A Kernel Method for the Optimization of the Margin Distribution

F. Aiolli, G. Da San Martino, and A. Sperduti

Dept. of Pure and Applied Mathematics, Via Trieste 63, 35131 Padova - Italy,

Abstract. Recent results in theoretical machine learning seem to suggest that nice properties of the margin distribution over a training set turns out in a good performance of a classifier. The same principle has been already used in SVM and other kernel based methods as the associated optimization problems try to maximize the minimum of these margins.

In this paper, we propose a kernel based method for the direct optimization of the margin distribution (KM-OMD). The method is motivated and analyzed from a game theoretical perspective. A quite efficient optimization algorithm is then proposed. Experimental results over a standard benchmark of 13 datasets have clearly shown state-of-the-art performances.

Keywords: Kernel Methods, Margin Distribution, Game Theory

1 Introduction

Much of last-decade theoretical work on learning machines has been devoted to study the aspects of learning methods that control the generalization performance. In essence, two main features seem to be responsible for the generalization performance of a classifier, namely, keeping low the complexity of the hypothesis space (e.g. by limiting the VC dimension) and producing models which achieve large margin (i.e. confidence in the prediction) over the training set.

The good empirical effectiveness of two of the most popular algorithms, Support Vector Machines (SVM) and AdaBoost, have been in fact explained by the high margin classifiers they are able to produce. Specifically, hard margin SVMs return the hyperplane which keeps all the examples farest away from it, thus maximizing the minimum of the margin over the training set (worst-case optimization of the margin distribution). Similarly, AdaBoost, has been demonstrated to greedily minimize a loss function which is tightly related to the distribution of the margins on the training set. Despite the AdaBoost ability to optimize the margin distribution on the training set, it has been shown in [1] that in certain cases, it can also increase the complexity of the weak hypotheses, thus possibly leading to overfitting phenomena.

The effect of the margin distribution on the generalization ability of learning machines have been studied in [2] and [3], while algorithms trying to optimize explicitly the margin distribution include [4], [5] and [6]. More recently, it has been shown [7] that quite good effectiveness can even be obtained by the optimization of the first moment of the margin distribution (the simple average value over the training set). In this case, the problem can be solved very efficiently, since computing the model has time complexity $O(n)$.

In this paper, we propose a kernel machine which explicitly tries to optimize the margin distribution. Specifically, this boils down to an optimization of a weighted combination of margins, via a distribution over the examples, with appropriate constraints related to the entropy (as a measure of complexity) of the distribution.

In Section 1.1 some notation used through the paper is introduced. In Section 2 a game-theoretical interpretation of hard margin SVM is given in the bipartite instance ranking framework (i.e. the problem to induce a separation between positive and negative instances in a binary task) and the problem of optimizing the margin distribution is studied from the same perspective. This game-theoretic analysis leads us to a simple method for optimizing the distribution of the margins. Then, in Section 3, an efficient optimization algorithm is derived. Experimental results are presented in Section 4. Finally, conclusions are drawn.

1.1 Notation and Background

In the context of binary calssification tasks, the aim of a learning algorithm is to return a classifier which minimizes the error on a (unknown) distribution $\mathcal{D}_{\mathcal{X}\times\mathcal{Y}}$ of input/output pairs $(\mathbf{x}_i, y_i)$, $\mathbf{x}_i \in \mathbb{R}^d$, $y_i \in \{-1, +1\}$. The input to the algorithm is a set of pre-classified examples pairs $\mathcal{S} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$. With $S^+ = \{\mathbf{x}_1^+, \ldots, \mathbf{x}_p^+\}$ we denote the set of p positive instances, where $\mathbf{x}_i^+$ is the i-th positive instance in S. Similarly $S^- = \{\mathbf{x}_1^-, \ldots, \mathbf{x}_n^-\}$ denotes the set of n negative instances. Clearly, $N = n + p$.
In this paper, we denote by $\varGamma_m \subseteq \mathbb{R}^m$ the set of m-dimensional probability vectors, i.e. $\varGamma_m = \{\gamma \in \mathbb{R}^m | \sum_{i=1}^m \gamma_i = 1, \gamma_i \geq 0\}$. The *convex hull* $\mathbf{ch}(\mathcal{C})$ of a set $\mathcal{C} = \{\mathbf{c}_1, \ldots, \mathbf{c}_m | \mathbf{c}_i \in \mathbb{R}^d\}$, is the set of all affine combinations of points in $\mathcal{C}$ such that the weights γ_i of the combination are non-negative, i.e. $\mathbf{ch}(\mathcal{C}) = \{\gamma_1\mathbf{c}_1 + \cdots + \gamma_m\mathbf{c}_m | \gamma_i \in \varGamma_m\}$. We also generalize this definition, by defining the η-norm-*convex hull* of a set $\mathcal{C} \in \mathbb{R}^d$ as the subset of $\mathbf{ch}(\mathcal{C})$ which has weights with (squared) norm smaller than a given value η, i.e. $\mathbf{ch}_\eta(\mathcal{C}) = \{\gamma_1\mathbf{c}_1 + \cdots + \gamma_m\mathbf{c}_m \in \mathbf{ch}(\mathcal{C}) | \; ||\gamma||^2 \leq \eta, \frac{1}{m} \leq \eta \leq 1\}$. Note that, whenever $\eta = \frac{1}{m}$, a trivial set consisting of a single point (the average of points in $\mathcal{C}$), is obtained, while whenever $\eta = 1$ this set will coincide with the convex hull.

2 Game theory, learning and margin distribution

A binary classification problem can be viewed from two different points of view. Specifically, let $h \in \mathcal{H}$ be an hypothesis space, mapping instances on real values. In a first scenario, let call it *instance classification*, a given hypothesis is said to be consistent with a training set if $y_i h(\mathbf{x}_i) > 0$ (classification constraints) for each example of the training set. In this case, the prediction on new instances can be performed by using the sign of the decision function.

In a second scenario, which we may call *bipartite instance ranking*, a given hypothesis is said consistent with a training set if $h(\mathbf{x}_i^+) - h(\mathbf{x}_j^-) > 0$ (order constraints) for any positive instance $\mathbf{x}_i^+$ and any negative instance $\mathbf{x}_j^-$. Note that when an hypothesis is consistent, then it is always possible to define a threshold which correctly separates positive from negative instances in the training set. In this paper, we mainly focus on this second view, even if a similar treatment can be pursued for the other setting.

In the following, we give an interpretation of the hard-margin SVM as a two players zero-sum game in the bipartite instance ranking scenario presented above. First of all, we recall that, in the classification context, the formulation of the learning problem is based on the maximization of the minimum of the margin in the training set. Then, we propose to slightly modify the pay-off function of the game in order to have a flexible way to control the optimization w.r.t. the distribution of the margin in the training set.

2.1 Hard Margin SVM as a zero-sum game

Consider the following zero-sum game defined for a bipartite instance ranking scenario. Let $\mathcal{P}_{MIN}$ (the nature) and $\mathcal{P}_{MAX}$ (the learner) be the two players. On each round of the game, $\mathcal{P}_{MAX}$ picks an hypothesis h from a given hypotheses space $\mathcal{H}$, while (simultaneously) $\mathcal{P}_{MIN}$ picks a pair of instances of different classes $\mathbf{z} = (\mathbf{x}^+, \mathbf{x}^-) \in \mathcal{S}^+ \times \mathcal{S}^-$. $\mathcal{P}_{MAX}$ wants to maximize its pay-off defined as the achieved margin $\rho_h(\mathbf{z})$ on the pair of examples which, in this particular setting, can be defined by $h(\mathbf{x}^+) - h(\mathbf{x}^-)$. Note that the value of the margin defined in this way is consistent with the bipartite instance ranking setting since it is greater than zero for a pair whenever the order constraint is satisfied for the pair, and less than zero otherwise.

Considering the hypothesis space of hyperplanes defined by unit-length weight vectors

$$\mathcal{H} = \{h(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} \ - \theta | \ \mathbf{w} \in \mathbb{R}^d s.t. \ ||\mathbf{w}|| = 1, \text{ and } \theta \in \mathbb{R}\},$$

the margin is defined by the difference of the *scores* of the instances, that is

$$\rho_{\mathbf{w}}(\mathbf{x}^+, \mathbf{x}^-) = \mathbf{w}^\top \mathbf{x}^+ - \theta - \mathbf{w}^\top \mathbf{x}^- + \theta = \mathbf{w}^\top (\mathbf{x}^+ - \mathbf{x}^-)$$

Let now be given a mixed strategy for the $\mathcal{P}_{MIN}$ player defined by $\gamma^+ \in \varGamma_p$, the probability of each positive instance to be selected, and $\gamma^- \in \varGamma_n$ the correspondent probabilities for negative instances. We can assume that these probabilities can be marginalized as the associated events are independent. In other words, the probability to pick a pair $(\mathbf{x}_i^+, \mathbf{x}_j^-)$ is simply given by $\gamma_i^+ \gamma_j^-$. Hence, the value of the game, i.e. the expected margin obtained in a game, will be:

$$\begin{aligned} V((\gamma^+, \gamma^-), \mathbf{w}) &= \textstyle\sum_{i,j} \gamma_i^+ \gamma_j^- \mathbf{w}^\top (\mathbf{x}_i^+ - \mathbf{x}_j^-) \\ &= \mathbf{w}^\top (\textstyle\sum_i \gamma_i^+ \mathbf{x}_i^+ (\sum_j \gamma_j^-) - \sum_j \gamma_j^- \mathbf{x}_j^- (\sum_i \gamma_i^+)) \\ &= \mathbf{w}^\top (\textstyle\sum_i \gamma_i^+ \mathbf{x}_i^+ - \sum_j \gamma_j^- \mathbf{x}_j^-) \end{aligned}$$

Then, when the player $\mathcal{P}_{MIN}$ is left free to choose its strategy, we obtain the following problem which determines the equilibrium of the game, that is

$$\min_{\gamma^+ \in \varGamma_p, \gamma^- \in \varGamma_n} \max_{\mathbf{w} \in \mathcal{H}} \ \mathbf{w}^\top (\sum_i \gamma_i^+ \mathbf{x}_i^+ - \sum_j \gamma_j^- \mathbf{x}_j^-)$$

Now, it easy to realize that a pure strategy is right available to the $\mathcal{P}_{MAX}$ player. In fact, it can maximize its pay-off by setting

$$\hat{\mathbf{w}} = \begin{cases} \text{any } \mathbf{w} \in \mathcal{H} & \text{if } \mathbf{v}(\gamma^+, \gamma^-) = \mathbf{0} \\ \frac{\mathbf{v}(\gamma^+,\gamma^-)}{||\mathbf{v}(\gamma^+,\gamma^-)||} & \text{otherwise} \end{cases}, \text{ where } \mathbf{v}(\gamma^+, \gamma^-) = \sum_i \gamma_i^+ \mathbf{x}_i^+ - \sum_i \gamma_i^- \mathbf{x}_i^-$$

Note that the condition $\mathbf{v}(\gamma^+, \gamma^-) = \mathbf{0}$ implies that the (signed) examples $y_i\mathbf{x}_i$ are not linearly independent, i.e. there exists a linear combination of these instances with not all null coefficients which is equal to the null vector. This condition is demonstrated to be necessary and sufficient for the non linear separability of a set (see [8]).
When the optimal strategy for $\mathcal{P}_{MAX}$ has been chosen, the expected value of the margin according to the probability distributions (γ^+, γ^-), i.e. the value of the game, will be:

$$E[\rho_{\hat{\mathbf{w}}}(\mathbf{x}^+, \mathbf{x}^-)] = \hat{\mathbf{w}}^\top \mathbf{v}(\gamma^+, \gamma^-) = ||\mathbf{v}(\gamma^+, \gamma^-)||$$

Note that the vector $\mathbf{v}(\gamma^+, \gamma^-)$ is defined by the difference of two vectors in the convex hulls of the positive and negative instances respectively, $\mathbf{v}_+ = \sum_i \gamma_i^+ \mathbf{x}_i^+ \in \mathbf{ch}(\mathcal{S}^+)$ and $\mathbf{v}_- = \sum_i \gamma_i^- \mathbf{x}_i^- \in \mathbf{ch}(\mathcal{S}^-)$. Moreover, when γ^+ and γ^- are uniform on their respective sets, the vector $\mathbf{v}(\gamma^+, \gamma^-)$ will be the difference between average points of the sets (a.k.a. their centroids).

Now, we are able to show that the best strategy for $\mathcal{P}_{MIN}$ is the solution obtained by an SVM. For this, let us rewrite the vector $\mathbf{v}(\gamma^+, \gamma^-)$ using a single vector of parameters,

$$\mathbf{v}(\gamma^+, \gamma^-) \equiv \mathbf{v}(\gamma) = \sum_i^N y_i \gamma_i \mathbf{x}_i$$

which can be obtained by a simple change of variables

$$\gamma_i = \begin{cases} \gamma_r^+ & \text{if } \mathbf{x}_i \text{ is the } r\text{-th } \textit{positive} \text{ example} \\ \gamma_r^- & \text{if } \mathbf{x}_i \text{ is the } r\text{-th } \textit{negative} \text{ example} \end{cases}$$

Using the fact that minimizing the squared norm is equivalent to minimize the norm itself, we may formulate the optimization problem to compute the best strategy for $\mathcal{P}_{MIN}$ (which aims to minimize the value of the game):

$$\min_{\gamma^+ \in \varGamma_p, \gamma^- \in \varGamma_n} ||\mathbf{v}(\gamma^+, \gamma^-)|| = \begin{cases} \min_\gamma ||\mathbf{v}(\gamma)||^2 \\ \text{s.t.} \sum_{i:y_i=y} \gamma_i = 1, \forall y \in \{-1, +1\}, \text{ and } \gamma_i \geq 0 \end{cases} \tag{1}$$

As already demonstrated in [9] the problem on the right of Eq. 1 is the same as hard margin SVM when a bias term is present. Specifically, the bias term is chosen as the score of the point standing in the middle between the points $\mathbf{v}_+$ and $\mathbf{v}_-$, i.e.

$$\theta = \frac{1}{2} \hat{\mathbf{w}}^\top (\mathbf{v}_+ + \mathbf{v}_-).$$

Then, the solutions of the two problems are the same. Specifically, the solution maximizes the minimum of the margins in the training set. Clearly, when the training set is not linearly separable, the solution of the problem in Eq.1 will be $\mathbf{v}(\gamma) = \mathbf{0}$.

2.2 Playing with Margin Distributions

The maximization of the minimum margin is not necessarily the optimal choice when dealing with a classification task. In fact, many recent works, including [4, 3], have demonstrated that the generalization error depends more properly on the distribution of (lowest) margins in the training set.

Our main idea is then to construct a problem which makes easy to play with the margin distribution. Specifically, we aim at a formulation that allow us to specify a given trade-off between the minimal value and the average value of the margin on the training set.

For this, we extend the previous game considering a further cost for the player $\mathcal{P}_{MIN}$. Specifically, we want to penalize, to a given extent, too pure strategies in such a way to have solutions which are robust with respect to different training example distributions. In this way, we expect to reduce the variance of the best strategy estimation when different training sets are drawn from the true distribution of examples $\mathcal{D}_{\mathcal{X}\times\mathcal{Y}}$. A good measure of the complexity of $\mathcal{P}_{MIN}$ behavior would certainly be the normalized entropy of its strategy which can be defined by

$$E(\gamma) = \frac{1}{2}\left(\frac{1}{\log(p)}\sum_i \gamma_i^+ \log\frac{1}{\gamma_i^+} + \frac{1}{\log(n)}\sum_i \gamma_i^- \log\frac{1}{\gamma_i^-}\right)$$

which has maximum value 1 whenever γ is the uniform distribution on both sets (completely unpredictable startegy) and is 0 when the distribution is picked on a single example per set (completely predictable pure strategy).

However, a (simpler) approximate version of the entropy as defined above, can be obtained by considering the 2-norm of the distribution. In fact, it is well known that, for any distribution $\gamma \in \Gamma_m$ it always holds that $||\gamma||_2 \leq ||\gamma||_1$. Moreover, $||\gamma||_2$ is minimal whenever γ is a uniform distribution and is equal to 1 whenever γ is a pure strategy. Specifically, we can consider the following approximation:

$$E(\gamma) \approx \frac{m}{m-1}(1 - ||\gamma||_2^2)$$

Considering the squared norm of the distribution, we can reformulate the strategy of $\mathcal{P}_{MIN}$ as a trade-off between two objective functions, with a trade-off parameter λ:

$$\min_{\gamma^+\in\Gamma_p,\gamma^-\in\Gamma_n} (1-\lambda)||\mathbf{v}(\gamma)||^2 + \lambda||\gamma||^2 \qquad (2)$$

It can be shown that the optimal vector $\mathbf{v}(\hat{\gamma})$ which is solution of the problem above, represents the vector joining two points, $\mathbf{v}_+$ into the positive norm-restricted convex hull, i.e. $\mathbf{v}_+ \in \mathbf{ch}_\eta(\mathcal{S}^+)$, and $\mathbf{v}_-$ into the negative norm-restricted convex hull, i.e. $\mathbf{v}_- \in \mathbf{ch}_\eta(\mathcal{S}^-)$, for opportune η.

Similarly to the hard margin SVM, the threshold is defined as the score of the point which is in the middle between these two points, i.e. $\theta = \frac{1}{2}\hat{\mathbf{w}}^\top(\mathbf{v}_+ + \mathbf{v}_-)$.

Finally, it is straightforward to see that this method generalizes (when $\lambda = 1$), the baseline method presented in [10] where the simple difference between the centroid of positives and the centroid of negatives is used as the weight vector, and obviously it generalizes the hard-margin SVM for $\lambda = 0$.

3 Optimization Algorithm

We now propose a very simple method to optimize the problem in Eq. (1). The proposed method optimizes the objective function by a SMO-like procedure which maintains a feasible solution γ at each step, starting from a feasible γ_{init}. Specifically, in order to maintain the solution in the feasible set, at each step, it chooses a pair of variables (γ_r, γ_q) associated to examples of the same class, let say y, and imposes an update of the form:

$$\gamma_r' \leftarrow \gamma_r + \epsilon, \; \gamma_q' \leftarrow \gamma_q - \epsilon$$

With this update, we can exactly measure the extent of the change which occurs to the vector $\mathbf{v}(\gamma)$, i.e.

$$\varDelta_{\mathbf{v}(\gamma)} = \mathbf{v}(\gamma') - \mathbf{v}(\gamma) = y\epsilon(\mathbf{x}_r - \mathbf{x}_q)$$

To improve the readability of the following derivations, let us denote by $s_j(\gamma)$ the quantity $s_j(\gamma) = \sum_i y_i \gamma_i \mathbf{x}_i^\top \mathbf{x}_j$. Now, we are able to evaluate how much, and how, the update on the γ's modifies the objective function.
Let us begin from the first term $G(\gamma) = ||\mathbf{v}(\gamma)||^2$. In this case, we have:

$$G(\gamma') = ||\mathbf{v}(\gamma')||^2 = ||\mathbf{v}(\gamma) + \varDelta_{\mathbf{v}(\gamma)}||^2 = ||\mathbf{v}(\gamma)||^2 + ||\varDelta_{\mathbf{v}(\gamma)}||^2 + 2\mathbf{v}(\gamma)\varDelta_{\mathbf{v}(\gamma)}$$

and hence the variation can be obtained by

$$\varDelta_{G(\gamma)} = G(\gamma') - G(\gamma) = ||\varDelta_{\mathbf{v}(\gamma)}||^2 + 2\mathbf{v}(\gamma)^\top \varDelta_{\mathbf{v}(\gamma)}$$

where $||\varDelta_{\mathbf{v}(\gamma)}||^2 = \epsilon^2 ||\mathbf{x}_r - \mathbf{x}_q||^2$ and $\mathbf{v}(\gamma)^\top \varDelta_{\mathbf{v}(\gamma)} = \epsilon y(s_r(\gamma) - s_q(\gamma))$.
Summarizing, we have

$$\varDelta_{G(\gamma)} = \epsilon^2 ||\mathbf{x}_r - \mathbf{x}_s||^2 + 2\epsilon y(s_r(\gamma) - s_q(\gamma))$$

For what concerns the second term $H(\gamma) = ||\gamma||^2$, we have

$$H(\gamma') = ||\gamma + \varDelta_\gamma||^2 = ||\gamma||^2 + ||\varDelta_\gamma||^2 + 2\gamma^\top \varDelta_\gamma = ||\gamma||^2 + 2\epsilon^2 + 2\epsilon(\gamma_r - \gamma_q)$$

and thus

$$\varDelta_{H(\gamma)} = H(\gamma') - H(\gamma) = 2\epsilon^2 + 2\epsilon(\gamma_r - \gamma_q)$$

Thus the overall objective function $L(\gamma) = (1-\lambda)||\mathbf{v}(\gamma)||^2 + \lambda||\gamma||^2$ will vary of an amount

$$\begin{aligned} \varDelta_{L(\gamma)} &= L(\gamma') - L(\gamma) = (1-\lambda)\varDelta_{G(\gamma)} + \lambda \varDelta_{H(\gamma)} \\ &= (1-\lambda)(\epsilon^2 ||\mathbf{x}_r - \mathbf{x}_q||^2 + 2y\epsilon(s_r(\gamma) - s_q(\gamma))) + 2\lambda(\epsilon^2 + \epsilon(\gamma_r - \gamma_q)) \\ &= ((1-\lambda)||\mathbf{x}_r - \mathbf{x}_q||^2 + 2\lambda)\epsilon^2 + 2((1-\lambda)y(s_r(\gamma) - s_q(\gamma)) + \lambda(\gamma_r - \gamma_q))\epsilon \end{aligned} \tag{3}$$

Finally, setting the derivative of $\varDelta_{L(\gamma)}$ w.r.t. to ϵ, to zero, we are able to find the point of minimum (or maximum improvement):

$$\frac{\partial \varDelta_{L(\gamma)}(\epsilon)}{\partial \epsilon} = 2\epsilon((1-\lambda)||\mathbf{x}_r - \mathbf{x}_q||^2 + 2\lambda) + 2((1-\lambda)(\rho_r - \rho_q) + \lambda(\gamma_r - \gamma_q)) = 0$$

thus obtaining

$$\hat{\epsilon} = \frac{\lambda(\gamma_q - \gamma_r) + (1-\lambda)y(s_q(\gamma) - s_r(\gamma))}{2\lambda + (1-\lambda)||\mathbf{x}_r - \mathbf{x}_q||^2} \tag{4}$$

In order to maintain the solution in the feasibility set, the constraints $\hat{\epsilon} < -\gamma_r$, and $\hat{\epsilon} < \gamma_q$ must be enforced. This is made by simply doing a cut with the formula:

$$\hat{\epsilon} = \min\{\hat{\epsilon}, -\gamma_r, \gamma_q\}.$$

3.1 The Algorithm

In the following a pseudo-code of the proposed algorithm is presented.

```
input:
  training set $S = \{(\mathbf{x}_i, y_i)\}_{i=1,\dots,N}$
  the convergence tolerance $\delta > 0$
initialize:
  $\gamma_i = \frac{1}{p}$ if $y_i = 1$, $\gamma_i = \frac{1}{n}$ if $y_i = -1$, $\{s_j = \sum_i y_i \gamma_i \mathbf{x}_i^\top \mathbf{x}_j\}_{j=1,\dots,N}$
repeat
    for each pair $(\gamma_r, \gamma_q)$ associated to examples of a same class $y$
        compute $\hat{\epsilon} = \hat{\epsilon}(\gamma_r, \gamma_q)$ as in Eq. 4
        set $\hat{\epsilon} = \min(\hat{\epsilon}, -\gamma_r)$
        set $\hat{\epsilon} = \min(\hat{\epsilon}, \gamma_q)$
        compute the delta loss $\Delta_{L(\gamma)}$ as in Eq. 3
        if $\Delta_{L(\gamma)}(\hat{\epsilon}) > \delta$ then (update step)
            $\gamma_r = \gamma_r + \epsilon, \gamma_q = \gamma_q - \epsilon$
            $s_j = s_j + \epsilon(\mathbf{x}_r^\top \mathbf{x}_j - \mathbf{x}_q^\top \mathbf{x}_j), j = 1, \dots, N$
        end if
    end for
until no update steps have been performed in this iteration
compute the threshold $\theta = \frac{1}{2}\sum_j s_j$
return the vector $\gamma$, and the threshold $\theta$.
```

4 Experiments and Results

The proposed approach has been tested against a popular benchmark consisting of 13 binary datasets[1]. For each dataset, there are 100 different training/test splits of the same data. The classification error is computed as the average error among all these realizations. For a detailed description of the datasets see [11]. In order to have a fair comparison with the results in [11], the same model selection methodology has been used: the best parameter setting for each of the first 5 realizations has been obtained using 5-fold cross validation. Then, the final value of the parameters, namely λ and γ (for the RBF kernel), is selected as the median of the 5 best values obtained in validation. With these parameters, our method has been run again on each training data and then tested for all 100 realizations.

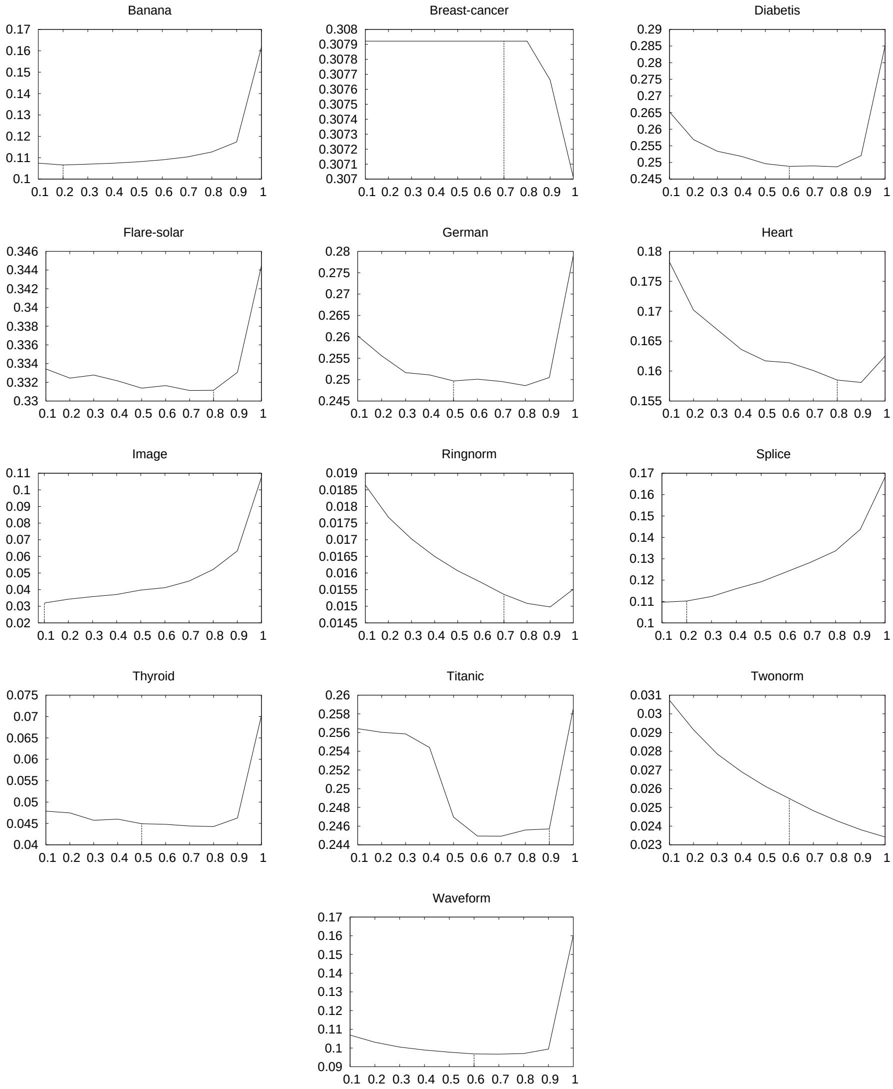

Fig. 1. Classification error (y-axis) with respect to lambda values (x-axis) for each dataset. The vertical dotted line represent the lambda value selected on validation.

	SVM	KM-OMD	Best in [11] [12]
banana	0.11530 (± 0.0660)	**0.10660** (± 0.0150)	0.10730 (± 0.0430)
breast-cancer	**0.26040** (± 0.0474)	0.30792 (± 0.1206)	0.24770 (± 0.0463)
diabetis	**0.23530** (± 0.0173)	0.24883 (± 0.0455)	0.23210 (± 0.0163)
flare-solar	**0.32430** (± 0.0182)	0.33115 (± 0.0489)	0.32430 (± 0.0182)
german	**0.23610** (± 0.0207)	0.24970 (± 0.0570)	0.23610 (± 0.0207)
heart	0.15950 (± 0.0326)	**0.15850** (± 0.0915)	0.15950 (± 0.0326)
image	**0.02960** (± 0.0060)	0.03198 (± 0.0106)	0.02670 (± 0.0061)
ringnorm	0.01660 (± 0.0012)	**0.01536** (± 0.0046)	0.01490 (± 0.0012)
splice	**0.10880** (± 0.0066)	0.11025 (± 0.0142)	0.09500 (± 0.0065)
thyroid	0.04800 (± 0.0219)	**0.04585** (± 0.0475)	0.04200 (± 0.0207)
titanic	**0.22420** (± 0.0102)	0.24570 (± 0.1288)	0.22420 (± 0.0102)
twonorm	0.02960 (± 0.0023)	**0.02548** (± 0.0042)	0.02610 (± 0.0015)
waveform	0.09880 (± 0.0043)	**0.09685** (± 0.0110)	0.09790 (± 0.0081)

Table 1. Classification error of SVM and KM-OMD (our method) on 13 datasets. Between brackets the standard deviation. Best method is in bold. When underlined, the method improves the best result, according to [11] and [12], obtained for the dataset (which is reported in the third column).

Table 1 summarizes the obtained results and compare them against the ones reported in [11] and [12]. Note that the proposed algorithm is better than SVM on 6 datasets. Moreover, on 4 datasets, our method improves over the best performing method.

Figure 1 shows how the error rate varies with respect to λ. Generally speaking, the error rate tends to decrease towards the optimum, and then increase again when λ approaches 1. On the other hand, Figure 2 shows how the training time is affected by λ values. This is only plotted for 2 datasets, namely Image and Waveform, since the of curves we obtained for the other datasets were similar. The two plots clearly show that learning with low λ values requires more training time, whereas models for higher λ values are faster to compute. It is worth noting that, in most cases, even high λ values (for which the models are much faster to train) give anyway good performances, or at least acceptable when the computational time is an issue.

5 Conclusions

We have addressed the problem of optimizing the distribution of the margins from a game-theoretical perspective. A simple method has been proposed which consist in optimizing a trade-off between two extreme optimization tasks: the maximization of the minimum margin and the maximization of the average of the margin. The experimental results have shown state-of-the-art performances for some datasets.

In future work, we would like to study under which conditions (e.g. conditions related to the data distribution) our method is to prefer to other state-of-the-art methods. Moreover, the very simple algorithm we have proposed in this paper is not optimized

[1] All datasets can be downloaded from: http://ida.first.fraunhofer.de/projects/bench/benchmarks.htm

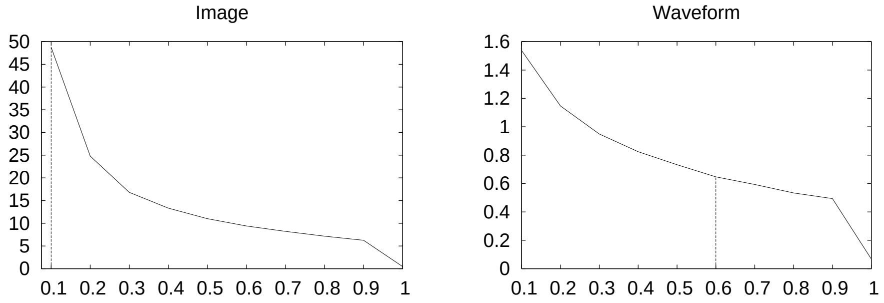

Fig. 2. Training time (y-axis) in seconds with respect to lambda values (x-axis) for Image and Waveform datasets. The vertical dotted line represent the lambda value selected on validation.

and its optimization could be another direction of our future research. When an optimized version of the algorithm will be available, fair time comparisons with state-of-the-art software for SVM, including SVMLight, will be possible.

References

1. Reyzin, L., Schapire, R.: How boosting the margin can also boost classifier complexity. In: Proceedings of the 23rd International Conference on Machine Learning (ICML). (2006)
2. Garg, A., Har-Peled, S., Roth, D.: On generalization bounds, projection profile, and margin distribution. In: Proceedings of the 11th International Conference on Machine Learning (ICML). (2002)
3. Shawe-Taylor, J., Cristianini, N.: Further results on the margin distribution. In: Proceedings of the 15th International Conference on Machine Learning (ICML). (2003)
4. Garg, A., Roth, D.: Margin distribution and learning algorithms. In: Proceedings of the 12th Conference on Computational Learning Theory (COLT). (1999)
5. Mason, L., Bartlett, P., Baxter, J.: Improved generalization trough explicit optimization of margins. Machine Learning **38-3** (2000) 243–255
6. Aiolli, F., Sperduti, A.: A re-weighting strategy for improving margins. Artifical Intelligence Journal **137** (2002) 197–216
7. Pelckmans, K., Suykens, J., Moor, B.D.: A risk minimization principle for a class of parzen estimators. In: Advances in Neural Information Processing Systems. (2007)
8. Siu, K.Y., Roychowdhury, V., Kailath, T.: Discrete Neural Computation. Englewood Cliffs, New Jersey: Prentice Hall (1995)
9. Bhattacharyya, C., Keerthi, S., Murthy, K., Shevade, S.: A fast iterative nearest point algorithm for support vector machine classifier design. IEEE Transactions on Neural Networks (2000)
10. Scholkopf, B., Smola, A.: Learning with Kernels. MIT Press, Cambridge, Massachusetts (2002)
11. Rätsch, G., Onoda, T., Müller, K.R.: Soft margins for adaboost. Mach. Learn. **42**(3) (2001) 287–320
12. Mika, S., Rätsch, G., Müller, K.R.: A mathematical programming approach to the kernel fisher algorithm. In: NIPS. (2000) 591–597