number of support vectors and number of iterations given by the CH-MDM and 2-SMO algorithms, with $\epsilon = 10^{-8}$

	Test err.		# SVs		# iters.	
Dataset	SMO	MDM	SMO	MDM	SMO	MDM
Titanic	22.8±1.2	22.8±1.2	150.0±0.0	150.0±0.0	363.1±20.2	402.1±16.7
Heart	15.7±3.2	15.7±3.2	163.3±2.4	163.3±2.4	338.9±13.0	399.1±14.9
Diabetes	23.1±1.6	23.1±1.6	412.7±7.7	412.7±7.7	1565.7±45.1	1666.1±38.2
Cancer	26.5±4.8	26.5±4.8	179.3±5.9	179.3±5.9	1140.6±52.0	1226.2±54.3
Thyroid	4.3±1.9	4.3±1.9	87.4±3.0	87.4±3.0	226.7±10.3	243.6±9.9
Flare	33.5±1.7	33.5±1.7	664.5±0.7	664.5±0.7	1398.4±53.1	1652.6±51.2
Splice	10.6±0.7	10.6±0.7	728.6±12.7	728.7±12.8	4402.3±635.8	4835.6±667.7
Image	2.9±0.5	2.9±0.5	215.5±11.5	215.3±11.5	34447.9±2117.9	39560.6±3203.9
German	23.56±2.0	23.5±2.0	590.22±12.4	590.0±12.4	19099.1±971.7	20441.6±711.3
Banana	10.4±0.4	10.4±0.4	230.9±14.0	230.9±14.0	1313.0±83.1	1364.6±91.3

deviations. Before giving the concrete results, we briefly comment on some implementation details. First, and as usual, all algorithms only involve dot products, that can be replaced through an appropriate positive definite kernel K. Next, we notice that many improvements have been made to the basic SMO and MDM algorithms, such as Platt's type I and type II updates or support vector shrinking. We will not consider them in our experiments as they are more or less applicable to both procedures and likely to have similar effects. We also point out that we must make sure that, for instance, $0 \leq \alpha_i + \delta_i \leq C$ for linear penalties' SMO and that $0 \leq \alpha_i + \delta_i \leq \mu$ for RCH_μ–NPP. This means that the δ_i will have to be adequately bounded from above and below as necessary. Finally, the b^o and b^* bias values are also different in SMO and MDM. For SMO we will take, as usual,

$$b^o = \frac{1}{N_{SV}} \sum_{i \in I_{SV}} (y_i - W^o \cdot X_i) = \frac{1}{N_{SV}} \left(\sum_{i \in I_{SV}} y_i - \sum_{i,j \in I_{SV}} \alpha_j y_j X_j \cdot X_i \right)$$
$$= \frac{1}{N_{SV}} \left(\sum_{i \in I_{SV}} y_i - \sum_{i,j \in I_{SV}} \alpha_j y_j K'(x_j, x_i) \right),$$

with N_{SV} the number of support vectors. For quadratic penalties we will use $K'(x_j, x_i) = K(x_j, x_i) + \delta_{ij}/C$ as the square penalty–adjusted version of a standard positive definite kernel K while we just take $K' = K$ for linear penalties. A simple geometric reasoning implies that the MDM bias will be

$$b^* = -W^* \cdot \frac{\left(W_+^* + W_-^*\right)}{2} = -\frac{1}{2} \sum_{i,j \in I_{SV}} \alpha_i \alpha_j y_i K'(x_i, x_j).$$

Table 2. Average test accuracies, number of support vectors and number of iterations given by the RCH-MDM and 1-SMO algorithms, with $\epsilon = 10^{-8}$. For test errors a * stands for a statistically significant difference in a Wilcoxon rank test.

Dataset	Test err.		# SVs		# iters.	
	SMO	MDM	SMO	MDM	SMO	MDM
Titanic	24.1±8.0	24.0±7.4	67.2±11.3	113.9±8.8	156.6±32.7	164.2±28.7
Heart	15.8±3.2	16.0±3.1	82.4±5.4	82.4±5.4	217.1±63.6	306.2±59.6
Diabetes	23.4±1.6*	23.7±1.8	264.9±7.2	264.8±7.2	464.7±91.5	741.4±81.9
Cancer	27.3±5.9*	28.9±4.8	113.6±6.5	113.8±6.3	1705.3±897.4	3352.7±3841.2
Thyroid	4.4±2.1	4.2±2.0	25.3±5.7	25.3±5.7	328.2±124.4	398.9±117.1
Flare	32.7±1.6*	32.8±1.6	477.1±12.2	508.9±9.9	862.2±391.0	1401.4±881.5
Splice	10.7±0.6	10.8±0.6	620.2±14.2	629.2±13.6	2569.6±177.4	2797.4±290.5
Image	3.0±0.4	3.0±0.5	167.6±9.2	172.0±8.8	47972.5±11219.0	56169.9±10309.4
German	23.62±2.1*	24.0±2.1	407.6±10.7	407.7±10.8	1660.6±149.2	1884.5±144.8
Banana	11.5±0.6*	11.6±0.6	89.6±10.1	89.5±10.0	38236.0±14307.7	43449.9±25339.9

4.1 Quadratic Penalties

It is well known that SVM algorithms for linearly separable problems extends immediately to non separable ones if square penalties $C\sum \xi_i^2$ are applied to margin slacks ξ_i [13]. We shall use a common initialization for both SMO and MDM choosing a single vector from each class and setting $\alpha_{i_1} = \alpha_{i_2} = 1$. As mentioned in section 2, the usual SMO stopping condition is $b_{low} \leq b_{up} + \epsilon$; for the MDM algorithm one might use either $\Delta \leq \epsilon$ or also $\Delta \leq \epsilon\|W\|^2/2$. While these conditions look similar, the norms of the SMO and MDM W vectors involved are very different. Thus, in order to make more homogeneous performance comparisons, we will use in both cases a similar relative precision criterion, stopping SMO when $W(\alpha) - W(\alpha') \leq \epsilon W(\alpha)$ and MDM when $\|W\|^2 - \|W'\|^2 \leq \epsilon\|W\|^2$.

We will compare the performance of the basic SMO and MDM implementation over three values: the number of training iterations they need, the number of support vectors the final SVMs have and the test accuracies of the final models. We will do so for a relative $\epsilon = 10^{-8}$ precision and the results of each method are shown in table 1. In all cases we have used Gaussian kernels $\exp\left(-\|x\|^2/2\sigma^2\right)$ and optimal σ and C have been estimated by cross–validation. It can be seen in the table that SMO is faster, as it needs less iterations to achieve the desired precisions. This is quite natural, as it has greater freedom when choosing at each iteration the maximum gain multipliers. On the other hand, the final models obtained seem to be very similar, as they essentially have the same accuracies and support vector numbers; moreover, after the appropriate scaling, the corresponding optimal dual function values were essentially the same. A * superscript for the test errors indicates a significant difference in a Wilcoxon rank test at the 10% level; the final test error values are similar to those in [18].

4.2 Linear Penalties

While for square penalties SMO and MDM use the same C parameter, the situation for linear penalty SMO and RCH–MDM is more complex. In fact, and as shown in the Appendix, the relationship between the C and μ parameters is now $\mu = 2C/\rho^o$, with $\rho^o = \|W^o\|^2 + C\sum \xi_i^o$. Hence C and μ are not independent, and they should be chosen differently depending on which algorithm is to be used. In our experiments we have chosen for C the values proposed in [18] and once SMO finishes for each training–test pair, we have subsequently trained RCH–MDM using a μ value computed as just explained. Moreover, while the previous two vector initialization for SMO is still possible, this is not so for RCH–MDM and in this case we have chosen each sample barycenters as the initial $W_\pm$ vectors. All this makes final model comparisons somewhat less homogeneous than the square penalty ones, as shown in table 2, where now final accuracies are similar for both methods (but less so than in the square penalty case) and SMO models clearly have less support vectors. This last fact is due, however, to the different initializations used: if SMO is trained starting from the barycenters (not a good idea anyway), its final models have more SVs, implying that RCH–MDM is better at removing wrong initial SV choices (the algorithm is in some sense designed for that to be true). In any case, and for the initializations used, SMO is again faster than RCH–MDM.

5 Discussion

The SMO algorithm for SVM construction and, on the other hand, the geometrically inspired NPP solving algorithms such as extended MDM are usually discussed as different, independent methods. We have shown in this note that, however, these two methods are in fact very closely related, as they can be seen as maximum gain algorithms for working sets of 2 multipliers. More precisely, the extended MDM algorithm typically used to solve NPP essentially coincides with a restricted form of SMO in which the working set multipliers correspond to sample patterns in the same class. As we have numerically illustrated for quadratic penalties, the basic SMO and MDM algorithms seem to arrive at the same models when a moderately high precision is imposed in their final minima. However, SMO seems to be faster, something quite natural, as it has greater freedom when choosing at each iteration the maximum gain multipliers. While the linear penalty comparison is more involved, it seems clear that SMO is again faster. Another contribution of the present work is a proposal of an MDM algorithm for RCH_μ–NPP considerably simpler than previous ones.

While this would seem to imply that there will not be great advantages from the consideration of geometric algorithms for SVM construction, we point out that the usual speed enhancements for SMO, such as shrinking, can also be applied to the MDM algorithm. On the other hand, there has been a considerable amount of work in efficient solutions of the Minimum Norm Problem (MNP) for convex sets, the question that lies at the heart of the MDM algorithm. Given the close relationship shown here between the SMO and MDM methods, it is

thus conceivable that insights gained for MNP algorithms can provide new ways of accelerating SMO and other algorithms derived from it.

References

1. Vapnik, V.: The Nature of Statistical Learning Theory. Springer, New York (1995)
2. Schölkopf, B., Smola, A.: Learning with kernels support vector machines, regularization, optimization, and beyond. MIT Press, Cambridge (2002)
3. Platt, J.: Fast training of support vector machines using sequential minimal optimization. Advances in Kernel Methods - Support Vector Machines, 185–208 (1999)
4. Joachims, T.: Making large-scale support vector machine learning practical. Advances in Kernel Methods - Support Vector Machines, 169–184 (1999)
5. Keerthi, S., Shevade, S., Bhattacharyya, C., Murthy, K.: Improvements to platt's smo algorithm for SVM classifier design. Neural Computation 13(3), 637–649 (2001)
6. Fan, R., Chen, P.H., Lin, C.: Working set selection using second order information for training support vector machines. Journal of Machine Learning Research 6, 1889–1918 (2005)
7. Bennett, K., Bredensteiner, E.: Duality and geometry in svm classifiers. In: Proc. 17th Int. Conf. Machine Learning, pp. 57–64 (2000)
8. Keerthi, S., Shevade, S., Bhattacharyya, C., Murthy, K.: A fast iterative nearest point algorithm for support vector machine classifier design. IEEE Transactions on Neural Networks 11(1), 124–136 (2000)
9. Franc, V.: Simple solvers for large quadratic programming tasks. In: Kropatsch, W., Sablatnig, R., Hanbury, A. (eds.) DAGM 2005. LNCS, vol. 3663, pp. 75–84. Springer, Heidelberg (2005)
10. Gilbert, E.: Minimizing the quadratic form on a convex set. SIAM J. Contr. 4, 61–79 (1966)
11. Franc, V., Hlaváč, V.: An iterative algorithm learning the maximal margin classifier. Pattern Recognition 36, 1985–1996 (2003)
12. Mitchell, B., Dem'yanov, V., Malozemov, V.: Finding the point of a polyhedron closest to the origin. SIAM J. Contr. 12, 19–26 (1974)
13. Shawe-Taylor, J., Cristianini, N.: On the generalisation of soft margin algorithms. IEEE Transactions on Information Theory 48(10), 2711–2735 (2002)
14. Glasmachers, T., Igel, C.: Second order smo improves svm online and active learning. Neural Computation 20(2), 374–382 (2008)
15. Hush, D., Scovel, C.: Polynomial-time decomposition algorithms for support vector machines. Machine Learning 51(1), 51–71 (2003)
16. Chang, C.C., Lin, C.J.: Training ν-support vector classifiers: Theory and algorithms. Neural Computation 13(9), 2119–2147 (2001)
17. Tao, Q., Wu, G.W., Wang, J.: A general soft method for learning svm classifiers with l1-norm penalty. Pattern Recogn. 41(3), 939–948 (2008)
18. Rätsch, G.: Benchmark repository, `ida.first.fraunhofer.de/projects/bench/benchmarks.htm`

Appendix: The Equivalence between SVM and NPP

We will consider in what follows the linear penalty case, the arguments for the penalty–free situation being similar and simpler. It is well known that the KKT

conditions for SVM imply that at the optimum $W^o = \sum \alpha_i^o y_i X_i$ we have $\alpha_i^o = C$ if $\xi_i^o > 0$, and also

$$\alpha_i^o \left(y_i \left(W^o \cdot X_i + b^o \right) - 1 + \xi_i^o \right) = 0,$$

that is, $\alpha_i^o = \alpha_i^o \left(y_i \left(W^o \cdot X_i + b^o \right) + \xi_i^o \right)$. Summing over i gives

$$\begin{aligned} \sum \alpha_i^o &= \sum \alpha_i^o y_i W^o \cdot X_i + b^o \sum \alpha_i^o y_i + \sum \alpha_i^o \xi_i^o \\ &= \|W^o\|^2 + C \sum \xi_i^o, \end{aligned}$$

since $\sum \alpha_i^o y_i = 0$. If we write $\rho^o = \|W^o\|^2 + C \sum \xi_i^o$ and define now

$$W' = \frac{2}{\rho^o} W^o = \sum_i \frac{2\alpha_i^o}{\rho^o} y_i X_i = \sum_i \alpha_i' y_i X_i,$$

with $\alpha_i' = 2\alpha_i^o / \rho^o$, we shall show that W' coincides with the optimal solution W^* to the RCH_μ problem, with $\mu = 2C/\rho^o$ To prove it, notice first that $\sum_i \alpha_i' y_i = 0$, $\sum_i \alpha_i' = 2$ and $\alpha_i' \leq \mu$. Thus, W' is a feasible solution of the RCH_μ problem. For any other RCH_μ feasible $W = \sum \alpha_i y_i X_i$, we have

$$\begin{aligned} W \cdot W' &= \sum_i \alpha_i y_i W' \cdot X_i = \frac{2}{\rho^o} \sum_i \alpha_i y_i W^o \cdot X_i = \frac{2}{\rho^o} \sum_i \alpha_i y_i \left(W^o \cdot X_i + b^o \right) \\ &\geq \frac{2}{\rho^o} \sum_i \alpha_i \left(1 - \xi_i^o \right) \geq \frac{2}{\rho^o} \left(\sum_i \alpha_i - \frac{2C}{\rho^o} \sum_i \xi_i^o \right) \\ &= \frac{2}{\rho^o} \left(2 - \frac{2C}{\rho^o} \sum_i \xi_i^o \right) = \frac{4}{(\rho^o)^2} \left(\rho^o - C \sum_i \xi_i^o \right) \\ &= \frac{4}{(\rho^o)^2} \|W^o\|^2 = \|W'\|^2. \end{aligned}$$

By Schwarz's inequality this implies $\|W\| \geq \|W'\|$ and, in particular $\|W^*\| \geq \|W'\|$, which by the uniqueness of the NPP solution implies $W' = W^*$.