Technische Universität Berlin

Forschungsberichte
der Fakultät IV – Elektrotechnik und Informatik

A Multi-Class Support Vector Machine Based on Scatter Criteria

Robert Jenssen Marius Kloft, Alexander Zien,
Sören Sonnenburg, and Klaus-Robert Müller

Bericht-Nr. 2009-14
ISSN 1436-9915

A Multi-Class Support Vector Machine Based on Scatter Criteria

Robert Jenssen[1], Marius Kloft[2], Alexander Zien[3,4], Sören Sonnenburg[3], and Klaus-Robert Müller[2]
[1] Department of Physics and Technology, University of Tromsø, Norway
[2] Machine Learning Group, Dept. of Computer Science, TU Berlin, Germany
[3] Friedrich Miescher Laboratory, Rätsch Group, Tübingen, Germany
[4] Fraunhofer Institute FIRST, Berlin, Germany

July 2, 2009

Abstract

We re-visit Support Vector Machines (SVMs) and provide a novel interpretation thereof in terms of weighted class means and scatter theory. The gained theoretical insight can be translated into a highly efficient extension to multi-class SVMs: mScatter-SVMs. Numerical simulations reveal that more than an order of magnitude speed-up can be gained while the classification performance remains largely unchanged at the level of the classical one vs. rest and one vs. one implementation of multi-class SVMs.

1 Introduction

Support Vector Machines (SVMs) [Vap98] have become one of the standard tools for binary classification. However their multi-class extensions are considered complex and still have rather demanding computational costs, as quantified in terms of the number N of training data points and the number C of classes. The classic heuristics reduce a multi-class problem to several binary sub-problems by decomposing it either into $C(C-1)/2$ one vs. one or into C one vs. rest tasks. A next generation of multi-class SVM formulates one large joint optimization problem [Vap98, WW99, BVB95, LLW04, HL02] and studies various ways for speed-up [CS01, FH02]; for this, notably $C \times N$ variables typically need to be optimized, compared to N for binary SVMs. An alternative line of research has re-interpreted multi-class SVMs as multidimensional subspace projection method at improved computing times [SST05].

The present work contributes a multi-class SVM algorithm of similar complexity as its binary counterpart. Also here an interesting re-interpretation provides the key insight. We start with the so-called μ-SVM formulation and point out its relation to *scattering* in pattern recognition (see Section 2). The theoretical insight obtained thereby allows to reformulate the multi-class optimization problem in terms of class means as well as a global reference with only N variables and a nice intuitive geometrical interpretation (see Fig. 1). After a brief remark about test rules (Section 3), details are given on our fast implementation scheme (Section 4). We follow up by detailed simulation studies (Section 5), revealing that our novel method is indeed faster at a similar generalization accuracy as previous multi-class methods, before concluding the paper (Section 6).

2 A New Interpretation of the μ-SVM

A number of equivalent SVM formulations have emerged over the years, e.g. C-, ν-, and μ-SVMs [Vap98, SSWB00, CB99], or also the convex hull formulation of SVMs [MT06]. In this paper, we will re-interprete the μ-SVM [CB99] in terms of weighted class mean vectors and scatter theory. Let us consider the following optimization problem of training a μ-SVM,

$$\begin{array}{ll} \min\limits_{\mathbf{w},b,\rho,\xi_i} & \frac{1}{2}\|\mathbf{w}\|^2 - 2\rho + \mu\sum_{i=1}^{N}\xi_i \\ \text{s.t.} & y_i(\mathbf{w}^\top\mathbf{x}_i + b) \geq \rho - \xi_i, \quad \text{and} \quad \xi_i \geq 0, \quad \forall i = 1,\ldots,N \end{array} \tag{1}$$

where $y_i \in \{-1, 1\}$ denotes the label of training data point $\mathbf{x}_i$ generated from class $\omega_1 : y_i = 1$ or $\omega_2 : y_i = -1$. This procedure maximizes the *margin* $\frac{2\rho}{\|\mathbf{w}\|}$ of the hyperplane determined by $\mathbf{w}$ and b (see Fig. 1 (a) for an illustration). The dual optimization problem becomes

$$\min_{\boldsymbol{\alpha}} \quad \frac{1}{2}\boldsymbol{\alpha}^\top\boldsymbol{\mathcal{K}}\boldsymbol{\alpha}, \quad \boldsymbol{\alpha} = \begin{bmatrix} \boldsymbol{\alpha}_1 \\ \boldsymbol{\alpha}_2 \end{bmatrix}, \quad \boldsymbol{\mathcal{K}} = \begin{bmatrix} \mathbf{K}_{11} & -\mathbf{K}_{12} \\ -\mathbf{K}_{21} & \mathbf{K}_{22} \end{bmatrix},$$

subject to $\boldsymbol{\alpha}_c^\top\mathbf{1} = 1,\ c = 1, 2$ and $0 \leq \alpha_i \leq \mu,\ \forall i$, where $\mathbf{1}$ is an all ones vector. The subscripts indicate the two classes and $\mathbf{K}_{cc'}$ are inner-product matrices within and between classes (see also [CB99]). This optimization determines $\mathbf{w}$

$$\mathbf{w} = \sum_{i:y_i=1}\alpha_i\mathbf{x}_i - \sum_{i:y_i=-1}\alpha_i\mathbf{x}_i. \tag{2}$$

In the classification of a data point $\mathbf{x}_t$ unseen to the SVM during training, the primal formulation suggests a testing rule

$$\mathbf{x}_t \to \omega_1 : \quad \mathbf{w}^\top\mathbf{x}_t + b > 0, \quad \text{otherwise} \quad \mathbf{x}_t \to \omega_2.$$

The dual formulation of the μ-SVM has an interesting interpretation in terms of distances between convex hulls, see for example [CB99, MT06].

2.1 Weighted Class Mean Vectors and Scatter Interpretation

By Lagrange theory, the μ-SVM hyperplane weight vector is given by Eq. (2). Let

$$\mathbf{m}_1 = \sum_{i:y_i=1}\alpha_i\mathbf{x}_i, \qquad \mathbf{m}_2 = \sum_{i:y_i=-1}\alpha_i\mathbf{x}_i.$$

Since the weighting coefficients are such that $\sum_{i:y_i=1}\alpha_i = \sum_{i:y_i=-1}\alpha_i = 1$ [CB99], $\mathbf{m}_1$ and $\mathbf{m}_2$ are per definition weighted class mean vectors.

Hence, it is readily observed that the μ-SVM optimization in the dual adjusts the weights of $\mathbf{m}_1$ and $\mathbf{m}_2$ in order to minimize the squared Euclidean distance between these two vectors, that is

$$\min_{\boldsymbol{\alpha}} \frac{1}{2}\|\mathbf{m}_1 - \mathbf{m}_2\|^2.$$

This operation will move $\mathbf{m}_1$ and $\mathbf{m}_2$ to the boundary region between the two classes. These weighted class mean vectors may be thought of as representatives of their respective classes. Hence, a class will not be represented by a "typical" example, e.g. the un-weighted class mean, but rather by a weighted combination of data points which are un-typical, in the sense that they are more similar to the other class. This property is exactly the prime catalyst of the superior generalization

ability of the SVM compared to methods which give equal weights to all data points. It is also possible to express the parameter b in geometrical terms by invoking the KKT conditions:

$$b = -\frac{1}{2}(\mathbf{m}_1 - \mathbf{m}_2)^\top(\mathbf{m}_1^{msv} + \mathbf{m}_2^{msv}),$$

where I_c^{msv} indicates the margin support vectors for which $0 < \alpha_i < \mu$ and $\mathbf{m}_c^{msv} = \frac{1}{\Omega_c}\sum_{i \in I_c^{msv}} \alpha_i \mathbf{x}_i$ with $\Omega_c = \sum_{i \in I_c^{msv}} \alpha_i$ are the weighted mean vectors of the margin support vectors for each class, and are therefore themselves situated on the $\pm\rho$ margin respectively. It is interesting to note that in the hard margin case, i.e. all $\xi_i = 0$, then $b = -\frac{1}{2}(\mathbf{m}_1 - \mathbf{m}_2)^\top(\mathbf{m}_1 + \mathbf{m}_2)$. This means that the separating hyperplane in the hard margin case passes through the point $\bar{\mathbf{m}} = \frac{1}{2}(\mathbf{m}_1 + \mathbf{m}_2)$ which is the arithmetic mean of $\mathbf{m}_1$ and $\mathbf{m}_2$. This is in direct analogy with the convex hull view. With the arithmetic mean $\bar{\mathbf{m}}$ introduced into the picture, it is also possible to consider the testing rule from a new viewpoint. We have

$$\mathbf{x}_t \to \omega_1 : \quad (\mathbf{m}_1 - \mathbf{m}_2)^\top \mathbf{x}_t - \frac{1}{2}(\mathbf{m}_1 - \mathbf{m}_2)^\top(\mathbf{m}_1^{msv} + \mathbf{m}_2^{msv}) \geq 0 \quad \text{otherwise} \quad \mathbf{x}_t \to \omega_2.$$

Since $\mathbf{m}_1 - \bar{\mathbf{m}} = \frac{1}{2}(\mathbf{m}_1 - \mathbf{m}_2)$ and $\mathbf{m}_2 - \bar{\mathbf{m}} = \frac{1}{2}(\mathbf{m}_2 - \mathbf{m}_1)$, an equivalent expression for the testing rule is

$$\mathbf{x}_t \to \omega_c : \quad \arg\max_{c=1,2} \quad (\mathbf{m}_c - \bar{\mathbf{m}})^\top \mathbf{x}_t - (\mathbf{m}_c - \bar{\mathbf{m}})^\top \mathbf{m}_c^{msv}. \tag{3}$$

In this formulation, both the functional distance and the geometric distance from the hyperplane $(\mathbf{m}_c - \bar{\mathbf{m}})^\top \mathbf{x} - (\mathbf{m}_c - \bar{\mathbf{m}})^\top \mathbf{m}_c^{msv}$ to the point $\mathbf{x} = \mathbf{m}_c^{msv}$ equals 0, since $\|\mathbf{m}_1 - \bar{\mathbf{m}}\| = \|\mathbf{m}_2 - \bar{\mathbf{m}}\|$ such that functional and geometric distances are equivalent. Based on the above discussion, we also have $\|\mathbf{m}_1 - \mathbf{m}_2\| = \|\mathbf{m}_1 - \bar{\mathbf{m}}\| + \|\mathbf{m}_2 - \bar{\mathbf{m}}\|$. This allows us to express the μ-SVM dual optimization problem differently, since for $C = 2$ classes

$$\min_{\boldsymbol{\alpha}} \frac{1}{2}\|\mathbf{m}_1 - \mathbf{m}_2\|^2 = \min_{\boldsymbol{\alpha}} \sum_{c=1}^{C} \frac{1}{C}\|\mathbf{m}_c - \bar{\mathbf{m}}\|^2. \tag{4}$$

Interestingly, this cost function is equal to the *between class scatter* [DHS01], where scatter is measured with respect to the weighted class mean vectors $\mathbf{m}_1$ and $\mathbf{m}_2$ and we assume equal class priors $P_1 = P_2 = \frac{1}{2}$. This means that we may interpret the μ-SVM as a method which in training adjust the weights of $\mathbf{m}_1$ and $\mathbf{m}_2$ in order to minimize the scatter between the classes under the equal prior assumption. This will force $\mathbf{m}_1$ and $\mathbf{m}_2$ to the boundary region, thus emphasizing the un-typical data points in each class. This is illustrated in Fig. 1 (b).

In this new framework, we note that in testing, each class is represented by two weighted mean vectors. One is the overall weighted mean vector $\mathbf{m}_c$ which depends on all the support vectors, i.e. all $\mathbf{x}_i$ for which $\alpha_i > 0$. The other representative is the weighted mean of the margin support vectors $\mathbf{m}_c^{msv}$ which depends only on those $\mathbf{x}_i$ for which $0 < \alpha_i < \mu$.

The weighted mean vectors and scatter-based view of the μ-SVM introduced here not only provides a new interpretation, it also suggests extensions in two related directions, by defining $P_c = \frac{1}{C}$ and $\mathbf{m}_c = \sum_{i:y_i=c} \alpha_i \mathbf{x}_i$ such that $\sum_{i:y_i=c} \alpha_i = 1$ and $\bar{\mathbf{m}} = \frac{1}{C}\sum_{c=1}^{C} \mathbf{m}_c$, Eq. (4) may be used for a multi-class formulation where in the dual the between-class scatter between multiple classes is minimized. An incorporation of unequal class priors into the picture is straight forward but beyond the scope of this contribution.

2.2 Multi-Class Extension of the μ-SVM

Inspired by the above reasoning, we propose a novel multi-class extension of the μ-SVM primal, namely

$$\begin{aligned} \min_{\mathbf{w},\mathbf{w}_c,b_c,\rho,\xi_i} \quad & \frac{1}{2}\sum_{c=1}^{C}\frac{1}{C}\|\mathbf{w}_c-\bar{\mathbf{w}}\|^2 - C\rho + \mu\sum_i \xi_i \\ \text{s.t.} \quad & (\mathbf{w}_{y_i}-\bar{\mathbf{w}})^\top \mathbf{x}_i + b_{y_i} \geq \rho - \xi_i, \quad \text{and} \quad \xi_i \geq 0, \quad \forall i, \end{aligned} \tag{5}$$

where $y_i \in \{1,\ldots,C\}$ is the label for $\mathbf{x}_i$. Note that for $C=2$ Eq. (1) is recovered with $b=b_1-b_2$ and $\mathbf{w}=\mathbf{w}_1-\mathbf{w}_2$. Again, ρ is a functional margin parameter and $\xi_i,\ i=1,\ldots,N$ are slack variables with respect to the margin. The first term in the above expression is the regularizer. See Fig. 1 (c). We add two more constraints, $\sum_{c=1}^{C} b_c = 0$ and $\bar{\mathbf{w}} = \frac{1}{C}\sum_{c=1}^{C}\mathbf{w}_c$. The first is necessary to avoid the trivial solution $\mathbf{w}_c = \bar{\mathbf{w}} = \mathbf{0}$ with $b_c = \rho \to +\infty$ (in other words, to ensure effective margin maximization). The second constraint specifies the role of $\bar{\mathbf{w}}$ as the arithmetic mean of $\mathbf{w}_c,\ c=1,\ldots,C$. This allows for two different interpretations of constraint Eq. (5): (a) While maximizing the functional margin ρ, the discriminant functions $f_c(\cdot)$ must obey

$$f_{y_i}(\mathbf{x}_i) = (\mathbf{w}_{y_i}-\bar{\mathbf{w}})^\top \mathbf{x}_i + b_{y_i} \geq \rho - \xi_i$$

and (b) while maximizing the functional margin ρ, the discriminant functions $g_c(\cdot)$ must obey

$$g_{y_i}(\mathbf{x}_i) = \mathbf{w}_{y_i}^\top \mathbf{x}_i \geq \bar{\mathbf{w}}^\top \mathbf{x}_i + \rho - \xi_i',$$

where $\xi_i' = \xi_i + b_{y_i}$ and where both b_{y_i} and ξ_i are defined with respect to the $\mathbf{w}_{y_i}-\bar{\mathbf{w}}$ weight vector. Interestingly, this means that during training $g_{y_i}(\cdot)$ *must classify each data point* $\mathbf{x}_i$ *greater than the mean classifier* $\bar{\mathbf{w}}^\top\mathbf{x}_i = \frac{1}{C}\sum_{c=1}^{C}\mathbf{w}_c^\top\mathbf{x}_i$ *by a margin* $\rho-\xi_i'$. We formulate the Lagrangian as

$$\begin{aligned} \mathcal{L} \quad = \quad & \frac{1}{2}\sum_{c=1}^{C}\frac{1}{C}\|\mathbf{w}_c-\bar{\mathbf{w}}\|^2 - C\rho + \mu\sum_i \xi_i - \sum_{i=1}^{N}\delta_i\xi_i \\ & + \sum_{i=1}^{N}\alpha_i\left[\rho - \xi_i - (\mathbf{w}_{y_i}-\bar{\mathbf{w}})^\top\mathbf{x}_i - b_{y_i}\right] + \left\langle \boldsymbol{\gamma}, \frac{1}{C}\sum_{c=1}^{C}\mathbf{w}_c - \bar{\mathbf{w}}\right\rangle + \epsilon\left(\sum_{c=1}^{C} b_c\right). \end{aligned}$$

When dualizing, we readily obtain e.g. $\mathbf{w}_t = \sum_{i:y_i=t}\alpha_i\mathbf{x}_i$, $\bar{\mathbf{w}} = \frac{1}{C}\sum_{i=1}^{N}\alpha_i\mathbf{x}_i$ and $\sum_{i:y_i=t}\alpha_i = 1$.

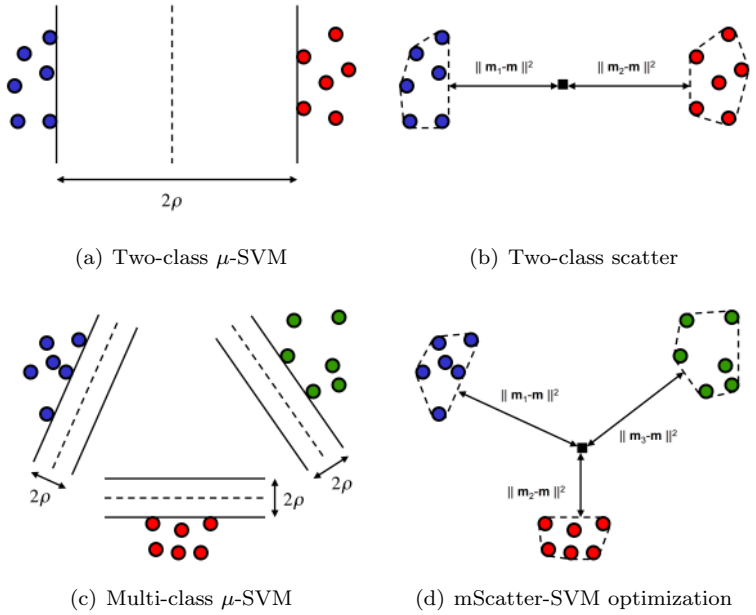
2.3 Dual Scatter Formulation of μ-SVM

Inserting $\mathbf{m}_c = \sum_{i:y_i=c}\alpha_i\mathbf{x}_i,\ c=1,\ldots,C$ and their arithmetic mean $\bar{\mathbf{m}} = \frac{1}{C}\sum_{c=1}^{C}\mathbf{m}_c$, we obtain the dual optimization problem

$$\min_{\boldsymbol{\alpha}} \frac{1}{2}\sum_{c=1}^{C}\frac{1}{C}\left\|\mathbf{m}_c - \bar{\mathbf{m}}\right\|^2.$$

Observe that the cost function to be optimized is proportional to the between-class scatter for multiple classes under the assumption of equal class priors, where scatter is measured with respect to the weighted class mean vectors $\mathbf{m}_c,\ c=1,\ldots,C$ and the overall arithmetic mean $\mathbf{m}$. See Fig. 1 (d). Furthermore, this may be expressed as a quadratic form $\frac{1}{2}\boldsymbol{\alpha}^\top\boldsymbol{\mathcal{K}}\boldsymbol{\alpha}$, where

$$\boldsymbol{\mathcal{K}} = \frac{1}{C}\begin{bmatrix} (C-1)\mathbf{K}_{11} & -\mathbf{K}_{12} & \ldots & -\mathbf{K}_{1C} \\ -\mathbf{K}_{21} & (C-1)\mathbf{K}_{22} & \ldots & -\mathbf{K}_{2C} \\ \vdots & \vdots & \ddots & \vdots \\ -\mathbf{K}_{C1} & -\mathbf{K}_{C2} & \ldots & (C-1)\mathbf{K}_{CC} \end{bmatrix},$$



(a) Two-class μ-SVM (b) Two-class scatter

(c) Multi-class μ-SVM (d) mScatter-SVM optimization

Figure 1: (a) The two-class μ-SVM maximizes a hyperplane margin. (We only illustrate hard margin case.) (b) The two-class μ-SVM minimizes between-class scatter. (c) Our multi-class μ-SVM maximizes the margin of several hyperplanes simultaneously. (d) The mScatter-SVM minimizes the between-class scatter wrt. weighted class mean vectors and a global reference point which is the arithmetic mean of the class means.

and $\mathbf{K}_{cc'}$, are inner-product matrices within and between classes. Finally, we get the optimization problem for what we will call the mScatter[1]-SVM as

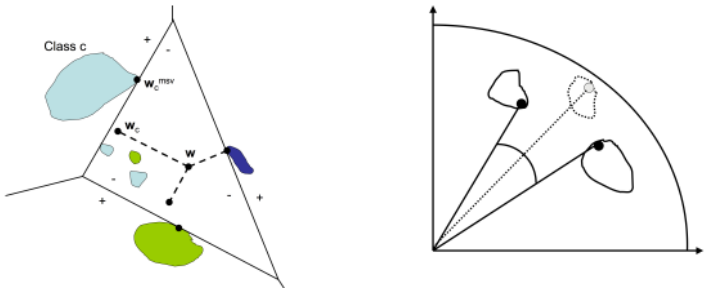
$$\begin{aligned} \min_{\boldsymbol{\alpha}} \quad & \frac{1}{2}\boldsymbol{\alpha}^\top \mathcal{K}\boldsymbol{\alpha} \\ \text{s.t} \quad & \boldsymbol{\alpha}_c^\top \mathbf{1} = 1, \quad c = 1, \ldots, C \quad \text{and} \quad 0 \leq \alpha_i \leq \mu \quad \forall i, \end{aligned} \tag{6}$$

where $\boldsymbol{\alpha}_c^\top \mathbf{1} = \sum_{i:y_i=c} \alpha_i$. These constraints also enforce $\mu \geq 1/N_{min}$ where N_{min} is the number of points in the smallest class. At the same time, $\mu \leq 1$ since $\boldsymbol{\alpha}_c^\top \mathbf{1} = 1,\ c = 1, \ldots, C$.

The matrix $\mathcal{K}$ is $(N \times N)$, positive semi-definite and symmetric. The optimization problem only involves N variables and $N + C$ simple constraints and the cost function is convex. This problem is well suited for quadratic programming packages like e.g. Mosek and fast SMO implementations are possible (see Section 4). Notice that the two-class μ-SVM dual appears as a special case.

Interestingly, we note that [NCAC08] reached an optimization problem of similar form, however, from a starting point of distances between pairwise convex hulls. The weighted mean vector perspective and our primal formulation are totally different, as well as the incorporation of slack variables. This allows for a more complete study of the optimization problem from the viewpoint of μ-SVMs, and opens up the possibility of a fast implementation (see Section 4). Furthermore, our testing rule based on geometrical distances follow directly from our derivation of the optimization problem (see Section 3), and leads to better classification results as mentioned in Section 5.

[1] margin scatter



(a) (b)

Figure 2: (a) Illustration of Test Rule 2 which is a direct extension of the μ-SVM test rule in the two-class case, using geometric distances. (b) The special geometry induced by the RBF kernel translates into a measure of angles, hence favoring Test Rule 1 which is actually used in the experiments.

As a further note, we would like to emphasize that the above theory was derived for cases where the classes are distributed spherically around a centroid of the weighed class mean vectors. However, such benign class distributions are of course not to be expected in real data. Obviously, by mapping to feature space the purely inner-product based mScatter-SVM algorithm can be reformulated with kernels; therefore, in the following, we will assume that a kernel function $k(\cdot,\cdot) = \langle \phi(\cdot), \phi(\cdot) \rangle$ is employed, where $\phi(\cdot)$ denotes the non-linear mapping (cf. [SMB$^+$99, SSM98]).

3 Test Rules for Classification

Based on the discriminant functions $g_c(\cdot)$ and $f_c(\cdot)$ two different testing rules for classification based on geometric distances (when testing based on weight vectors of different lengths, geometric distances should be used) are possible: *Test Rule 1* that is based on the angular spread with respect to its class representative as

$$\mathbf{x}_t \rightarrow \omega_c : \quad \arg\max_c \quad \frac{\mathbf{m}_c^\top}{\|\mathbf{m}_c\|}\mathbf{x}_t$$

and *Test Rule 2* which we mention here for completeness

$$\mathbf{x}_t \rightarrow \omega_c : \quad \arg\max_c \quad \frac{1}{\|\mathbf{m}_c - \bar{\mathbf{m}}\|}\left[(\mathbf{m}_c - \bar{\mathbf{m}})^\top \mathbf{x}_t - (\mathbf{m}_c - \bar{\mathbf{m}})^\top \mathbf{m}_c^{msv}\right],$$

where $\mathbf{m}_c$ and $\mathbf{m}_c^{msv}$ are the class representatives and $\bar{\mathbf{m}} = \frac{1}{C}\sum_{c=1}^{C} \mathbf{m}_c$. Notice that this test rule reduces to Eq. (3) in the two-class case. Figure 2 illustrates both rules.

We use in this paper the most well-known and widely used non-linear kernel function, namely the RBF kernel. The RBF kernel induces a special geometry since it maps data points to a quarter sphere in the kernel feature space. All data points $\phi(\mathbf{x}_i),\ i = 1, \ldots, N$ are therefore of unit (constant) length, and the evaluation of Euclidean distances between points reduces to a measure of angles. For discrimination we may make use of this property by employing the angular discriminant function represented by *Test Rule 1*, which in our experience gives the best results.

4 A Fast Implementation

With ever increasing data sets, solving complex mathematical programs such as the one in Eq. (6) with off-the-shelf solvers quickly becomes impractical. Therefore dedicated efficient optimizers have been developed for the well-understood quadratical programs (QPs) that emerge from binary SVMs. Here we describe the implementation of such a high-performace solver for the mScatter training problem.

Many efficient SVM training methods rely on decomposition techniques; examples are chunking [Joa99] and Sequential Minimal Optimization (SMO, e.g. [FCL05]). The idea of decomposition is to iteratively improve a solution candidate by solving a sequence of subproblems: to optimize a small number of variables (the so-called working set) while, for that moment, freezing all others. In chunking, the subproblems typically contain a few dozen variables and may be solved with off-the-shelf optimizers. In SMO, the working sets consist of exactly two variables, such that analytical optimization is possible.

Apart from the working set size, the critical design choice is the selection of the variables for the sub-problem: the convergence speed for the global optimization depends on the amount of progress that the sub-problems allow for. To make SMO efficient, clever selection strategies for the two variables α_i^k, α_j^k to be optimized at iteration k are required. A proven strategy based on second order information is implemented in LibSVM [CL01], both for C-SVMs and ν-SVMs. We exploit the fact that our problem (6) is a close relative of the ν-SVM dual, and extend the SMO implementation of LibSVM to be suitable for the mScatter-SVM.

To do so, we start from the 2-class ν-formulation as stated in Eq. (28) in [FCL05] and repeated here

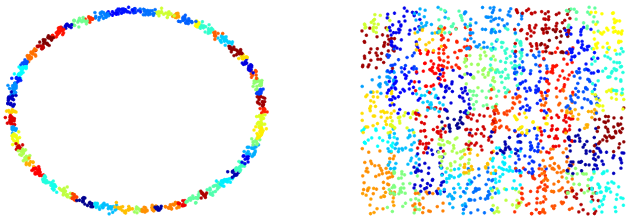
$$\begin{aligned} \min_{\boldsymbol{\alpha'}} \quad & \frac{1}{2}\boldsymbol{\alpha'}^\top \boldsymbol{\mathcal{K}} \boldsymbol{\alpha'} \\ \text{s.t} \quad & \boldsymbol{y}^\top \boldsymbol{\alpha'} = 0, \quad \boldsymbol{\alpha'}^\top \mathbf{1} = \nu, \quad 0 \leq \boldsymbol{\alpha'} \leq \frac{1}{N}\mathbf{1}. \end{aligned}$$

We first notice that the two equality constraints can be expressed by class-wise total weight mass conditions: $\boldsymbol{\alpha'_1}^\top \mathbf{1} = \boldsymbol{\alpha'_{-1}}^\top \mathbf{1} = \nu/2$. Due to these equality constraints, reasonable subproblems require $y_i = y_j$; otherwise, neither α'_i nor α'_j could be changed. Consequently this constraint is implemented by the selection strategy and a proper choice of the initial solution candiate. Note that feasible initial points also require $\nu \leq C \cdot N_{min}/N$. To recover problem (6), we need to perform a variable transformation $\boldsymbol{\alpha'} \mapsto C\frac{\boldsymbol{\alpha}}{\mu \cdot N}$ combined with the choice $\nu = C/(\mu \cdot N)$.

For 2-class problems, LibSVM's working set selection strategy for ν-SVMs (cf. WSS 5 in [FCL05]) traverses the active set twice and thus requires an effort of $\mathcal{O}(2N + 2T)$, where T is the time to compute a kernel row. A straightforward generalisation traverses the active set for each of the C classes leading to an effort of $\mathcal{O}(CN + CT)$ which is what we used throughout experiments. However, when ordering examples, such that $y_i \leq y_j$ for $i < j$ and by creating C arrays to hold the maximum class-wise gradient etc. the computational complexity can be further reduced to $\mathcal{O}(N + C \cdot T + C)$. An efficient implementation will be made available upon acceptance of the paper.

5 Experimental Results

The experiments essentially establish two main points. First, the presented method generalizes similarly well as existing multiclass strategies, like one vs. one and one vs. rest. Second, there is a considerable speed-up to be gained by our novel mScatter-SVM algorithm.



(a) Circle (b) Checker

Figure 3: Visualization of toy data sets: (a) 100 class circle data set (b) 100 class checker data set

Dataset	Checker-Board			Circle			Method
Error [%]	37	46	45	4	15	16	$k-$NN
	46	48	49	24	20	22	OVR
	37	40	40	6	15	16	MS-SVM
Time (s)	0.00	0.19	20.77	0.00	0.04	5.39	$k-$NN
	0.01	1.62	165.27	0.01	3.73	1549.57	OVR
	0.01	1.06	138.38	0.00	0.14	55.61	MS-SVM
#Classes	10	100	1000	10	100	1000	
N	200	2000	20000	100	1000	10000	

Table 1: Time comparison of the proposed mScatter SVM (MS-SVM) with respect to the traditional $k-$NN and one-against-rest (OVR) SVM training strategies. Times include training on $N/2$ examples using the optimal parameter settings and testing on the remaining $N/2$ examples. Compared to the OVR-SVM, the proposed MS-SVM achieves faster training and testing times for all experiments (speedup factor up to 27) while achieving better accuracies. Note that these data sets contain a fixed number k of examples per class and thus $k-$NN (given here for reference) achieves the lowest classification error.

To systematically evaluate the properties of our algorithms we first conduct a toy experiment on two artificially generated data set types: 2d-checker-boards with slightly overlapping fields and one class for each field, and 2d-data sets of Gaussians evenly distributed on a circle. These data sets are illustrated in Fig. 3. In these data sets both the number of classes and the number of data points are increased (cf. Table 1). For the checker (circle) data set we generated 20 (10) points per class and split the data set evenly into training and validation set (with an equal number of points in each class). We then apply k nearest neighbor ($k-$NN), one vs. rest C-SVM (OVR), and our proposed mScatter-SVM. We perform model selection over several parameters on the validation set[2]. We then measure time (training+prediction) and classification error rates (in percent, rounded) for the *best* performing model.

As is clearly seen from Table 1, our mScatter-SVM is much faster than the classical multiclass one vs. rest (OVR) strategy[3]. In addition, the mScatter-SVM obtains much lower error rates compared to OVR. Commonly, OVR or OVA deliver highest prediction accuracies. However, this requires model parameter tuning in a class-specific way. Ultimately they fail to scale due to the more involved optimization problem. These experiments illustrate in particular the speed-up properties of our algorithm while maintaining good generalization.

[2]For $k-$NN, $k \in \{1,3,5,7,9\}$, for SVMs RBF-kernels of width $\sigma^2 \in \{0.1,1,5\}$, $SVM_C \in \{1,10,100\}$ and $\nu \in \{C/N, 0.5, 0.999\}$.

[3]We did not perform one vs. one (OVO) training, as is not tractable when the number of classes is large.

	# training data	# testing data	# class	# attributes
Iris	150		3	4
Wine	178		3	13
Glass	214		6	13
Vowel	528		11	10
Segment	2310		7	19
Dna	2000	1186	3	180
Satimage	4435	2000	6	36

Table 2: Properties of multi-class benchmark data sets used in main study of mScatter-SVM generalization ability.

	Test Rule 1	[Vap98, WW99]	One vs. one
Iris	97.33	97.33	97.33
Wine	98.33	98.88	99.44
Glass	71.43	71.03	71.50
Vowel	99.43	98.49	99.05
Segment	97.40	97.58	97.40
Dna	98.39	95.62	95.45
Satimage	90.65	91.25	91.30

Table 3: Classification results using RBF kernel and 10-fold cross-validation over 75 σ-parameters and 26 μ-parameters on several Real-world data sets.

Having established that our method is fast, we turn in our second experiment to a more thorough evaluation of the generalization ability of mScatter-SVM. To this end, we study a number of multi-class benchmark data sets from the LIBSVM web-site [CL01]. These data sets are pre-processed such that each attribute is linearly scaled to $[-1, 1]$ (except `Dna`). The properties of these data sets are listed in Table 2.

For data sets `Iris`, `Wine`, `Glass`, `Vowel` and `Segment` we perform 10-fold cross-validation and report the best average success rate in percent. We cross-validate over 26 μ-parameters and 75-σ parameters, where σ governs the width of the RBF kernel. We evaluate this many parameters because we can afford to with our fast method. For data sets `Dna` and `Satimage` training and validation sets are available, in addition to the test set. Here, we use the validation set to determine the best combination of kernel size and μ. We then test using these parameters based on the same training data. Some of the same data sets were used in a large comparison study of several multi-class SVM approaches in [HL02]. As a generic reference point we show some of the results obtained in that study for joint optimization methods [Vap98, WW99] as well as the one vs. one approach. The mScatter-SVM performs comparable for most data sets and even better in some cases (notably on `Dna`), while for a couple of data sets the method performs only slightly worse. Overall, mScatter-SVM clearly manages to maintain a good generalization ability compared to the other much more computationally demanding methods. We mention that we have implemented the heuristically obtained testing rule in [NCAC08], obtaining quite poor results in several cases (e.g. `Glass`: 63.33, `Segment`: 94.91).

6 Conclusion

Many problems in science and technology are intrinsically multi-class problems with a large number of categories to be distinguished. While there are established excellent methods for two-class problems, it is a challenge to extend them towards a large number of classes in a scalable way. The straight forward extension of SVMs from two to multiple classes – namely by training all $C(C-1)/2$ one vs. one class combinations – will ultimately lead to infeasible computing times for a large number of classes. Our novel mScatter-SVM algorithm aims to alleviate this limiting factor while at the same time maintaining comparible generalization performance; it thus yields a huge speedup. The algorithmic insight to achieve this comes from a reformulation of the classical SVM in terms of class means and the scatter involved. In doing so, the arithmetic mean over all class means becomes a pivotal quantity and helps to maintain a tractable number of constraints. Intuitively, the resulting mathematical program (cf. Eq. (5)) enforces every sample to be *by a margin more similar to it's class mean than to the overall mean*. The assumptions under which our method will work well also becomes transparent, namely, there should be a certain *homogeneity* among the classes wrt. noise, outliers and regularization treatment. The reference to a global mean introduces a global regularization or stiffness of the model. There are, of course, learning problems that may require a fine grained class-wise regularization that is systematically only available by the one vs. one multi-class approach. Note however that there is a sufficiently vast body of multi-class problems that match our assumptions above.

So far, we have used RBF kernels in combination with an angle based classification rule for deciding between multiple classes. We aim to further study which classification rule is suitable for which kernel transformation. Future work will also exploit the mScatter-SVM algorithm for image annotation, in subphoneme classification for speech recognition, and in computational biology.

References

[BVB95] V. Blanz, V. Vapnik, and C. Burges. Multiclass Discrimination with an Extended Support Vector Machine. Talk at AT&T Bell Labs, 1995.

[CB99] D. J. Crisp and C. J. C. Burges. A Geometric Interpretation of ν-SVM Classifiers. In *Advances in Neural Information Processing Systems, 11*, pages 244–250, MIT Press, Cambridge, 1999.

[CL01] C.-C. Chang and C.-J. Lin. *LIBSVM: A Library for Support Vector Machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[CS01] K. Crammer and Y. Singer. On the Algorithmic Implementation of Multiclass Kernel-based Vector Machines. *Journal of Machine Learning Research*, 2:265–292, 2001.

[DHS01] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. John Wiley & Sons, New York, second edition, 2001.

[FCL05] R.-E. Fan, P.-H. Chen, and C.-J. Lin. Working Set Selection Using Second Order Information for Training Support Vector Machines. *Journal of Machine Learning Research*, 6:1889–1918, 2005.

[FH02] V. Franc and V. Hlavac. Multi-class Support Vector Machine. In *Proceedings of International Conference on Pattern Recognition*, pages 236–239, Quebec, Canada, 11-15 August, 2002.

[HL02] C.-W. Hsu and C.-J. Lin. A Comparison of Methods for Multiclass Support Vector Machines. *IEEE Transactions on Neural Networks*, 13(2):415–425, 2002.

[Joa99] T. Joachims. Making Large–Scale SVM Learning Practical. In B. Schölkopf, C.J.C. Burges, and A.J. Smola, editors, *Advances in Kernel Methods — Support Vector Learning*, pages 169–184, Cambridge, MA, USA, 1999. MIT Press.

[LLW04] Y. Lee, Y. Lin, and G. Wahba. Multicategory Support Vector Machines: Theory and Application for Classification of Microarray Data and Satellite Radiance Data. *Journal of the American Statistical Association*, 99(465):67–81, 2004.

[MT06] M. Mavroforakis and S. Theodoridis. A Geometric Approach to Support Vector Machine (SVM) Classification. *IEEE Transactions on Neural Networks*, 17(3):671–682, 2006.

[NCAC08] R. Nanculef, C. Concha, H. Allende, and D. Candel. Multicategory SVMs by Minimizing the Distances Among Convex-Hull Prototypes. In *Proceedings of International Conference on Hybrid Intelligent Systems*, pages 423–429, Barcelona, Spain, September 10-12, 2008.

[SMB+99] B. Schölkopf, S. Mika, C. J. C. Burges, P. Knirsch, K.-R. Müller, G. Rätsch, and A. J. Smola. Input Space Versus Feature Space in Kernel-Based Methods. *IEEE Transactions on Neural Networks*, 10(5):1299–1319, 1999.

[SSM98] B. Schölkopf, A. J. Smola, and K.-R. Müller. Nonlinear Component Analysis as a Kernel Eigenvalue Problem. *Neural Computation*, 10:1299–1319, 1998.

[SST05] S. Szedmak and J. Shawe-Taylor. Multiclass Learning at One Class Complexity. Technical report, School of Electronics and Computer Science, University of Southampton, UK., 2005.

[SSWB00] B. Schölkopf, A. J. Smola, R. C. Williamson, and P. L. Bartlett. New Support Vector Algorithms. *Neural Computation*, 12(5):1207–1245, 2000.

[Vap98] V. N. Vapnik. *Statistical Learning Theory*. John Wiley & Sons, New York, 1998.

[WW99] J. Weston and C. Watkins. Support Vector Machines for Multi Class Pattern Recognition. In *Proceedings of European Symposium on Artificial Neural Networks*, pages 219–224, Bruges, Belgium, April 21-23, 1999.