A NEW SCATTER-BASED MULTI-CLASS SUPPORT VECTOR MACHINE

Robert Jenssen[1], *Marius Kloft*[2,3], *Sören Sonnenburg*[2,4], *Alexander Zien*[5] *and Klaus-Robert Müller*[2,6,7]

[1] Department of Physics and Technology, University of Tromsø, Norway
[2]Machine Learning Laboratory, Berlin Institute of Technology, Berlin, Germany
[3]Computer Science Division, University of California at Berkeley, USA
[4]Friedrich Miescher Institute of the Max Planck Society, Tübingen, Germany
[5]Life Biosystems GmbH, Heidelberg, Germany
[6]Bernstein Center for Computational Neuroscience, Berlin, Germany
[7]Institute for Pure and Applied Mathematics (IPAM), University of California at Los Angeles, USA

ABSTRACT

We provide a novel interpretation of the dual of support vector machines (SVMs) in terms of scatter with respect to class prototypes and their mean. As a key contribution, we extend this framework to multiple classes, providing a new joint Scatter SVM algorithm, at the level of its binary counterpart in the number of optimization variables. We identify the associated primal problem and develop a fast chunking-based optimizer. Promising results are reported, also compared to the state-of-the-art, at lower computational complexity.

Index Terms— μ-SVM, scatter, multi-class

1. INTRODUCTION

The support support vector machine (SVM) [1] is normally defined in terms of a classification hyperplane between two classes, leading to the primal optimization problem. The primal is most often translated into a *dual* optimization problem in n variables, where n is the number of data points. For multi-class problems, the SVM is often executed in a one-vs.-one (OVO) or one-vs.-rest (OVR) mode. Some efforts have been made to develop *joint* multi-class SVMs [2, 3, 4, 5, 6], by extending the primal of binary SVMs. This has the effect of increasing the number of optimization variables in the dual, typically to $n \times C$, where C is the number of classes, often under a huge amount of constraints. This limits practical usability, due to increased computational complexity.

Even though the actual optimization is carried out in the dual space, little has been done to analyze properties of SVMs in view of the dual. One exeption is [7], where SVMs are interpreted in terms of information theoretic learning. Another exception is the convex hull view [8]. This alternative view yields additional insight about the algorithm and has also lead to algorithmic improvements [9]. An extension from the binary case to the multi-class case have furthermore been proposed in [10]. The dual view therefore in this case provides a richer theory by complementing the primal view.

In this paper, we contribute a new view of the dual of binary SVMs, concentrating on the so-called μ-SVM [11], as a minimization of *between-class scatter* with respect to the class prototypes and their arithmetic mean. Importantly, we note that scatter is inherently a multi-class quantity, suggesting therefore a natural extension of the μ-SVM to operate *jointly* on C classes. Interestingly, this key contribution, fittingly referred to as Scatter SVM, does not introduce more variables to be optimized than the number n of training examples, while keeping the number of constraints low. This is a major computational saving compared to the aforementined previous joint SVM approaches.

A special case of the optimization problem developed in this paper turns out to resemble [10], although from a completely different starting point. This work surpasses [10] in several aspects, by developing a complete dual-primal theory, opening up different opportunities wrt. the loss function used and defining the actual score function to use in testing, and by developing an efficient solver based on sequential minimal and chunking optimization.

This paper is organized as follows. In Section 2, the dual of μ-SVMs is analyzed in terms of scatter and extended to multiple classes. The primal view is discussed in Sec. 3, experiments are reported in Sec. 4, and the paper is concluded by Sec. 5.

2. SCATTER SVM

SVMs are normally defined in terms of a class-separating score function, or hyperplane, $f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + b$,which is determined in such a way that the margin of the hyperplane is maximized. Let a labeled sample be given by

Financed in part by the Research Council of Norway (171125/V30), by the German Bundesministerium für Bildung und Forschung (REMIND FKZ 01-IS07007A), by the FP7-ICT PASCAL2 Network of Excellence (ICT-216886), by the German Research Foundation (MU 987/6-1 and RA 1894/1-1) and by the German Academic Exchange Service (Kloft).

$\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1...,n}$, where each example $\mathbf{x}_i$ is drawn from a domain $\mathcal{X} \in \mathcal{R}^d$ and $y \in \{1, 2\}$. The μ-SVM [11] optimization problem is given by

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

Here, 2ρ is the functional margin of the hyperplane, and the parameter μ controls the emphasis on the minimization of margin violations, quantified by the slack variables ξ_i.

By introducing Lagrange multipliers $\alpha_i,\ i = 1, \ldots, n$, collected in the $(n \times 1)$ vector $\boldsymbol{\alpha} = \left[\boldsymbol{\alpha}_1^\top\ \boldsymbol{\alpha}_2^\top\right]^\top$, where $\boldsymbol{\alpha}_c$ stores $\{\alpha_i\}_{i:y_i=c}$, $c = 1, 2$, the *dual* optimization problem becomes

$$\begin{array}{ll} \min\limits_{\boldsymbol{\alpha}} & \frac{1}{2}\boldsymbol{\alpha}^\top\mathcal{K}\boldsymbol{\alpha} \\ \text{s.t.} & \mathbf{0} \leq \boldsymbol{\alpha} \leq \mu\mathbf{1} \\ & \boldsymbol{\alpha}^\top\mathbf{1} = 2 \\ & \boldsymbol{\alpha}_1^\top\mathbf{1} = \boldsymbol{\alpha}_2^\top\mathbf{1}, \end{array} \tag{2}$$

where $\mathbf{1}$ is an all ones vector[1] and

$$\mathcal{K} = \begin{bmatrix} \mathbf{K}_{11} & -\mathbf{K}_{12} \\ -\mathbf{K}_{21} & \mathbf{K}_{22} \end{bmatrix}.$$

The subscripts indicate the two classes and $\mathbf{K}_{cc'}$ are inner-product matrices within and between classes. Obviously, the constraints in Eq. (2) enforce $\boldsymbol{\alpha}_c^\top\mathbf{1} = 1,\ c = 1, 2$.

The optimization determines $\mathbf{w}$ explicitly as

$$\mathbf{w} = \sum_{i:y_i=1} \alpha_i\mathbf{x}_i - \sum_{i:y_i=2} \alpha_i\mathbf{x}_i, \tag{3}$$

where the non-zero α_i's correspond to the support vectors. The bias b is implicitly determined via the Karush-Kuhn-Tucker (KKT) conditions. If the bias b is omitted, the last constraint in Eq. (2) disappears. This is a mild restriction for high dimensional spaces, since it amounts to reducing the number of degrees of freedom by one (see also [12]).

Let $\mathbf{m}_c = \sum_{i:y_i=c} \alpha_i\mathbf{x}_i, \quad c \in \{1, 2\}$, be a class prototype, where the weights α_i determine the properties of the prototype. Observe that we may express the μ-SVM hyperplane weight vector, given by Eq. (3), in terms of prototypes as $\mathbf{w} = \mathbf{m}_1 - \mathbf{m}_2$. It follows that $\|\mathbf{m}_1 - \mathbf{m}_2\|^2 = \boldsymbol{\alpha}^\top\mathcal{K}\boldsymbol{\alpha}$, and we may by Eq. (2) conclude that the μ-SVM in the *dual* corresponds to minimizing the squared Euclidean distance between the class prototypes $\mathbf{m}_1$ and $\mathbf{m}_2$. In terms of the class prototypes, the score function is expressed as $f(\mathbf{x}) = (\mathbf{m}_1 - \mathbf{m}_2)^\top\mathbf{x} + b$ if the bias is included in the primal, or just $f(\mathbf{x}) = (\mathbf{m}_1 - \mathbf{m}_2)^\top\mathbf{x}$, if not.

Interestingly, by introducing the arithmetic mean $\bar{\mathbf{m}} = \frac{1}{2}(\mathbf{m}_1 + \mathbf{m}_2)$ of the prototypes into the picture, the quantity $\sum_{c=1}^2 \|\mathbf{m}_c - \bar{\mathbf{m}}\|^2$ equals $\|\mathbf{m}_1 - \mathbf{m}_2\|^2$ up to a constant, and thus also equals $\boldsymbol{\alpha}^\top\mathcal{K}\boldsymbol{\alpha}$ up to a constant. This provides a new geometrical way of viewing the dual of the μ-SVM, which may be related to the *multi-class* notion of between-class *scatter* in pattern recognition. Scatter is normally defined as $\sum_{c=1}^C P_c\|\mathbf{v}_c - \bar{\mathbf{v}}\|^2$ [13], with respect to class means $\mathbf{v}_c = \sum_{i:y_i=c} \frac{1}{n_c}\mathbf{x}_i,\ c = 1, \ldots, C$, and the global mean $\bar{\mathbf{v}} = \sum_{c=1}^C P_c\mathbf{v}_c$, where P_c is the prior class probability of the c'th class. Hence, for $C = 2$, by introducing the weights α_i for each data point $\mathbf{x}_i$ and by defining the scatter with respect to the class prototypes $\mathbf{m}_c,\ c = 1, 2$, and their arithmetic mean under the equal class probability assumption, the cost function $\sum_{c=1}^2 \|\mathbf{m}_c - \bar{\mathbf{m}}\|^2$ is obtained.

A direct extension of the scatter-based view of the dual to C classes is proposed here as

$$\begin{array}{ll} \min\limits_{\boldsymbol{\alpha}} & \frac{1}{2}\sum_{c=1}^C \|\mathbf{m}_c - \bar{\mathbf{m}}\|^2 \\ \text{s.t.} & \mathbf{0} \leq \boldsymbol{\alpha} \leq \mu\mathbf{1} \\ & \boldsymbol{\alpha}^\top\mathbf{1} = C \\ & \boldsymbol{\alpha}_c^\top\mathbf{1} = 1,\ c = 1, \ldots, C \quad (if\ bias), \end{array} \tag{4}$$

for $\mathbf{m}_c = \sum_{i:y_i=c} \alpha_i\mathbf{x}_i$, $\bar{\mathbf{m}} = \frac{1}{C}\sum_{c=1}^C \mathbf{m}_c$ and weights $\boldsymbol{\alpha} = \left[\boldsymbol{\alpha}_1^\top\ \ldots\ \boldsymbol{\alpha}_C^\top\right]^\top$, where $\boldsymbol{\alpha}_c$ stores $\{\alpha_i\}_{i:y_i=c}$, $c = 1, \ldots, C$. This constitutes a direct extension of scatter to multiple classes. In this formulation, it is optional whether or not to include the last constraint, depending on the score function bias parameter (primal view), discussed shortly.

It is easily shown that $\sum_{c=1}^C \|\mathbf{m}_c - \bar{\mathbf{m}}\|^2 = \boldsymbol{\alpha}^\top\mathcal{K}\boldsymbol{\alpha}$, up to a constant, where

$$\mathcal{K} = \begin{bmatrix} \gamma\mathbf{K}_{11} & -\mathbf{K}_{12} & \ldots & -\mathbf{K}_{1C} \\ -\mathbf{K}_{21} & \gamma\mathbf{K}_{22} & \ldots & -\mathbf{K}_{2C} \\ \vdots & \vdots & \ddots & \vdots \\ -\mathbf{K}_{C1} & -\mathbf{K}_{C2} & \ldots & \gamma\mathbf{K}_{CC} \end{bmatrix}, \tag{5}$$

$\gamma = C - 1$ and $\mathbf{K}_{cc'}$ are inner-product matrices within and between classes. Hence, the optimization problem Eq. (4) may also be expressed as

$$\begin{array}{ll} \min\limits_{\boldsymbol{\alpha}} & \frac{1}{2}\boldsymbol{\alpha}^\top\mathcal{K}\boldsymbol{\alpha} \\ \text{s.t.} & \mathbf{0} \leq \boldsymbol{\alpha} \leq \mu\mathbf{1} \\ & \boldsymbol{\alpha}^\top\mathbf{1} = C \\ & \boldsymbol{\alpha}_c^\top\mathbf{1} = 1,\ c = 1, \ldots, C \quad (if\ bias), \end{array} \tag{6}$$

The matrix $\mathcal{K}$ is $(n \times n)$ and positive semi-definite, and therefore leads to an optimization problem over a quadratic form (cf. Eq. (6)), which constitutes a convex cost function. The box constraints enforce $\mu \geq 1/N_{min}$ where N_{min} is the number of points in the smallest class. This *Scatter SVM* problem can be solved efficiently by quadratic programming. There are merely n variables to be optimized, as opposed to $n \times C$ variables for joint approaches like [3, 4]. With the bias included, there are $\mathcal{O}(n + C)$ simple constraints. This problem is basically equal to [10]. However, if the bias is omitted, there are even less constraints, only $\mathcal{O}(n + 1)$. This latter

[1] The length of $\mathbf{1}$ is given by the context.

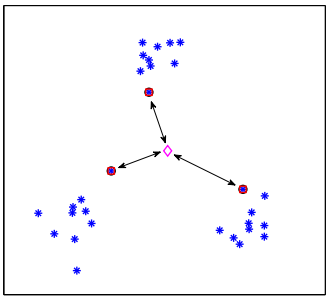


Fig. 1. The result of training Scatter SVM on three classes (toy data set).

optimization problem is the one we primarily focus on in the experiments in Section 4. We are thus faced with an optimization problem of much lower computational complexity than previous joint approaches.

In fact, Eq. (6) lends itself nicely to a solver based on sequential minimal optimization [14] or chunking optimization [15], respectively, depending on whether the bias is included or not. We have developed very efficient and dedicated solvers for each case, where in the with-bias mode, the algorithm is based on LIBSVM [16], and in the without-bias mode, the algorithm is based on SVMlight [15]. Details of these procedures are deferred to a longer paper. Both versions are implemented in the SHOGUN toolbox [17], publicly available for download at `http://www.shogun-toolbox.org/`. We will illustrate in Section 4 that the Scatter SVM provides a fast and computationally efficient joint approach.

Figure 1 shows the result of training Scatter SVM on a toy three-class data set. In this case, there is only one support vector for each class, thus acting as a class representative. The arrows indicate the minimized distances between class representatives and their geometric mean. Of course this data set has a "benign" structure, in that the classes are nicely distributed around a center point. It is obvious that one may construct cases where the reference to the mean of the class prototypes may be problematic. However, by mapping the data to a richer space, of higher dimensionality, such issues are avoided. For this reason, and also for increasing the probability of linearly separable classes, we in general employ the kernel induced non-linear mapping $\psi : \mathcal{X} \to \mathcal{H}$, to a Hilbert space $\mathcal{H}$ [18]. Kernel functions $k(\mathbf{x}, \mathbf{x}') = \langle \psi(\mathbf{x}), \psi(\mathbf{x}') \rangle_{\mathcal{H}}$ are thus utilized to compute inner products in $\mathcal{H}$.

3. A REGULARIZED RISK MINIMIZATION FRAMEWORK

For upcoming derivations, we focus on affine-linear models of the form $f_c(\mathbf{x}) = \mathbf{w}_c^\top \psi(\mathbf{x}) + b_c$. As discussed earlier, the bias parameter b_c may be removed in the derivations, which is a mild restriction for the high dimensional space $\mathcal{H}$ we consider. Let the goal be to find a hypothesis $f = (f_1, ..., f_C)$ that has low error on new and unseen data. Labels are predicted according to $c^* = \operatorname{argmax}_c f_c(\mathbf{x})$. Regularized risk minimization returns the minimizer f^*, given by $f^* = \min_f \Omega(f) + \mu \mathrm{R}_{\mathrm{emp}}(f)$, where the empirical risk $\mathrm{R}_{\mathrm{emp}}(f) = \frac{1}{n} \sum_{i=1}^n l\left[s(f, \mathbf{x}_i, y_i)\right]$, wrt. a convex loss function $l[\cdot]$, and where $\Omega(f)$ is the regularizer.

Commonly, $s(f, \mathbf{x}, y) = f_y(\mathbf{x}) - \operatorname{argmax}_{c \neq y} f_c(\mathbf{x})$, i.e.. loss will be defined wrt. $f_y(\mathbf{x})$ and the best model $f_{c \neq y}(\mathbf{x})$. However, such an approach gives rise to a large number of constraints [6, 19]. As a remedy to this issue, we propose as a different and novel requirement that a hypothesis should score better than an *average* hypothesis, that is

$$s(f, \mathbf{x}, y) = f_y(\mathbf{x}) - \frac{1}{C} \sum_{c=1}^{C} f_c(\mathbf{x}).$$

Including for the time being the bias, the average hypothesis thus becomes $\bar{f}(\mathbf{x}) = \bar{\mathbf{w}}^\top \mathbf{x} + \bar{b}$ and

$$s(f, \mathbf{x}, y) = (\mathbf{w}_y - \bar{\mathbf{w}})^\top \psi(\mathbf{x}) + b_y - \bar{b}, \tag{7}$$

where $\bar{\mathbf{w}} = \frac{1}{C} \sum_{c=1}^C \mathbf{w}_c$ and $\bar{b} = \frac{1}{C} \sum_{c=1}^C b_c$. Each hyperplane $\mathbf{w}_c - \bar{\mathbf{w}}$, $c = 1, \ldots, C$, is associated with a margin ρ. The following quadratic regularizer aims to penalize the norms of these hyperplanes while at the same time maximizing the margins

$$\Omega(f) = \frac{1}{2} \sum_c \|\mathbf{w}_c - \bar{\mathbf{w}}\|^2 - C\rho. \tag{8}$$

The regularized risk thus becomes

$$\frac{1}{2} \sum_{c=1}^{C} ||\mathbf{w}_c - \bar{\mathbf{w}}||^2 - C\rho + \mu \sum_i l\left[(\mathbf{w}_{y_i} - \bar{\mathbf{w}})^\top \psi(\mathbf{x}_i) + b_{y_i} - \bar{b}\right].$$

Expanding the loss terms into slack variables leads to the *primal optimization problem*

$$\begin{array}{ll} \min\limits_{\mathbf{w}_c, \mathbf{w}, b, \rho, \mathbf{t}} & \frac{1}{2} \sum_c ||\mathbf{w}_c - \bar{\mathbf{w}}||^2 - C\rho + \mu \sum_i l(t_i) \\ s.t. & \langle \mathbf{w}_{y_i} - \bar{\mathbf{w}}, \psi(\mathbf{x}_i) \rangle + b_{y_i} \geq \rho - t_i, \quad \forall i \\ & \bar{\mathbf{w}} = \frac{1}{C} \sum_{c=1}^C \mathbf{w}_c \\ & \bar{b} = \frac{1}{C} \sum_{c=1}^C b_c = 0. \end{array} \tag{9}$$

The condition $\bar{b} = 0$ is necessary in order to obtain the primal of the binary μ-SVM as a special case of Eq. (9) and to avoid the trivial solution $\mathbf{w}_c = \bar{\mathbf{w}} = \mathbf{0}$ with $b_c = \rho \to \infty$.

Optimization is often considerably easier in the dual space. As it will turn out, we can derive the dual problem of Eq. (9) without knowing the loss function l, instead it is sufficient to work with the Fenchel-Legendre dual $l^*(x) = \sup_t xt - l(t)$ (e.g. cf. [20, 21]. The approach taken

is first to formulate the Lagrangian of Eq. (9), identify the Lagrangian saddle point problem, for then to completely remove the dependency on the primal variables by inserting the Fenchel-Legendre dual. Due to space constraints, details of this derivation are deferred to a longer version of this paper. However, this yields $\mathbf{w}_c = \sum_{i:y_i=c} \alpha_i \psi(\mathbf{x}_i)$, $\forall c$, which is equal to the expression for the class representative $\mathbf{m}_c$ in $\mathcal{H}$. The generalized dual problem obtained is

$$\sup_{\boldsymbol{\alpha}} \; -\frac{1}{2}\boldsymbol{\alpha}^\top \mathcal{K}\boldsymbol{\alpha} - \mu \sum_i l^*(-\mu^{-1}\alpha_i), \qquad (10)$$

$\boldsymbol{\alpha} : \boldsymbol{\alpha}^\top \mathbf{1} = C,\ \boldsymbol{\alpha}_c^\top \mathbf{1} = 1,\ c = 1, \ldots, C$ (if bias) where l^* is the Fenchel-Legendre conjugate function, which we subsequently denote as *dual loss* of l.

This formulation admits several possible loss functions. Utilizing the *hinge loss* $l(t) = \max(0, 1-t)$ into Eq. (10), noting that the dual loss is $l^*(t) = t$ if $-1 \leq t \leq 0$ and ∞ elsewise (cf. Table 3 in [22]), we obtain the dual given in Eq. (6), where the last constraint only applies if the bias parameter is included in the primal formulation for the score functions. Interestingly, Eq. (10) shows that the utilization of different loss functions will produce different optimization problems. It is left to future work to investigate such issues more closely, but it illustrates some of the versatility of our approach.

4. EXPERIMENTS

The aim of the experimental section is to highlight properties of Scatter SVM in terms of sparsity, generalization ability and computational efficiency, by performing classification on some well-known benchmark data sets used in the literature (see e.g. [23, 6]).

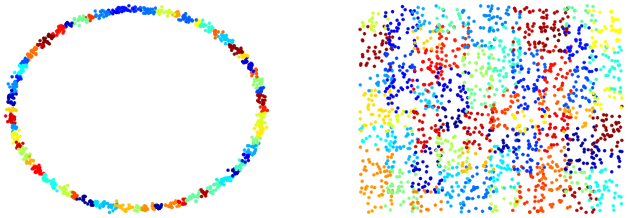
In all experiments, the RBF-kernel is adopted. This is the most widely used kernel function, given by

$$k(\mathbf{x}_i, \mathbf{x}_j) = e^{-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2}, \qquad (11)$$

where $\gamma = \frac{1}{2\sigma^2}$.

4.1. Experiment on Controlled Artificial Data

We first perform a "sanity" check of the Scatter SVM in a controlled scenario. Two data sets, often used in the literature (e.g. see [18]), are generated: 2d-checker-boards and 2d-Gaussians evenly distributed on a circle, illustrated in Fig. 2. 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). For this experiment, the Scatter SVM is executed in with-bias mode, and is contrasted to a one-vs.-rest (OVR) C-SVM. Both methods are based on LIBSVM as implemented in the SHOGUN toolbox. We perform model



(a) Circle (b) Checker

Fig. 2. Visualization of toy data sets: (a) 100 class circle data set (b) 100 class checker data set

USPS # SVs	0	6	9
Scatter SVM	53	47	31
OVR SVM	64 (8)	74 (14)	39 (17)

Table 2. USPS-based analysis of SVs and sparsity.

selection over the parameters on the validation set[2]. We then measure time (training + prediction) and classification error rates (in percent, rounded) for the *best* performing model.

With reference to Table 1, the execution times of Scatter SVM compare favorably to the OVR C-SVM, and in the most extreme case correspond to a speed up factor up to 27. Scatter SVM achieves a higher generalization ability than OVR. This might be because these data sets contain a fixed number of examples per class and are thus well suited for Scatter SVM. In other words, selecting this data may imply a bias towards Scatter SVM. However, these experiments illustrate in particular the speed-up properties of our algorithm while maintaining good generalization.

4.2. Case-Based Analysis of SVs and Sparsity

We perform an experiment in order to analyze the sparsity of Scatter SVM. A three-class data set is created by extracting the classes "0", "6" and "9" from the U.S. Postal Service (USPS) data set. We randomly select 1500 data points for training, and create a validation set for determining an appropriate kernel size. For this, and all remaining experiments, Scatter SVM operates in the without-bias mode based on a SHOGUN SVMlight implementation. The "μ" parameter in Scatter SVM translates into a "C" parameter, similar to the parameter in the OVR C-SVM. Both methods are now trained on eleven logarithmically C-parameters from 10^{-3} to 10^3. The validation procedure is performed over 76 kernel sizes $\gamma = 2^\kappa$ for κ between -10 and 5 in steps of 0.2 in Eq. (11). Scatter SVM and the OVR C-SVM obtain best validation results corresponding to 99.87 and 99.38 percent success rate, respectively. If $\alpha_i > 10^{-6}$ defines a SV, then Scatter SVM produces 131 SVs, corresponding to 8.7% of the train-

[2]For SVMs RBF-kernels of width $\sigma^2 \in \{0.1, 1, 5\}$, $SVM_C \in \{0.01, 0.1, 1, 10, 100\}$, and $\nu \in \{C/N, 0.5, 0.999\}$.

Dataset	Checker-Board			Circle				Method
Error [%]	35	49	50	22	24	22	21	OVR SVM
	24	40	41	14	17	18	17	Scatter SVM
Time (s)	0.05	1.77	102.15	0.02	3.51	1,229.30	197,236.71	OVR SVM
	0.06	1.59	85.21	0.01	2.11	46.27	42,401.26	Scatter SVM
#Classes	10	100	1,000	10	100	1,000	10,000	
N	200	2,000	20,000	100	1,000	10,000	100,000	

Table 1. Time comparison of the proposed Scatter SVM to the OVR LIBSVM training strategy.

ing data. The number of SVs for each class is shown in Table 2, together with the SV structure for the C-SVM. The number in parenthesis indicate the number of unique SVs of that class obtained in the "rest" part of the training. The number of all unique SVs is 216 corresponding to 14.4% of the training data. These experiments show that Scatter SVM may perform on par with a OVR C-SVM with respect to the sparsity of the solution. This we consider encouraging.

4.3. Generalization Ability on Benchmark Data Sets

To investigate further the generalization ability of Scatter SVM, we perform classification experiments on some well-known benchmark multi-class data sets commonly encountered in the literature (see e.g. [23, 6, 3]). The data sets are listed in Table 3. For those cases where specific test data sets are missing, we perform 10-fold cross-validation over the parameters and report the best result. If a test set is available, we simply report the best result over all combinations of parameters. The data sets are obtained from the LIBSVM web-site[3], (except MNIST) pre-processed such that all attributes are in the range $[-1, 1]$. The MNIST data[4] is normalized to unit length.

In this experiment, the Scatter SVM is contrasted to OVR C-SVM, one-vs.-one (OVO) C-SVM and Crammer and Singer's (CS) [6] multi-class SVM. All methods are trained for the same set of parameters and kernel sizes as in the previous section. The results, shown in Table 3, indicate that Scatter SVM has been able to generalize well, and to obtain classification results which are comparable to these state-of-the-art alternatives. Considering that Scatter SVM constitutes a more restricted model with far less variables of optimization, we consider these results encouraging, in the sense that Scatter SVM may perform well at a reduced computational cost. For example, running CS on the "Vowel" data (full cross-validation) required 3 days of computations. All the three other methods only required a small fraction of that time.

The tendency seems to be that where the results differ somewhat, the OVO C-SVM, in particular, has an edge. This is not surprising compared to Scatter SVM, since the reference to the global mean in Scatter SVM introduces a form of stiffness in terms of the regularization of the model, which will require a certain homogeneity among the classes, with respect to e.g. noise and outliers, to be at its most effective. For noisy data sets, a more fine grained class wise regularization approach will have many more variables of optimization available to capture the fine structure in the data, at the expense of computational simplicity. The USPS data may represent such an example, where Scatter SVM performs worse than all the alternatives.

[3] `http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/multiclass.html`

[4] Obtained from `http://cs.nyu.edu/~roweis/data.html`

5. CONCLUSIONS

By providing a new interpretation of the dual of μ-SVMs in terms of scatter, we have have proposed and implemented a multi-class extension named Scatter SVM. Promising results have been obtained.

6. REFERENCES

[1] C. Cortes and V.N. Vapnik, "Support vector networks," *Machine Learning*, vol. 20, pp. 273–297, 1995.

[2] Erin J. Bredensteiner and Kristin P. Bennett, "Multicategory Classification by Support Vector Machines," *Comput. Optim. Appl.*, vol. 12, no. 1-3, pp. 53–79, 1999.

[3] J. Weston and C. Watkins, "Support Vector Machines for Multi Class Pattern Recognition," in *Proceedings of European Symposium on Artificial Neural Networks*, Bruges, Belgium, April 21-23, 1999.

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

[5] Y. Lee, Y. Lin, and G. Wahba, "Multicategory Support Vector Machines: Theory and Application to the Classification of Microarray Data and Satellite Radiance Data," *Journal of the American Statistical Association*, vol. 99, no. 465, pp. 67–81, 2004.

	# train/test	# class	# attributes	Scatter SVM	OVR SVM	OVO SVM	CS
Iris	150	3	4	97.33 ± 3.44	97.33 ± 3.44	97.33 ± 3.44	97.33 ± 3.44
Wine	178	3	13	98.33 ± 2.68	98.33 ± 2.68	98.89 ± 2.34	98.89 ± 2.34
Glass	214	6	13	71.90 ± 7.60	70.95 ± 8.53	72.86 ± 8.11	70.48 ± 10.95
Vowel	528	11	10	99.24 ± 0.98	99.06 ± 1.33	99.44 ± 0.91	99.06 ± 1.33
Segment	2310	7	19	97.62 ± 1.25	97.49 ± 1.08	97.71 ± 1.06	97.40 ± 1.14
MNIST (0-4)	2000	5	784	99.00 ± 0.62	99.20 ± 0.59	99.20 ± 0.42	99.20 ± 0.59
Satimage	4435/2000	6	36	90.60	90.95	91.00	90.55
Dna	2000/1186	3	180	98.57	98.40	98.31	98.31
USPS	7291/2007	10	256	94.92	95.76	95.47	95.47

Table 3. Classification results on several real-world data sets.

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

[7] R. Jenssen, D. Erdogmus, J. C. Principe, and T. Eltoft, "Some Equivalences between Kernel Methods and Information Theoretic Methods," *Journal of VLSI Signal Processing*, vol. 45, pp. 49–65, 2006.

[8] Michael E. Mavroforakis, Margaritis Sdralis, and Sergios Theodoridis, "A Novel SVM Geometric Algorithm Based on Reduced Convex Hulls," *Pattern Recognition, International Conference on*, vol. 2, pp. 564–568, 2006.

[9] S. S. Keerthi, S. K. Shevade, C. Bhattacharyya, and K. R. K. Murthy, "A Fast Iterative Nearest Point Algorithm for Support Vector Machine Classifier Design," *IEEE Transactions on Neural Networks*, vol. 11, pp. 124–136, 2000.

[10] Ricardo Nanculef, Carlos Concha, Héctor Allende, Diego Candel, and Claudio Moraga, "AD-SVMs: A Light Extension of SVMs for Multicategory Classification," *Int. J. Hybrid Intell. Syst.*, vol. 6, no. 2, pp. 69–79, 2009.

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

[12] C. J. C. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition," *Knowledge Discovery and Data Mining*, vol. 2, no. 2, pp. 121–167, 1998.

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

[14] J. C. Platt, "Fast Training of Support Vector Machines using Sequential Minimal Optimization," in *Advances in Kernel Methods — Support Vector Learning*, B. Schölkopf, C.J.C. Burges, and A.J. Smola, Eds., Cambridge, MA, USA, 1999, pp. 185–208, MIT Press.

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

[16] 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`.

[17] Sören Sonnenburg, Gunnar Rätsch, Sebastian Henschel, Christian Widmer, Jonas Behr, Alexander Zien, Fabio de Bona, Alexander Binder, Christian Gehl, and Vojtech Franc, "The SHOGUN Machine Learning Toolbox," *Journal of Machine Learning Research*, 2010.

[18] B. Schölkopf and A.J. Smola, *Learning with Kernels*, MIT Press, Cambridge, MA, 2002.

[19] T. Joachims, T. Finley, and Chun-Nam Yu, "Cutting-Plane Training of Structural SVMs," *Machine Learning*, vol. 77, no. 1, pp. 27–59, 2009.

[20] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambrigde University Press, Cambridge, UK, 2004.

[21] A. J. Smola, S. V. N. Vishwanathan, and Quoc Le, "Bundle Methods for Machine Learning," in *Advances in Neural Information Processing Systems 20*, 2008.

[22] Ryan M. Rifkin and Ross A. Lippert, "Value Regularization and Fenchel Duality," *J. Mach. Learn. Res.*, vol. 8, pp. 441–479, 2007.

[23] C.-W. Hsu and C.-J. Lin, "A Comparison of Methods for Multiclass Support Vector Machines," *IEEE Transactions on Neural Networks*, vol. 13, no. 2, pp. 415–425, 2002.