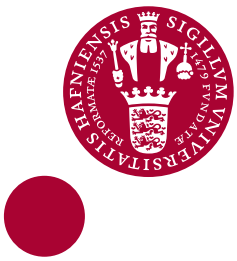

A unified view on multi-class support vector classification

Doan, Ürün; Glasmachers, Tobias; Igel, Christian

Published in:
Journal of Machine Learning Research

Publication date:
2016

Document version
Publisher's PDF, also known as Version of record

Journal of Machine Learning Research 17 (2016) 1-32 Submitted 8/11; Revised 8/15; Published 4/16

A Unified View on Multi-class Support Vector Classification

Ürün Doğan UDOGAN@MICROSOFT.COM
Microsoft Research

Tobias Glasmachers TOBIAS.GLASMACHERS@INI.RUB.DE
Institut für Neuroinformatik
Ruhr-Universität Bochum, Germany

Christian Igel IGEL@DIKU.DK
Department of Computer Science
University of Copenhagen, Denmark

Editor: Ingo Steinwart

Abstract

A unified view on multi-class support vector machines (SVMs) is presented, covering most prominent variants including the one-vs-all approach and the algorithms proposed by Weston & Watkins, Crammer & Singer, Lee, Lin, & Wahba, and Liu & Yuan. The unification leads to a template for the quadratic training problems and new multi-class SVM formulations. Within our framework, we provide a comparative analysis of the various notions of multi-class margin and margin-based loss. In particular, we demonstrate limitations of the loss function considered, for instance, in the Crammer & Singer machine.

We analyze Fisher consistency of multi-class loss functions and universal consistency of the various machines. On the one hand, we give examples of SVMs that are, in a particular hyperparameter regime, universally consistent without being based on a Fisher consistent loss. These include the canonical extension of SVMs to multiple classes as proposed by Weston & Watkins and Vapnik as well as the one-vs-all approach. On the other hand, it is demonstrated that machines based on Fisher consistent loss functions can fail to identify proper decision boundaries in low-dimensional feature spaces.

We compared the performance of nine different multi-class SVMs in a thorough empirical study. Our results suggest to use the Weston & Watkins SVM, which can be trained comparatively fast and gives good accuracies on benchmark functions. If training time is a major concern, the one-vs-all approach is the method of choice.

Keywords: support vector machines, multi-class classification, consistency

1. Introduction

Support vector machines (SVMs, Boser et al., 1992; Cortes and Vapnik, 1995) are founded on the intuitive geometric concepts of large margin separation and regularized risk minimization, and they are well embedded into statistical learning theory. However, there is no unique canonical extension to the case of multiple classes, neither from a geometric nor from a learning theoretical point of view. Instead, several variants relying on slightly different notions of margin and margin-based loss have been proposed. We present a unified view on multi-class SVMs. This view allows to identify similarities and differences between exist-

ing approaches. It provides a thorough framework for analyzing, designing, and efficiently training multi-class SVMs.

There exist generic ways of doing multi-category classification based on arbitrary binary learning machines by combining hypotheses from independently trained binary classifiers. In contrast, the focus of this study is on multi-class extensions of SVMs that aim at generalizing the notions of margin and large margin loss. These so-called all-in-one machines cast the learning problem into a single quadratic program, which allows them to take all class relationships into account simultaneously. The first of these all-in-one formulations was independently proposed by Weston and Watkins (1999), Vapnik (1998), and Bredensteiner and Bennett (1999). Crammer and Singer (2002) modified this machine, and their approach is frequently used, in particular when dealing with structured output. In addition to these popular methods, we consider the machine by Lee et al. (2004), a variant proposed by Liu and Yuan (2011), as well as multi-class maximum margin regression (Szedmak et al., 2006). The latter is equivalent to a multi-class SVM suggested by Zou et al. (2008). From a geometric point of view, the various approaches differ in the way they extend the concepts of margin and margin violation (or, to be more precise, margin-based loss) to multiple classes. Differentiating the machines form a learning-theoretical perspective is more difficult. Albeit there have been extensions of generalization bounds derived for binary SVMs to the multi-class case (e.g., Guermeur, 2007; Doğan et al., 2012), these do not allow to deduce relative advantages or disadvantages of certain multi-class SVM formulations. But there are hints from theoretical analyzes of consistency. In particular, the loss function proposed by Lee et al. was the first multi-class large margin loss known to be Fisher consistent (Lee et al., 2004; Tewari and Bartlett, 2007; Liu, 2007). The only other loss functions we are aware of sharing this property are those proposed in the work of Liu and Yuan (2011). In practice, however, the decision of which multi-class SVM to use is most often not governed by theoretical arguments, but by training-time considerations and the availability of efficient implementations. For instance, the canonical extension of binary SVMs to multiple classes proposed by Weston and Watkins and the machine by Lee et al. are rarely used. These approaches are theoretically sound and experiments indicate that they lead to well-generalizing hypotheses, but efficient training algorithms have been lacking so far.

Against this background, we analyze large-margin multi-category classification theoretically and derive fast training algorithms for established as well as for new learning machines, which we then evaluate empirically. Several researchers investigated multi-class SVM classification methods before, either experimentally (Hsu and Lin, 2002; Rifkin and Klautau, 2004) or conceptually (Allwein et al., 2001; Hill and Doucet, 2007; Liu, 2007; Guermeur, 2007). The experimental papers by Hsu and Lin (2002) and Rifkin and Klautau (2004) focus on the empirical comparison of methods. Some well established multi-class SVMs are missing in these studies. This also holds true for the conceptually oriented work by Allwein et al. (2001). The inspiring theoretical studies by Hill and Doucet (2007) and Liu (2007) completely lack any empirical comparison. The work most closely related to ours is perhaps the framework proposed by Hill and Doucet. They analyze several multi-class SVMs and present a solver for machine training, however, our study goes beyond their important achievements in several aspects. While Hill and Doucet's framework unifies already existing methods, one can neither easily develop new machines within the framework nor clearly identify the conceptual differences between machines. Further, their analysis

does not provide hints on how loss functions may affect the generalization performance of different machines. Finally, it is important to provide an extensive and unbiased empirical comparison.

This article is organized as follows. The next section presents our unifying framework, which categorizes multi-class SVMs according to their choice of multi-class loss function, margin function, and aggregation operator in subsections 2.1–2.3. Subsection 2.4 shows how the existing multi-class machines fit into our scheme. Then we derive new SVM variants suggested by the framework. Subsection 2.6 states a template for quadratic programs describing the learning problems induced by machines falling into our framework. Section 3 analyzes how different design choices affect the Fisher and universal consistency of the loss functions and the classifiers, respectively. The section closes with showcase experiments on synthetic test problems designed to highlight consequences of certain design choices. Section 4 presents an extensive experimental evaluation of various SVM variants, both with linear as well as Gaussian kernels. The theoretical as well as empirical findings finally lead us to conclusions including recommendations for the use of multi-class SVMs in practice.

2. A Framework for Multi-category SVM Classification

Multi-class SVMs construct hypotheses $h : X \to Y$ from training data $\big((x_1, y_1), \ldots, (x_\ell, y_\ell)\big) \in \big(X \times Y\big)^\ell$, where X and Y are the input and label space, respectively. The data points are assumed to be sampled i.i.d. from a distribution P on $X \times Y$. We restrict our considerations to the standard case of a finite label space and set $Y = \{1, \ldots, d\}$ (there exist extensions of multi-class SVMs to infinite label spaces, e.g., Bordes et al. e.g., 2008). The goal of training a multi-category SVM is to map the training data to a hypothesis minimizing the risk (generalization error) $\mathcal{R}(h) = \int\limits_{X \times Y} L(h(x), y) \ \mathrm{d}P(x, y)$, where the loss function L is typically the 0-1-loss, which is zero if both arguments are identical and one otherwise.

For multi-category classification the binary SVM concept of thresholding a real-valued decision function $f : X \to \mathbb{R}$ at zero is insufficient. The canonical extension to $d > 2$ classes is to employ a vector-valued linear decision function $f : X \to \mathbb{R}^d$, where the maximal component indicates the decision. Using a kernel-induced feature space, the corresponding hypothesis takes the form

$$h : X \to Y = \{1, \ldots, d\}\,; \quad x \mapsto \arg\max_{c \in Y} \big\{ f_c(x) \big\} = \arg\max_{c \in Y} \big\{ \langle w_c, \phi(x) \rangle + b_c \big\} \ , \qquad (1)$$

where $\phi : X \to \mathcal{H}$ is a feature map into an inner product space $\mathcal{H}$, $w_1, \ldots, w_d \in \mathcal{H}$ are class-wise weight vectors, and $b_1, \ldots, b_d \in \mathbb{R}$ are class-wise bias/offset values. The feature map is defined by a positive definite kernel function $k : X \times X \to \mathbb{R}$ with the property $k(x, x') = \langle \phi(x), \phi(x') \rangle$. We presume that the arg max operator in Equation (1) returns a single class index (ties may, for example, be broken at random). To ease the notation, we define the risk $\mathcal{R}(f)$ to be equal to the corresponding $\mathcal{R}(h)$.

Adding the same function $g : X \to \mathbb{R}$ to all components f_c of the decision function does not change the decision as $\arg\max_{c \in Y} \Big\{ f_c(x) \Big\} = \arg\max_{c \in Y} \Big\{ f_c(x) + g(x) \Big\}$. This is often undesired, for example, for the sake of uniqueness of a solution. This problem can be fixed

by enforcing the so-called *sum-to-zero* constraint

$$\sum_{c=1}^{d} f_c(x) = 0 \ . \tag{2}$$

At least for universal kernels (Steinwart, 2002a) the constraint is equivalent to $\sum_{c=1}^{d} w_c = 0$ and $\sum_{c=1}^{d} b_c = 0$. With this constraint we have $f_2 = -f_1$ for the special case of binary classification, and the hypothesis $h(x) = \operatorname{sign}(f_1(x))$ maps the first class to $+1$ and the second to -1. Thus, the binary SVM can be recovered as a special case of the multi-class SVM, which is an important design criterion.

2.1 Multi-class Loss Functions

Support vector machines select a hypothesis by minimizing the regularized empirical risk functional

$$\frac{1}{2}\|f\|^2 + C \cdot \sum_{i=1}^{\ell} L(f(x_i), y_i) \ , \tag{3}$$

where L is a large-margin loss function and $C > 0$ is a user-defined regularization constant. The squared norm is defined as $\|f\|^2 = \sum_{c \in Y} \|f_c\|^2 = \sum_{c \in Y} \|w_c\|^2$. Most existing approaches to multi-class large margin classification differ only in the type of loss function, which should typically be an upper bound on the 0-1-loss, lead to an optimization problem that can be solved efficiently, and vanish on an open set to allow for sparse solutions. The key observation for our unifying framework is that multi-class SVM loss functions can be decomposed into meaningful components, namely a set of margin functions for the different classes, a large-margin loss for binary problems, and an aggregation operator, combining the various target margin violations into a single loss value. This decomposition is found in all existing approaches to large margin multi-class classification. Given a labeled point $(x, y) \in X \times Y$ and the value $f(x) \in \mathbb{R}^d$ of the decision function, the general form of most multi-class loss functions is

$$L\big(f(x), y\big) = \Delta\Big(\widehat{L}^{\mathrm{bin}}\big(\mu(f(x), y), y\big)\Big) \ ,$$

or conceptually "$L = \Delta \circ \widehat{L}^{\mathrm{bin}} \circ \mu$". Here μ is a *margin function*, Δ is an *aggregation operator*, and L^{bin} is a loss function for binary classification, such as the hinge loss $L^{\mathrm{hinge}}(\mu) = \max\{0, 1 - \mu\}$, the squared hinge loss, or the Huber loss. With $\widehat{L}^{\mathrm{bin}}$ we denote the component-wise application of this loss function to the margin values.

In the next two subsections, we formalize the concepts of margin functions and aggregation operators and give examples of how existing machines can be expressed in our framework.

2.2 Margin Functions

The following definition extends the concept of a scalar margin of a training example to a vector-valued margin function.

Definition 1 (margin function) *A* margin function *is a non-constant function*

$$\mu : \mathbb{R}^d \times Y \to \mathbb{R}^d; \quad \mu = (\mu_1, \ldots, \mu_d)$$

of a prediction $f(x) \in \mathbb{R}^d$ *and a label* $y \in Y$ *that is linear in its first argument and fulfills the properties:*

1. $\forall y \in Y : \mu_y(f(x), y)$ *is non-decreasing in* $f_y(x)$,
2. $\forall c, y \in Y, c \neq y : \mu_c(f(x), y)$ *is non-increasing in* $f_c(x)$,
3. $\Big[\forall c \in Y : \mu_c(f(x), y) \geq 0$ *and* $\exists c \in Y : \mu_c(f(x), y) > 0\Big] \Rightarrow \big[\arg\max_{c \in Y} \big\{f_c(x)\big\} = y\big]$

Due to linearity in $f(x)$ margin functions are of the form

$$\mu_c\big(f(x), y\big) = \sum_{m=1}^{d} \nu_{y,c,m} \cdot f_m(x_i)$$

and can thus be expressed in terms of coefficients $\nu_{y,c,m}$. Property 3 in the above definition ensures that positive margins imply correct classification. Note that the reverse statement is not guaranteed, so, correct classification does not necessarily imply that all margin functions are non-negative. The definition is not as general as it may seem. If we assume an equal treatment of all classes, then the degrees of freedom in $\nu_{y,c,m}$ are drastically reduced.

The following types of margins are of primary interest:

Definition 2 (relative and absolute margins) *The* relative margin function *is given by*

$$\mu_c^{rel}(f(x), y) = \frac{1}{2}\big(f_y(x) - f_c(x)\big)$$

and the absolute margin function *by*

$$\mu_c^{abs}(f(x), y) = \begin{cases} +f_c(x) & \text{for } c = y \\ -f_c(x) & \text{for } c \in Y \setminus \{y\} \end{cases}.$$

The corresponding coefficients are

$$\nu_{y,c,m} = \frac{1}{2}\left(\delta_{y,m} - \delta_{c,m}\right) \qquad \text{(relative)}$$

and

$$\nu_{y,c,m} = \delta_{m,c} \cdot (2\delta_{c,y} - 1) = -(-1)^{\delta_{c,y}} \cdot \delta_{m,c} \qquad \text{(absolute)}$$

for the relative and the absolute margin, respectively, where $\delta_{a,b} = 1$ if $a = b$ and $\delta_{a,b} = 0$ otherwise denotes the Kronecker symbol. Both margin functions can be combined, as in the work of Liu and Yuan (2011). They have a number of interesting properties:

- The coefficients $\nu_{y,c,m}$ are sparse, so that the computation of a margin value μ_c from $f(x)$ is a constant time operation (independent of the number of classes).

- The "own-class-margin" component $\mu_y^{\text{rel}}(f(x), y)$ of relative margins is always zero. For a two-class problem, the other component coincides with the margin $y \cdot f(x)$ of the binary SVM (with labels $y \in \{\pm 1\}$ and real-valued decision function f).
- If the classification according to Equation (1) is correct, then no component of the relative margin function is negative. This implication does not necessarily hold for absolute margin functions regardless of whether the sum-to-zero constraint is enforced or not.
- It holds $\sum_m \nu_{y,c,m} = 0$ for relative but not for absolute margins. Adding the same function to all components f_c of the decision function does not change the decision (1). Hence it is reasonable to demand that adding the same value to all $\nu_{y,c,m}$ does not change the margins. This property is violated for absolute margins. This renders both the sign and the absolute value of a single margin (i.e., a component of the margin function) meaningless. This issue could be addressed by enforcing $\sum_m \nu_{y,c,m} = 0$ also for absolute margins.
- In principle, the sum-to-zero constraint (2) can be avoided at the level of the decision function by shifting the coefficients ν such that they fulfill $\sum_m \nu_{y,c,m} = 0$. However, this would destroy the sparsity of the coefficients. Thus, it can be argued that the sum-to-zero constraint $\sum_c f_c = 0$ is a preferable solution.
- A regularizer of the form $\sum_{c=1}^d \|w_c\|^2$ has the effect that the optimal solution of a relative margin SVM always fulfills the sum-to-zero constraint. This has been observed for particular machines by Guermeur (2007) and by Wu and Liu (2007), but can be shown to be true for all relative margin SVMs. Adding the same vector w to all weight vectors w_c (or the same function $g(x) = \langle w, \phi(x) \rangle$ to all components f_c) does not affect relative margins and thus the empirical error term in the primal problem. Thus, to find the optimal w we have to minimize the complexity term

$$\sum_{c=1}^{d} \langle w_c + w, w_c + w \rangle = \sum_{c=1}^{d} \langle w_c, w_c \rangle + 2 \cdot \sum_{c=1}^{d} \langle w_c, w \rangle + d \cdot \langle w, w \rangle$$

  w.r.t. w. Setting the derivative $2 \cdot \left(\sum_{c=1}^d w_c + d \cdot w \right)$ to zero gives $w = -\frac{1}{d} \sum_{c=1}^d w_c$, which results in $\sum_{c=1}^d (w_c + w) = 0$.
- Undesired situations can occur if the absolute margin function is applied without the sum-to-zero constraint. Consider, for instance, the case of all components of f being negative. In this case the "own class" margin μ_y is negative while all "other class" margins μ_c, $c \neq y$, are positive, and the example could still be classified correctly. This seems counter-intuitive, although formally allowed by Definition 1. Thus, we argue that the sum-to-zero constraint should be demanded for any multi-class SVM. The one-versus-all machine (OVA, see below) relies on absolute margins, but the sum-to-zero constraint is not enforced. However, in OVA it is not possible that all components of the margin function are negative.

2.3 Aggregation Operators

Definition 1 ensures that positive margins μ_c indicate correct classification. Therefore it is natural to construct a large margin loss from target margin violations, defined as

$$v_c(f(x), y) = \max\left\{0, \gamma_{y,c} - \mu_c\big(f(x), y\big)\right\} .$$

The target margins $\gamma_{y,c}$ are typically chosen to be $\gamma_{y,c} = 1$, which we assume in the following if not stated otherwise. The margin violations $v_c = L^{\text{hinge}}(\mu_c(f(x), y)$ correspond to the hinge loss applied to the different margin components. As indicated above, other binary SVM loss functions can be applied at this point. In this case the resulting multi-class SVMs will be compatible with the corresponding binary SVM based on that surrogate loss (e.g., the squared hinge loss is considered by Guermeur, 2012).

The d margin violations need to be combined into a single cost value:

Definition 3 (aggregation operator) *An* aggregation operator *is a non-constant function*

$$\Delta : (\mathbb{R}_0^+)^d \times Y \to \mathbb{R}_0^+$$

of margin violations $v = (v_1, \ldots, v_d)$ and a label $y \in Y$ with the properties

- $\Delta\big((0, \ldots, 0), y\big) = 0$, *and*
- $\Delta((v_1, \ldots, v_d), y)$ *is monotonically increasing in all components v_i of the first argument.*

The following aggregation operators are employed in machines that have been proposed in the literature:

Definition 4 *We define the following aggregation operators:*

$$\Delta^{self}\big((v_1, \ldots, v_d), y\big) = v_y \qquad (\textit{self} \text{ operator})$$

$$\Delta^{t\text{-}max}\big((v_1, \ldots, v_d), y\big) = \max_{c \in Y}\left\{v_c\right\} \qquad (\textit{total-max} \text{ operator})$$

$$\Delta^{o\text{-}max}\big((v_1, \ldots, v_d), y\big) = \max_{c \in Y \setminus \{y\}}\left\{v_c\right\} \qquad (\textit{max-over-others} \text{ operator})$$

$$\Delta^{t\text{-}sum}\big((v_1, \ldots, v_d), y\big) = \sum_{c \in Y} v_c \ , \qquad (\textit{total-sum} \text{ operator})$$

$$\Delta^{o\text{-}sum}\big((v_1, \ldots, v_d), y\big) = \sum_{c \in Y \setminus \{y\}} v_c \ . \qquad (\textit{sum-over-others} \text{ operator})$$

These operators can be understood as computing the value of a linear program, as outlined in the supplementary material (Section A).

2.4 Unification of Existing Machines

This study considers all-in-one approaches to SVM multi-class classification. These methods extend the SVM concept of large-margin classification to the multi-class case and cast the learning task as a single optimization problem. The first all-in-one methods proposed independently by Weston and Watkins (1999), Vapnik (1998, Section 10.10), and Bredensteiner and Bennett (1999) turned out to be equivalent, up to rescaling of the decision functions and the regularization parameter $C > 0$. We refer to this method as WW or as WW-SVM. It is expressed within our framework as the machine relying on the sum operator[1] applied to relative margins.

An alternative multi-class SVM was proposed by Crammer and Singer (2002). Like the WW-SVM, the CS machine takes all class relations into account simultaneously and solves a single optimization problem, however, with fewer slack variables in its primal optimization problem. It corresponds to combining relative margins with the max-over-others operator.

There are two key differences between the WW- and CS-SVMs. First, the aggregation operators are different. Second, the CS-SVM has originally been defined only for hypotheses without bias term. The machine can be extended to hypotheses with bias (Hsu and Lin, 2002), but efficient training algorithms have been missing for this extension. In our framework we do not distinguish between hypothesis classes with and without bias; all machines are defined for both classes.

Lee, Lin and Wahba (2004) proposed a distinct approach to multi-class SVM classification. We refer to this approach as the LLW-SVM. It was the first machine to use absolute margins in an all-in-one approach. The absolute margin violations are combined under the sum-to-zero constraint by means of the sum-over-others operator.

The LLW machine was the first multi-class SVM with a classification calibrated loss function, which guarantees its Fisher consistency (Lee et al., 2004; Tewari and Bartlett, 2007; Liu, 2007; Liu and Yuan, 2011). That is, in the limit of infinite data minimal risk in the LLW sense implies minimal 0-1-risk and thus a Bayes-optimal decision rule.

The multi-class maximum margin regression (MMR) method proposed by Szedmak et al. (2006) is driven by the observation that the quadratic term

$$\sum_{i=1}^{\ell} \sum_{j=1}^{\ell} \alpha_i \alpha_j y_i y_j k(x_i, x_j)$$

in the Wolfe dual of the binary SVM (Cortes and Vapnik, 1995; Bottou and Lin, 2007) can be interpreted as the squared norm with respect to the product kernel $k \otimes k_y$, where the "label kernel" $k_y : \{\pm 1\} \times \{\pm 1\} \to \mathbb{R}$ is defined as $k_y(y_i, y_j) = y_i y_j$. This is the standard inner product on the label space $Y = \{-1, +1\} \subset \mathbb{R}$. This idea is generalized to multiple classes by considering the standard inner product on $\mathbb{R}^d$ applied to label prototype vectors $\{\hat{y}_1, \ldots, \hat{y}_d\} = Y \subset \mathbb{R}^d$ representing the classes. This corresponds one-to-one to the definition of a positive definite kernel $k_y(c, c') = \langle \hat{y}_c, \hat{y}_{c'} \rangle$ on the standardized label space $Y = \{1, \ldots, d\}$. Two different sets of prototypes have been proposed, namely

$$\hat{y}_c = \sqrt{\frac{d-1}{d}} \left(\frac{-1}{d-1}, \ldots, 1, \ldots, \frac{-1}{d-1} \right)$$

1. For relative margins, the sum-over-others and the total-sum operator coincide.

and $\hat{y}_c = (0, \ldots, 1, \ldots, 0)$, where the c-th component equals 1. The first set corresponds to a symmetric setup of unit vectors with maximal angles between prototypes, while the second setup relies on an orthonormal basis. We refer to these variants as MMR and MMR$^\perp$, respectively. They correspond to the label kernels

$$k_y(y_i, y_j) = \begin{cases} 1 & \text{for } y_i = y_j \\ -\frac{1}{d-1} & \text{otherwise} \end{cases} \qquad \text{and} \qquad k_y^\perp(y_i, y_j) = \delta_{y_i, y_i} \ .$$

Both of these machines are obtained by applying the self aggregation operator to absolute margins, where the MMR machine enforces the sum-to-zero constraint, while the MMR$^\perp$ machine does not.

The MMR$^\perp$ method applied to a two-class problem does typically not give the same decision function as the standard binary SVM. With its orthogonal label kernel, it explicitly ignores all interactions between classes, so that the machine effectively trains d independent machines, similar to one-class SVMs for the different classes.

The unique selling proposition of the MMR and MMR$^\perp$ machines is their applicability to problems with large numbers of classes. By design, the optimization problem of the MMR machine has only ℓ dual variables. Even better, in case of the MMR$^\perp$ machine the problem decomposes into d independent sub-problems. However, this means that the MMR$^\perp$-SVM does not take any inter-class relationships in the data into account.

Our unification also captures the *reinforced multicategory SVM* (RM-SVM) recently proposed by Liu and Yuan (2011). This machine, or family of machines, weights two types of margin violations using a parameter $\gamma \in [0, 1]$. In our framework, it can be viewed as relying on a *reinforced margin function* defined as

$$\mu_c^{\mathrm{r}}(f(x), y) = \begin{cases} \gamma f_c(x) & \text{for } c = y \\ -(1-\gamma) f_c(x) & \text{for } c \in Y \setminus \{y\} \end{cases}$$

and using the total-sum operator and the coefficients

$$\gamma_{y,c} = \begin{cases} \gamma(d-1) & \text{for } c = y \\ (1-\gamma) & \text{for } c \in Y \setminus \{y\} \end{cases} \tag{4}$$

for combining the margin violations. The sum-to-zero constraint is enforced. For $\gamma = 0$, RM equals LLW and for $\gamma = 1$ it equals MMR with a scaled target margin.

There exist several general techniques to build a multi-category classifier based on some binary classifiers, which can be chosen to be binary SVMs. The most prominent of these so-called sequential approaches are the one-versus-one (OVO) and the one-versus-all (OVA) scheme. As an OVO approach inevitably requires some additional non-canonical decision making procedure different from (1) (e.g., Hastie and Tibshirani, 1998; Platt et al., 2000; Chang and Lin, 2011) and the optimal choice of this procedure is highly task-dependent, OVO-type approaches are out of the scope of this study (error-correcting-output-codes, ECOC, provide a general framework covering OVA and many OVO methods, see Dietterich and Bakiri, 1995; Allwein et al., 2001; Passerini et al., 2004).

However, the OVA scheme fits into our framework. It relies on only d different binary decision functions, each separating one class from all the rest (Vapnik, 1998, Section 10.10).

Usually this class is considered the "positive" class ($y = +1$) in the binary problem, and all other classes are mapped to the "negative" class ($y = -1$). The resulting decision function can be thought of as voting for the class under consideration. Let $y_i^c = +1$ if $c = y_i$ and $y_i^c = -1$ if $c \neq y_i$ denote the binary label of the i-th example when training the machine separating class c from the rest. Then OVA constructs decision functions f_c according to

$$\min_{f_c} \quad \frac{1}{2}\|f_c\|^2 + C \cdot \sum_{i=1}^{\ell} L^{\text{hinge}}\big(y_i^c \cdot f_c(x_i)\big) \ .$$

and plugs them into Equation (1). Solving these d independent optimization problems at once results in a problem of type (3). This proceeding amounts to summing up the margin violations of all binary problems. Thus, OVA can be viewed as total-sum aggregation applied to absolute margins, without enforcing the sum-to-zero constraint.

Our framework highlights the conceptual similarities and differences of the above-mentioned machines, summarized in Table 1.

machine	margin μ	aggregation operator Δ	sum-to-zero constraint	bias term
WW	relative	sum-over-others	optional	yes
CS	relative	maximum-over-others	optional	no
LLW	absolute	sum-over-others	yes	yes
MMR	absolute	self	yes	yes
$\text{MMR}^{\perp}$	absolute	self	no	yes
RM	absolute	total-sum (affine combination of self and sum-over-others)	yes	yes
OVA	absolute	total-sum	no	yes

Table 1: Properties of the various multi-category SVMs in terms of the unifying framework. The last column indicates the presence of the bias or offset term in the original formulation found in the literature; all machines can be formulated with or without bias term.

2.5 Completing the Picture: New Multi-class SVMs

Our unification makes it easy to construct novel loss functions based on the principles extracted from existing approaches. From Table 1 it becomes obvious that some combinations of established margin functions and aggregation operators have not yet been explored. In the following, we define the corresponding machines and name them by a three-letter code, abbreviating the margin and loss types as indicated in bold in the following. The first of these new SVMs is the AMO machine, combining the **a**bsolute margin concept with the **m**ax-over-**o**thers operator. The machine can be understood as a combination of the LLW-SVM, from which it takes the margin concept, and the CS-SVM with its max-over-others aggregation.

The other two machines, ATM and ATS, use **a**bsolute margins with the **t**otal **m**aximum and the **t**otal **s**um operators. The ATS-SVM resembles the RM-SVM with $\gamma = 0.5$, however,

ATS uses target margins of $\gamma_{y,c} = 1$ for $y, c \in \{1, \ldots, d\}$, while RM-SVM uses the target margins defined by (4) which differ depending on whether c equals y or not. The only difference between the OVA scheme and the ATS machine is the presence of the sum-to-zero constraint in the latter. Still, the ATS machine is an all-in-one machine, because its training problem is not a composition of independent (binary) problems.

2.6 Unified Dual Problem

Training SVMs amounts to solving convex quadratic optimization problems. The traditional approach to solving these problems when using non-linear kernels is considering the corresponding dual formulations. In the following, we derive the Wolfe dual of the unified primal problem (3). The structure of the dual is often well-suited for the application of decomposition algorithms, which have proven to be a powerful choice for training SVMs with non-linear kernels.

For the case without sum-to-zero constraint, we arrive at the following dual problem (see supplement Section B for details):

$$\begin{aligned} \max_{\alpha} \quad & \sum_{i,p} \gamma_{y_i,p} \cdot \alpha_{i,p} - \frac{1}{2} \sum_{i,p} \sum_{j,q} M_{y_i,p,y_j,q} \cdot k(x_i, x_j) \cdot \alpha_{i,p} \cdot \alpha_{j,q} && (5) \\ \text{s.t.} \quad & \forall i, p: \quad 0 \leq \alpha_{i,p} \\ & \forall i, r: \quad \sum_{p \in P_y^r} \alpha_{i,p} \leq C \\ & \forall c: \sum_{i,p} \alpha_{i,p} \nu_{y_i,p,c} = 0 && (6) \end{aligned}$$

with

$$M_{y_i,p,y_j,q} = \sum_{c} \nu_{y_i,p,c} \nu_{y_j,q,c} \ .$$

The bias parameters b_c can be obtained from the KKT optimality conditions.

If the sum-to-zero constraint is enforced, we obtain the same structure of the dual problem, but this time with

$$M_{y_i,p,y_j,q} = \sum_{m,n} \left(\delta_{m,n} - \frac{1}{d} \right) \nu_{y_i,p,m} \nu_{y_j,q,n} \ .$$

We call these auxiliary constants $M_{y_i,p,y_j,q}$ used in the formulation of the dual problems *kernel modifiers*, because they appear as multipliers of the kernel in the dual problem. Let the map $I : \{1, \ldots, \ell\} \times P_y \to \{1, \ldots, \ell |P_y|\}$ assign a unique index to each pair of training pattern and constraint. We define $Q \in \mathbb{R}^{\ell |P_y| \times \ell |P_y|}$ by

$$Q_{I(i,p),I(j,q)} = M_{y_i,p,y_j,q} \cdot k(x_i, x_j) \ .$$

This matrix, together with the vector $V \in \mathbb{R}^{\ell |P_y|}$, $V_{I(i,p)} = \gamma_{y_i,p}$, defines the quadratic objective function

$$W(\alpha) = V^T \alpha - \frac{1}{2} \alpha^T Q \alpha \tag{7}$$

of the dual problem, written in vector notation.

The weight vectors are given by

$$w_c = \sum_{i,p} \alpha_{i,p} \nu_{y_i,p,c} \phi(x_i) - \eta \tag{8}$$

with $\eta = \sum_{i,p} \alpha_{i,p} \left(\frac{1}{d} \sum_c \nu_{y_i,p,c}\right) \phi(x_i)$ in case of a sum-to-zero constraint and zero otherwise.

For solving the problem with bias parameters, which lead to d equality constraints, we adopt the *iterative problem alternation* approach by Kienzle and Schölkopf (2005).

2.7 Training Complexity

Table 2 gives the asymptotic training times of various approaches. Under the assumption that the runtime bounds are tight, the table suggests that training MMR$^\perp$ is faster than training OVO, which is slightly faster than MMR, which is again faster than training OVA, which is finally followed by the various all-on-one SVMs with $O(d)$ dual variables per training example. The asymptotic nature of the runtime bounds hides factors considerably influencing training times in practice: Different machines often perform best for different hyper-parameters, and different concepts of separability (as discussed in Section 2 and Section 3) have an impact.

MMR$^\perp$	OVO	MMR	OVA	WW / CS / LLW AMO / ATM / ATS / RM
$\sum_{c=1}^{d} \mathcal{C}(\ell_c)$ $\widehat{=}\, d \cdot \mathcal{C}(\ell/d)$	$\sum_{1 \le c < e \le d} \mathcal{C}(\ell_c + \ell_e)$ $\widehat{=}\, d^2 \cdot \mathcal{C}(\ell/d)$	$\mathcal{C}(\ell)$	$d \cdot \mathcal{C}(\ell)$	$\mathcal{C}(d\ell)$

Table 2: Asymptotic runtime of the training algorithms under the assumption that solving the different n-dimensional quadratic programs is in the complexity class $\mathcal{C}(n)$. There are good arguments to assume $\mathcal{C}(n) = O(n^q)$ for some q between 2 and 3 (Joachims, 1998; Bottou and Lin, 2007). The number of training patterns is denoted by ℓ, the number of examples per class c by ℓ_c, and the number of different classes by d. In the second row the complexity is given under the assumption of (roughly) balanced classes, that is, $\ell_c \in \Theta(\ell/d)$.

3. A Closer Look at Aggregation Operators and Margin Concepts

In this section, we analyze the concepts underlying the various multi-class SVM variants.

First, we will consider the asymptotic behavior in the limit of infinite data by studying consistency. This analysis is closely related to two streams of research. First, we will focus on the Fisher consistency of the loss functions employed for SVM training. This type of consistency, also known as classification calibration, has already been studied for a number of multi-class losses (Lee et al., 2004; Tewari and Bartlett, 2007; Liu, 2007; Liu and Yuan, 2011). It is only indirectly concerned with the SVM itself, since the effect of the regularizer is ignored. Second, we will consider universal consistency building on the well-known result for binary SVMs by Steinwart (2002b). It states that with a suitable

dependency $C(\ell)$ of the regularization parameter C on the data set size ℓ, for any universal kernel, the SVM approaches the Bayes-optimal decision in the limit of infinite data. In addition to the theoretical analysis, we will empirically study the performance of the SVMs on artificial learning tasks that have been designed to highlight crucial differences between margin concepts and aggregation operators.

3.1 Fisher Consistency of Multi-class Loss Functions

In the following, we will analyze the Fisher consistency of the loss functions considered in this study. Let $L(f(x), y)$ denote the loss function applied by a multi-class SVM for training. Let $f^* : X \to \mathbb{R}^d$ denote a minimizer (among all measurable functions) of the risk under the data generating distribution P. A loss function is Fisher consistent if

$$\arg\max \left\{ f_c^*(x) \,\middle|\, c \in Y \right\} \subset \arg\max \left\{ P(y \,|\, x) \,\middle|\, y \in Y \right\} \ .$$

In the following, we use the statements that a multi-class SVM variant and its training loss are Fisher consistent interchangeably.

The consistency of the LLW loss was established by Lee et al. (2004), and it was confirmed by Liu (2007) that this is the only example of a Fisher consistent multi-class loss among the popular WW-, CS-, MMR-, and LLW-SVMs. The ATS machine can be viewed as belonging to the family of reinforced multicategory SVMs. Thus, the Fisher consistency of the ATS loss follows from the analysis by Liu and Yuan (2011).

We will investigate the consistency of the new machines AMO and ATM following the proceeding by Liu (2007). We adopt the short notation $P_y = P(y \,|\, x)$ for the probability of observing class y given x and define $[t]_+ = \max\{0, t\}$. It holds:

Theorem 5 *Let $L(f(x), y)$ denote either the loss function used by the AMO machine or the loss function used by the ATM machine, that is, the loss resulting from application of either the max-over-others or the total-max operator to absolute margins:*

$$L(f(x), y) = \max_{c \in Y \setminus \{y\}} \left\{ v_c^{abs}(f(x), y) \right\} = \left[1 + \max_{c \in Y \setminus \{y\}} \left\{ f_c(x) \right\} \right]_+ \tag{AMO}$$

or

$$L(f(x), y) = \max_{c \in Y} \left\{ v_c^{abs}(f(x), y) \right\} = \max \left\{ \left[1 + \max_{c \in Y \setminus \{y\}} \left\{ f_c(x) \right\} \right]_+ , \left[1 - f_y(x) \right]_+ \right\} \ . \tag{ATM}$$

Then the minimizer f^* of the corresponding risk $\mathcal{R} = \mathbb{E}[L(f(x), y)]$, subject to the sum-to-zero constraint $\sum_{c \in Y} f_c(x) = 0$, satisfies:

- *If there exists a* majority class *$y \in Y$ such that $P_y > (d-1)/d$, then $f_y^*(x) = d - 1$ and $f_c^*(x) = -1$ for all $c \in Y \setminus \{y\}$.*
- *If $P_y < (d-1)/d$ for all $y \in Y$, then $f^*(x) = 0$.*

The proof, which is given in the supplementary material (Section C), works by analyzing the point-wise risk $\mathcal{R}_x = \sum_{y \in Y} P(y \,|\, x) L(f(x), y)$ and computing the function values $f_c(x)$ minimizing $\mathcal{R}_x$. Thus, the Fisher consistency of the loss functions used in the all-in-one machines considering absolute margins can be summarized as follows:

Corollary 1 *It follows directly from the previous results that the AMO-loss and the ATM-loss are not Fisher consistent while the LLW-loss and the ATS-loss are Fisher consistent.*

In light of the conceptual similarity between the ATS machine and the LLW-SVM, it is not surprising that the ATS-loss is Fisher consistent. However, combining margin violations by means of the maximum-operator appears to be problematic, since neither the AMO nor the ATM machine (the two "max-loss siblings" of the Fisher consistent LLW and ATS machines) share this property.

The question arises how Fisher consistency is related to (a) the margin type, (b) the aggregation of margins in the loss, and (c) the sum-to-zero constraint. Part (a) is easy to answer, since Liu (2007) has proved inconsistency of the relative margin machines (WW and CS). Thus, we focus on absolute margins in the following. We investigate the impact of the sum-to-zero constraint on the maximum and the sum of margin violations. It turns out that dropping the constraint destroys Fisher consistency:

Theorem 6 *Let $L(f(x), y)$ denote the maximum or the sum over absolute margin violations, and assume $P_y < 1/2$ for all $y \in Y$. Then the unconstrained minimizer f^* of the corresponding risk $\mathcal{R} = \mathbb{E}[L(f(x), y)]$ satisfies $\forall c \in Y : f_c^*(x) = -1$ for the sum and $\forall c \in Y : f_c^*(x) = 0$ for the maximum.*

We do not prove this statement here; the proof is analog to the one of Theorem 5 using the techniques developed by Liu (2007). The optimal solution for the maximum operator is all zeros and satisfies the sum-to-zero constraint (cf. Theorem 5), and therefore dropping the constraint does not make any difference. However, the constraint is relevant when the sum operator is employed. The resulting solution does not sum to zero, and it is not Bayes-optimal. This result confirms the importance of the constraint for Fisher consistency.

The ATS machine can be interpreted as the OVA machine with sum-to-zero constraint, and Theorem 6 covers the OVA machine, which is inconsistent in the absence of a majority class. Thus, adding the sum-to-zero constraint makes the OVA machine consistent, at the cost that the different weight vectors w_c cannot be obtained independently anymore.

3.2 Universal Consistency

While Fisher consistency is just a property of the loss function, universal consistency considers the whole machine including the regularizer and its impact relative to the data set size. Steinwart's universal consistency analysis for binary SVMs builds on the fact that the hinge loss is classification calibrated for binary problems. It can be extended to multi-class SVMs with Fisher consistent losses under the assumption of the same training set size dependent choice $C(\ell)$ of the regularization parameter as assumed for the binary SVMs. Thus, from Corollary 1 we get:

Theorem 7 *The LLW-SVM and the ATS-SVM with universal kernel and regularization parameter chosen according to Theorem 2 by Steinwart (2002b) are universally consistent classifiers.*

We omit the proof of this statement, because is a straight-forward extension of the involved proof for binary machines.

It can be shown that using the same rule for $C(\ell)$ as in Theorem 2 from Steinwart (2002b) with a non-Fisher consistent loss does not yield a universally consistent classifier. However, this negative result does not exclude the existence of a different dependency of C on ℓ so that the resulting classifier is universally consistent. Thus, the question arises whether for an SVM surrogate loss function L there exists a dependency $C(\ell)$ of the regularization parameter on the training set size such that the resulting classifier is universally consistent. In the following, we answer the question for all SVM variants covered in this study—with the surprising result that also the MMR-, WW-, and OVA-SVMs can be made consistent when operated with a completely different trade-off $C(\ell)$:

Theorem 8 *For the Gaussian kernel $k(x, x') = \exp(-\gamma\|x - x'\|^2)$ on an input space $X \subset \mathbb{R}^p$, the machines MMR-SVM, WW-SVM, LLW-SVM, ATS-SVM, and OVA-SVM with regularization parameter $C(\ell) \leq 1/(d \cdot \ell)$ and kernel parameter $\gamma(\ell)$ fulfilling $\lim_{\ell\to\infty} \gamma(\ell) = \infty$ and $\lim_{\ell\to\infty} \ell \cdot \gamma(\ell)^{-p/2} = \infty$ are universally consistent classifiers.*

Proof Let us first consider the all-in-one machines. The absolute values of all components of the quadratic matrix Q in the dual problem (7) are upper bounded by one. All components of the dual solution α are bounded by $C(\ell) = 1/(d \cdot \ell)$. The sum in the derivative

$$\frac{\partial W(\alpha)}{\partial \alpha_{i,p}} = 1 - \sum_{j,q} Q_{I(i,p),I(j,q)} \cdot \alpha_{j,q}$$

runs over at most $d \cdot \ell$ summands, each of which is bounded by $\frac{1}{d\cdot\ell}$. Thus all gradient components are non-negative in the whole box-shaped feasible region. It follows that all dual variables $\alpha_{i,p}$ end up at the upper bound C. From Equation (8) and the definition of the machines it can be seen that for the primal solution it holds $w_c = \tilde{w}_c - \tilde{\eta}$ with

- $\tilde{\eta} = \eta$ and $\tilde{w}_c = C \cdot \sum_{i|y_i=c} \phi(x_i)$ for the MMR machine,
- $\tilde{\eta} = \eta - C \cdot \sum_i \phi(x_i)$ and $\tilde{w}_c = C \cdot d/2 \cdot \sum_{i|y_i=c} \phi(x_i)$ for the WW machine,
- $\tilde{\eta} = \eta - C \cdot \sum_i \phi(x_i)$ and $\tilde{w}_c = C \cdot \sum_{i|y_i=c} \phi(x_i)$ for the LLW machine,
- $\tilde{\eta} = \eta - C \cdot \sum_i \phi(x_i)$ and $\tilde{w}_c = 2C \cdot \sum_{i|y_i=c} \phi(x_i)$ for the ATS machine.

The weight vectors $\tilde{w}_c$ are a scaled version of the kernel density estimator (KDE), which is well known to be universally consistent. The above conditions on $\gamma(\ell)$ translate into the preconditions of Theorem 10.1 by Devroye et al. (1996, p. 150) for the kernel width $h = \sqrt{1/\gamma(\ell)}$. The KDE output fed into the arg max decision rule (1) ignores the additive constant $\tilde{\eta}$ and yields consistent classification predictions.

The OVA result can be proven in the same way. Again, all dual variables take the value C. It is easy to see that the primal SVM solution is then a scaled KDE plus a constant vector: $w_c = \tilde{w}_c - \tilde{\eta}$, with $\tilde{\eta} = \sum_i \phi(x_i)$ and $\tilde{w}_c = 2 \cdot \sum_{i|y_i=c} \phi(x_i)$. Thus, the resulting classifier is universally consistent. ■

The above statement is a straight-forward reduction of the SVM to a simple kernel density estimator. The same result holds for binary SVMs. It is a much simpler statement with

a much simpler proof than the famous universal consistency result by Steinwart (2002b). The core idea of Steinwart's analysis is to carefully increase the modeling power of the SVM with growing number of training points so that the impact of regularizer decays to zero, while at the same time overfitting is avoided. In contrast, we keep the impact of the regularizer constantly high. When operating the SVM in this regime it loses its most important features, namely the large margin approach and the sparsity (Steinwart and Christmann, 2008). This may seem highly undesirable, but from the viewpoint of classification accuracy it does not matter much whether an SVM with a specific value of C works well because of a large margin or because of the consistency of KDE. Of course, KDE estimation is less sample and memory efficient than large-margin prediction. Sample efficiency is one of the primary arguments brought forward by Vapnik (1998) for preferring direct estimation of the decision boundary over a plug-in classifier based on KDE.

Theorem 8 establishes the consistency of the MMR, WW, and OVA machines, despite the Fisher inconsistency of their losses. This implies that an analysis of the loss function in isolation in terms of Fisher consistency is not sufficient for fully understanding the predictions made by these machines.

The simple trick of reverting to KDE-based prediction is not trivial, since it does not work for all large margin losses:

Theorem 9 *There exists no function $C(\ell)$ so that CS-SVM, AMO-SVM, or ATM-SVM become universally consistent.*

Proof Consider a single-point input space $X = \{x_0\}$ with any kernel $k(x_0, x_0) > 0$ (any positive kernel is universal on this space). Consider a problem with $d = 3$ classes. It is completely described by the class probabilities; w.l.o.g. they fulfill $p_1 \geq p_2 \geq p_3$. Lemma 4 by Liu (2007) for the CS machine, and Theorem 5 for the AMO-SVM and the ATM-SVM, respectively, offer choices of these probabilities so that the minimizer of the empirical risk is $f_1(x_0) = f_2(x_0) = f_3(x_0) = 0$, with corresponding weight vectors $w_1 = w_2 = w_3 = 0$. This inconsistent solution also minimizes the regularizer, and thus the regularized risk for all $C > 0$.[2] ■

The proof lifts results on Fisher-(in)consistency from the level of the loss function to inconsistency of the actual SVM, taking the regularizer into account. The case of the MMR, WW and OVA machines treated above makes clear that this extension is non-trivial, since for these machines the regularizer makes the difference between consistency and inconsistency. We regard this type of analysis as highly relevant, since SVMs do not actually minimize the empirical risk, but instead a combination of empirical risk and a specific regularizer.

The present analysis covers the three losses involving the max-aggregation completely, since for no C (or $C(\ell)$) they result in a universally consistent SVM. On the other hand, there is a gap in our analysis of machines employing sum-aggregation losses: For $\ell \cdot C(\ell) \to \infty$ Steinwart's analysis applies, while the trivial reduction to a KDE works only for $\ell \cdot C(\ell)$ (asymptotically) bounded by a rather small constant, so that all target margins are always

2. An alternative proof strategy is to construct a non-zero dual optimal solution that corresponds to the primal zero solution.

violated. The question whether an SVM can be consistent for parameters in the gap in between these regimes is left open.

3.3 Artificial Benchmark Problems

In the following, we will show experimental results on highly controlled problems with analytically defined distributions. This investigation complements and enriches the previous theoretical analysis. Some of the theoretical results on Fisher consistency of loss functions can be demonstrated in a non-asymptotic setting. The theoretical results on universal consistency require a universal kernel and thus an "arbitrarily rich" feature space. This is, however, not always the case in practice, where restricted and low-dimensional feature spaces can occur. We visualize the behavior of nine different multi-class SVMs over a range of values of the regularization parameter. The simple test problems serve as "counter examples" that help to understand when and why some of the machines deliver substantially sub-optimal solutions.

We define a basic test problem with two variants, one noise-free and one with label noise. The domain $X = S^1 = \left\{x \in \mathbb{R}^2 \,\middle|\, \|x\| = 1\right\}$ of these test problems is the unit circle. The circle is parameterized by $t \in [0, 20[$ through the curve $\beta(t) = \big(\cos(t \cdot \pi/10), \sin(t \cdot \pi/10)\big)$. The problem involves three classes $Y = \{1, 2, 3\}$, see Figure 1 and Figure 2. For the noiseless problem data points $(x, y) \in X \times Y$ are generated as follows. First the label $y \in Y$ is drawn from the uniform distribution on Y. Then x is drawn uniformly at random from the sector X_y. The sectors have different sizes. They are defined as $X_1 = \beta([0, 5))$, $X_2 = \beta([5, 11))$, and $X_3 = \beta([11, 20))$. The only change for the noisy case is that 90% of the labels are reassigned uniformly at random. In other words the distribution of the x-component remains unchanged, and conditioned on $x \in X_z$ the event $y = z$ has probability 40%, while the other two cases have probabilities of 30%. In both variants of the problem it is Bayes-optimal to predict label y on sector X_y. This solution can be realized by a linear model without offset term.

These problems are rather elementary. They are low-dimensional and thus easy to visualize. Their only difficulty lies in the uneven sector sizes. This also results in different densities of the x-components in the different sectors. This design avoids the exploitation of artificial symmetries. The construction is so that in the noisy variant there is no majority class at any point.

We have conducted a parameter study with all nine machines (with linear kernel and without offset term). In the noise-free case we have drawn $\ell = 100$ samples. Since the problem is linearly separable and noise-free, we have tested rather large values of $C = 10^n$ with $n \in \{0, 1, 2, 3, 4\}$ (there are no interesting effects outside this range). The resulting classifiers are shown in Figure 1. The information content of noisy samples is significantly reduced, which is why we use $\ell = 500$ and $n \in \{-4, -3, -2, -1, 0\}$ here (again, results do not change outside this range). The classifier predictions are visualized in Figure 2.

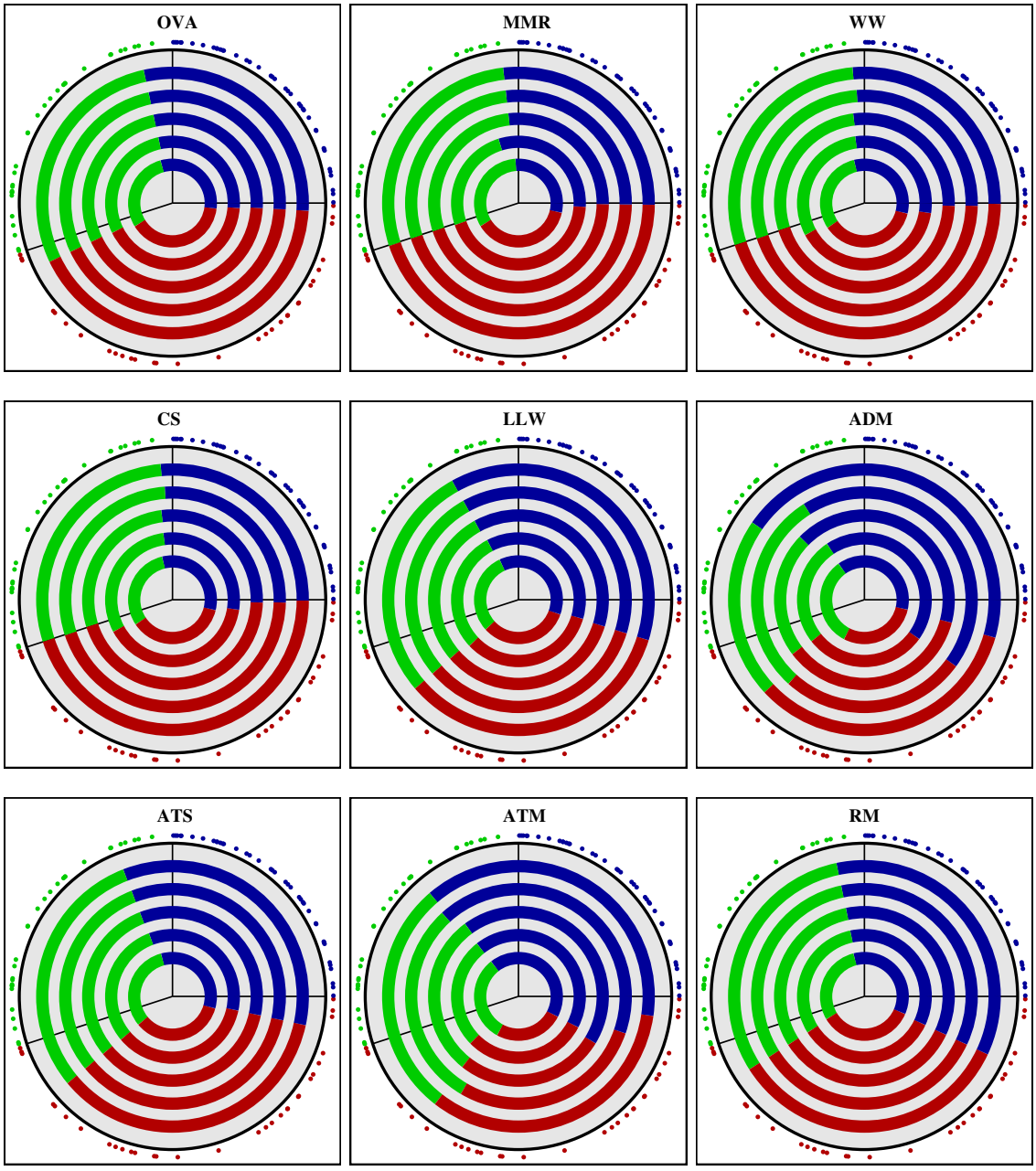

Figure 1: Noiseless circle problem. The sector separators are the decision boundaries of the Bayes-optimal predictor. Classes 1, 2, and 3 are indicated by colors blue, green, and red, respectively. The points on the outside of each graph are the 100 training samples. The colored circles indicate the classifier predictions for $C = 10^n$, $n \in \{0, 1, 2, 3, 4\}$, increasing from inner to outer circles.

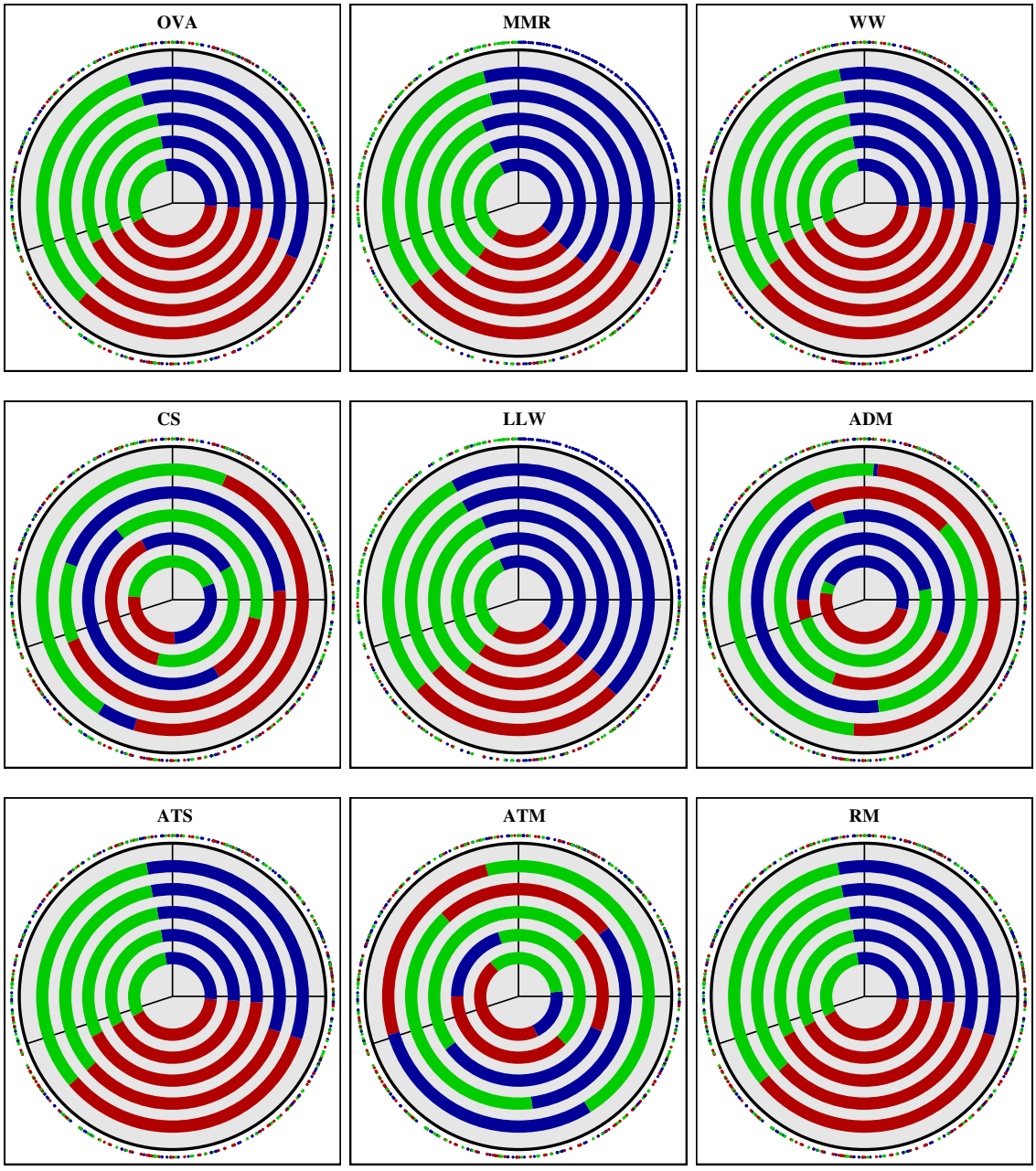

Figure 2: Noisy circle problem. The sector separators are the decision boundaries of the Bayes-optimal predictor. Classes 1, 2, and 3 are indicated by colors blue, green, and red, respectively. The points on the outside of each graph are the 500 training samples. The colored circles indicate the classifier predictions for $C = 10^n$, $n \in \{-4, -3, -2, -1, 0\}$, increasing from inner to outer circles.

Figure 1 shows that the relative margin machines WW and CS excel, in particular for high values of C. Also the MMR machine relying on the self-margin component performs well. In contrast, the absolute margin machines LLW, AMO, ATS, RM, and ATM all fail, over the whole range of parameters (despite the universal consistency of some of the approaches in rich enough feature spaces). The same defect can be observed for OVA, but in weaker form.

The reason for this failure is that the definition of absolute margins does not allow to solve the given problem without violation of the target margin. This holds even for the Bayes-optimal classifier. Thus, absolute margins are not compatible with the form of the decision function. This is obvious for the OVA approach on the circle problem, because one class cannot be separated from the others by a single linear hyperplane through the origin. It turns out that this effect is not a direct consequence of the OVA approach, but of absolute margins in general.

Figure 2 displays the performance in the presence of massive label noise. The performance of the WW machine remains good, but as expected it needs a much smaller value of C. The performance of MMR degrades a bit, while OVA does well in this case. The sub-optimal performance of the LLW machine is largely unchanged, while the closely related ATS and RM profit from the self-margin component (which by itself works well, as shown by the MMR results). Most strikingly, the CS machine with its relative margin as well as the AMO and ATM machines with their absolute margins give random predictions. All three approaches rely on maximum-based aggregation operators. This behavior is perfectly explained by the Fisher consistency results (also reflected in the proof of Theorem 5): Since there is no majority class, all losses favor the trivial solution $w_1 = w_2 = w_3 = 0$. This solution is also optimal for the regularizer. Thus, it is attained for all values of C, and the figures show nothing but numerical noise. In the noisy case the combination of loss components with maximum-based aggregation operators results in guessing performance. Notably this includes the well established CS machine.

In low dimensional feature spaces we see absolute margin machines fail in the noiseless case and maximum-based aggregation operators in the noisy case.[3] Despite the simplicity of the synthetic problems, only a single machine solves both cases, namely the WW approach. The simplistic OVA and MMR machines perform worse, but still satisfactory.

4. Empirical Comparison

This section empirically compares nine approaches to multi-category SVM learning, namely the sequential OVA scheme, the established all-in-one methods WW, CS, LLW, RM, and MMR, as well as the new methods called AMO, ATS, and ATM. All algorithms were implemented in the Shark open source machine learning library (Igel et al., 2008). Source code for reproducing the results can be found in the supplementary material. First, twelve standard benchmark data sets were considered for non-linear SVM learning, and careful model selection was conducted. This already makes our experiments the most extensive comparison of multi-class SVMs so far. Second, additional experiments for linear SVMs are presented later in this section.

3. This result does of course not exclude the existence of a similar problem where WW also fails. However, we did not manage to construct such an instance.

4.1 Multi-class Benchmark Problems

The descriptive statistics of the twelve data sets used to evaluate the non-linear multi-class SVM methods are given in Section D of the supplementary material. All features of all data sets were pre-processed by rescaling to unit variance. This rescaling was done for each split into training and test data individually based on the statistics of the particular training set.

4.2 Model Selection

In all non-linear SVM experiments, Gaussian kernels $k_\gamma(x_1, x_2) = \exp(-\gamma||x_1 - x_2||^2)$ were used. The bandwidth γ of the Gaussian kernel and the regularization parameter C of the machine were determined by nested grid search. Repeated cross-validation was employed as model selection criterion. Five-fold cross-validation was repeated ten times using ten independent random splits into five folds. This stabilizes the model selection procedure especially for small data sets. Candidate parameters were evaluated on the 5×10 validation subsets and the configuration yielding the best average performance was chosen. If any of the selected model parameters was at the grid boundary, then the grid was extended accordingly.

We set the initial grid to $\gamma \in \{2^{-12+3i} \,|\, i = 0, 1, \ldots, 4\}$ and $C \in \{2^{3i} \,|\, i = 0, 1, \ldots, 4\}$. Let (γ_0, C_0) denote the parameter configuration picked in the first stage. Then in the second stage the parameters were further refined on the grid $\gamma \in \{2^i \cdot \gamma_0 \,|\, i = -2, -1, 0, 1, 2\}$ and $C \in \{2^i \cdot C_0 \,|\, i = -2, -1, 0, 1, 2\}$. For linear SVMs we applied the same approach restricted to the complexity control parameter C. The hyperparameters as determined by the grid-searches are given in Section E of the supplementary material.

Table 3 provides a comparison reflecting the computational demand of parameter tuning. It lists the total training time for computing the 5-fold cross validation error on a 5×5 grid of the outer grid search loop, centered on the optimal parameters. Training times differ by several orders of magnitude depending on the SVM variant.

Data set	OVA	MMR	WW	CS	LLW	AMO	ATS	ADS	RM
Car	36.45	12.89	131.7	1027	6112	40231	2399	34046	4106
Glass	1.10	0.32	2.05	318.6	48.50	369.2	247.1	186.1	69.87
Iris	1.41	0.13	13.02	1.30	25.58	4.63	647.7	35.01	243.1
Red wine	53.87	10.51	57.68	2574	7354	75159	6486	67000	2235

Table 3: Time in seconds for computing the 5-fold cross validation error on a single core over a 5×5 grid around the optimal parameter values.

4.3 Evaluation

The multi-class SVMs were evaluated as follows. Using the best parameters found during model selection, 100 machines were trained on 100 random splits into training and test data (preserving the original set sizes).[4] In this way properties of the test error distribution

4. Doing the full model selection 100 times would be the better setup, however, it was computationally too demanding.

can be estimated. We report empirical mean and standard deviation. Furthermore, the 100 repetitions allow to test results for significant differences. For each problem we have compared each machine to the best performing one with a paired U-test (at significance level $\alpha = 0.01$). We are not interested in the test results per se; instead, the tests provide thresholds for judging performance as competitive or inferior. It must be noted that the 100 repetitions share the same data and are thus not independent. This means that the confidence level needs adjustment. However, the U-test still provides a meaningful thresholding mechanism.

4.4 Stopping Condition

For a fair comparison of training times of different SVMs, it is of importance to choose comparable stopping criteria for the quadratic programming. Unfortunately, this is hardly possible in the experiments presented in this study, because the quadratic programs differ. However, in the case of just two classes all machines solve the same problem. Therefore the stopping condition was selected such that for a binary problem these machines would give the same solution. The common threshold of $\varepsilon = 10^{-3}$ on violations of the Karush-Kuhn-Tucker (KKT) conditions of optimality was used as stopping criterion (Bottou and Lin, 2007).

To rule out artifacts of this choice several experiments were repeated with an accuracy of $\varepsilon = 10^{-5}$. These experiments did not result in improved performance. Thus, the choice of the stopping criterion may in the worst case influence training times, but not the test accuracies reported in Section 4.5 and Section 4.6.

For all non-linear machines, the maximum number of SMO iterations was limited to 10.000 times the number of dual variables. The limit for linear machines was 10 times higher.[5] This was necessary to keep the grid searches computationally tractable. However, the few parameter configurations that had hit this limit corresponded to "degenerated" machines (i.e., bad solutions), typically with far too small γ and too large C. Thus, this proceeding did not influence the outcome of the model selection process.

4.5 Non-linear SVM Results

The test accuracies of all nine machines are listed in Table 4. The table also indicates whether a paired U-test judges the 100 test accuracies as significantly worse than the machine with the best performance. The relative accuracies of the 9 machines are represented graphically in Figure 3.

A similar plot in Figure 4 displays training times. The training times varied vastly with the choice of the hyper-parameters, that is, they were strongly dependent on the outcome of the model selection procedure. In some cases there exist configurations with similar prediction performance but different optimization times. Thus, the timing results were subject to non-negligible noise and should thus be interpreted with care.

5. The higher limit accounts for the fact that single iterations are less efficient, because in the algorithm proposed by Hsieh et al. (2008) and Fan et al. (2008) and refined by Glasmachers and Doğan (2013, 2014) the choice of the active variable does not take gradient and gain information into account, see Section 4.6.

	OVA	MMR	WW	CS	LLW	AMO	ATS	ATM	RM
Abalone	26.89	26.51	**27.51**	21.33	26.65	20.42	26.64	20.58	22.02
	± 1.26	± 1.31	± 1.19	± 0.95	± 1.25	± 1.13	± 1.24	± 1.43	± 1.09
Car	98.37	97.58	**98.62**	**98.61**	98.14	98.11	98.14	98.09	98.24
	± 0.72	± 0.82	± 0.72	± 0.71	± 0.79	± 0.80	± 0.77	± 0.74	± 0.62
Glass	**68.69**	**68.03**	**68.78**	**69.03**	68.03	**69.52**	68.38	**69.53**	**69.09**
	± 4.61	± 5.33	± 5.24	± 4.48	± 4.80	± 5.01	± 4.89	± 4.95	± 5.68
Iris	**96.66**	94.71	**96.35**	95.26	**96.22**	95.08	**95.86**	95.11	**96.02**
	± 2.43	± 2.99	± 2.59	± 2.88	± 2.51	± 2.57	± 2.77	± 2.52	± 2.64
Opt. digits	**98.80**	98.25	**98.77**	98.76	**98.85**	97.87	**98.84**	**98.84**	**98.90**
	± 0.25	± 0.26	± 0.24	± 0.24	± 0.22	± 0.29	± 0.22	± 0.22	± 0.23
Page blocks	96.73	96.65	**96.83**	**96.78**	96.69	96.63	96.69	96.63	**96.75**
	± 0.46	± 0.46	± 0.42	± 0.44	± 0.47	± 0.43	± 0.45	± 0.43	± 0.37
Sat	**92.19**	91.75	**92.19**	**92.19**	**92.13**	**92.14**	**92.15**	**92.14**	**92.20**
	± 0.53	± 0.54	± 0.53	± 0.54	± 0.53	± 0.49	± 0.52	± 0.48	± 0.50
Segment	**96.41**	96.22	**96.39**	**96.36**	**96.43**	**96.41**	**96.41**	**96.42**	**96.60**
	± 0.73	± 0.71	± 0.73	± 0.60	± 0.73	± 0.68	± 0.73	± 0.75	± 0.65
Soy bean	89.58	87.24	89.13	89.88	89.81	88.86	**90.41**	88.72	**91.16**
	± 3.04	± 3.40	± 2.96	± 3.11	± 3.32	± 3.00	± 3.23	± 3.21	± 2.52
Vehicle	**84.90**	73.94	84.28	83.99	**84.86**	83.74	**84.83**	83.84	**84.71**
	± 1.88	± 2.17	± 2.00	± 1.90	± 1.99	± 2.30	± 2.13	± 2.30	± 1.81
Red wine	**63.93**	**64.13**	**63.87**	**63.90**	**63.91**	**64.06**	**63.93**	**64.06**	**63.99**
	± 2.07	± 1.83	± 1.91	± 2.02	± 2.13	± 2.03	± 2.07	± 2.03	± 1.87
White wine	64.03	63.96	**64.86**	64.04	64.02	**64.83**	64.39	**64.83**	**64.27**
	± 1.12	± 1.18	± 1.17	± 1.15	± 1.11	± 1.17	± 1.09	± 1.17	± 1.05

Table 4: Classification accuracies in percent of correctly classified test examples. The table lists mean values and standard deviations. In each row bold numbers indicate that the result is not significantly worse than the best one (paired U-test, $\alpha = 0.01$).

In many applications not only training times but also testing times are of relevance. The time it takes to compute the prediction of an SVM classifier is usually dominated by the kernel computations. In the multi-class case this means that sparsity in the dual variables is not enough: only variables that do not appear in any of the expansions of the weight vectors $w_1, \ldots, w_d$ can be dropped in the evaluation of the classifier. We count every training example that is used in the construction of any of the weight vectors as a support vector. The fraction of support vectors is reported in Figure 5.

4.6 Linear SVM Results

In recent years there has been increased interest in linear SVM classifiers. These are a viable alternative to non-linear SVMs in particular in high-dimensional input spaces commonly found, for instance, in text mining and bioinformatics problems. A powerful coordinate descent solver for the dual linear SVM problem was proposed by Hsieh et al. (2008) and

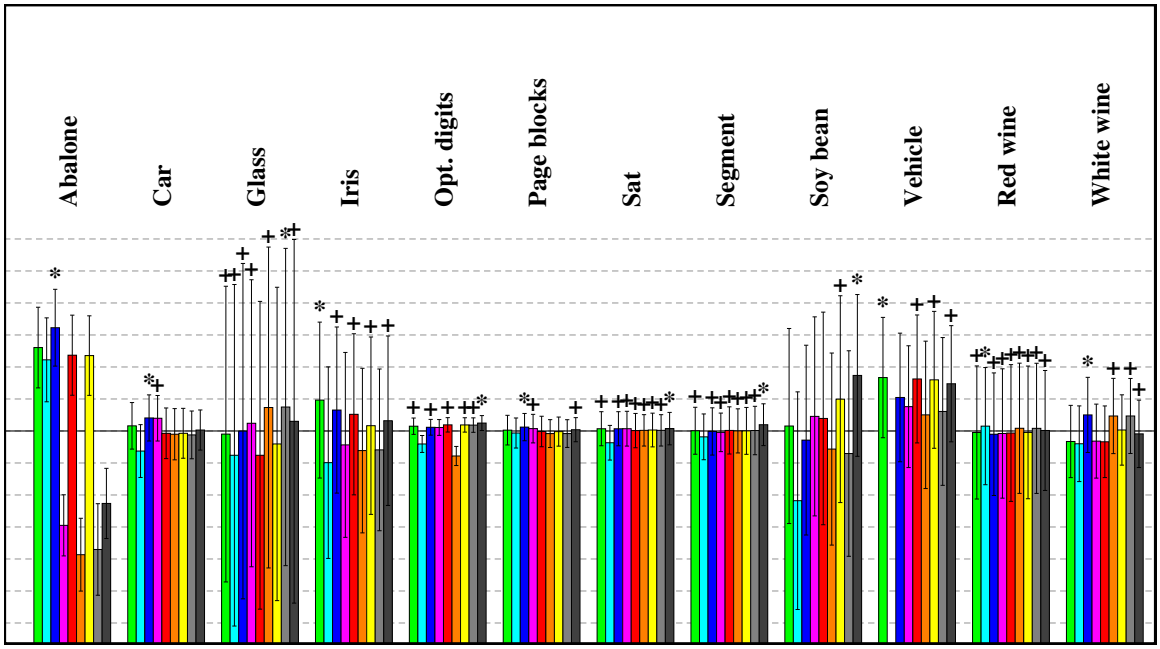

Figure 3: Relative accuracies of all nine machines. The average performance per data set defines the mid-line. Each dashed line marks 1% of deviation. The colored bars denote (relative) classification accuracy (higher is better). The error bars indicate the standard deviation across multiple training/test splits. The best result is indicated with a star ($*$). Statistically insignificant differences to the best solution (U-test, $\alpha = 0.01$) are marked with a plus sign ($+$). Colors represent models as follows: green (OVA), cyan (MMR), blue (WW), pink (CS), red (LLW), orange (AMO), yellow (ATS), light gray (ATM), and dark gray (RM).

Fan et al. (2008) and has been recently extended by Glasmachers and Doğan (2013, 2014). All linear SVM results reported in this section were obtained with an extension of the latter technique to the multi-class case.

For the linear kernel case we have extended the testbed beyond the twelve problems to data sets with different characteristics, which are typically used to evaluate linear multi-class models, see supplementary material (Section E). These data sets are available from the libsvm data collection.[6] We have included problems with rather low-dimenional as well as extremely high-dimensional input spaces. The test errors of the nine linear multi-class SVMs are summarized in Table 5.

6. `http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/`

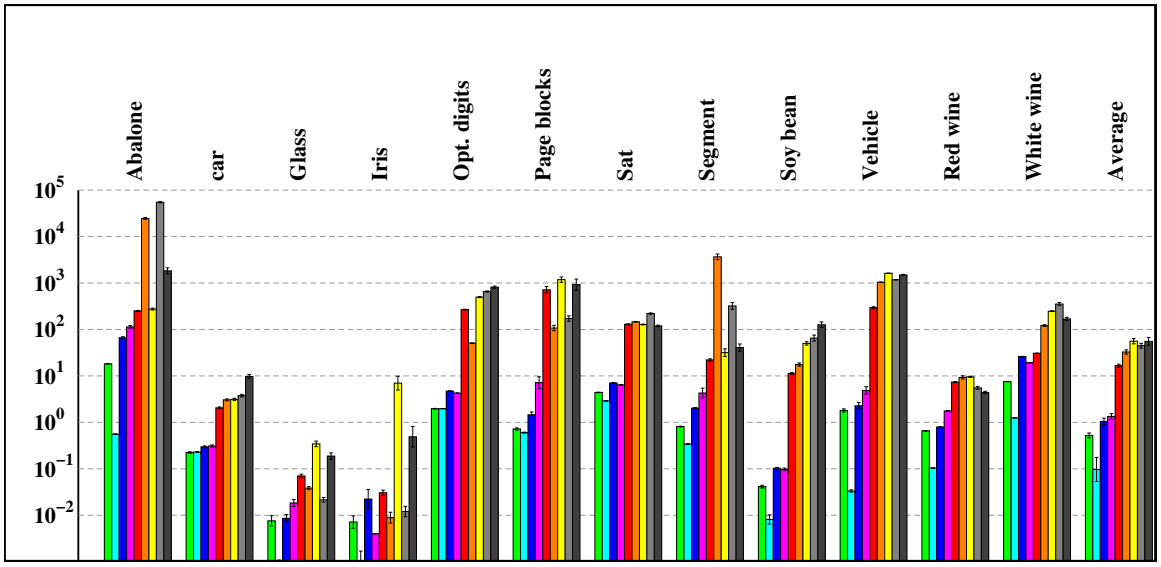

Figure 4: Average training times (logarithmic scale) of all nine machines, in seconds (lower is better). Colors represent models as follows: green (OVA), cyan (MMR), blue (WW), pink (CS), red (LLW), orange (AMO), yellow (ATS), light gray (ATM), and dark gray (RM). The (geometric) average training time across all 12 problems is found on the right.

4.7 Discussion

Our results allow to compare the nine different machines from different angles. We generally view accuracy as a primary goal and training speed and prediction speed (related to sparsity of the solution) as secondary criteria.

The accuracy results of the non-linear SVMs presented in Table 4 and Figure 3 were mixed. The prediction performances of most machines were rather close to each other. This is in accordance with the findings of Hsu and Lin (2002) and Rifkin and Klautau (2004). Larger deviations could be observed on the problems Abalone and Vehicle. The maximum combination operator absolute margin machines AMO and ATM fell behind on the Abalone problem, while the self-margin MMR machine did not give satisfactory performance on the Vehicle problem. Overall the WW and RM machines performed best, for example, when counting the number of data sets where the solution was either best or not significantly worse than the best result. The MMR machines showed the weakest performance. However, it appears than most machines give rather competitive results; OVA, CS, LLW and ATS are hard to put in a clear order, and they are only slightly worse than WW and RM (if at all).

This picture changes completely for linear SVMs. The results presented in Table 5 reveal drastic differences between machines, as already found by Hsu and Lin (2002). There were only two problems where all machines achieved roughly comparable performance, namely News-20 and Sector. These problems come with extremely high-dimensional feature spaces and are thus similar in character to the non-linear SVM problems. Here the classification

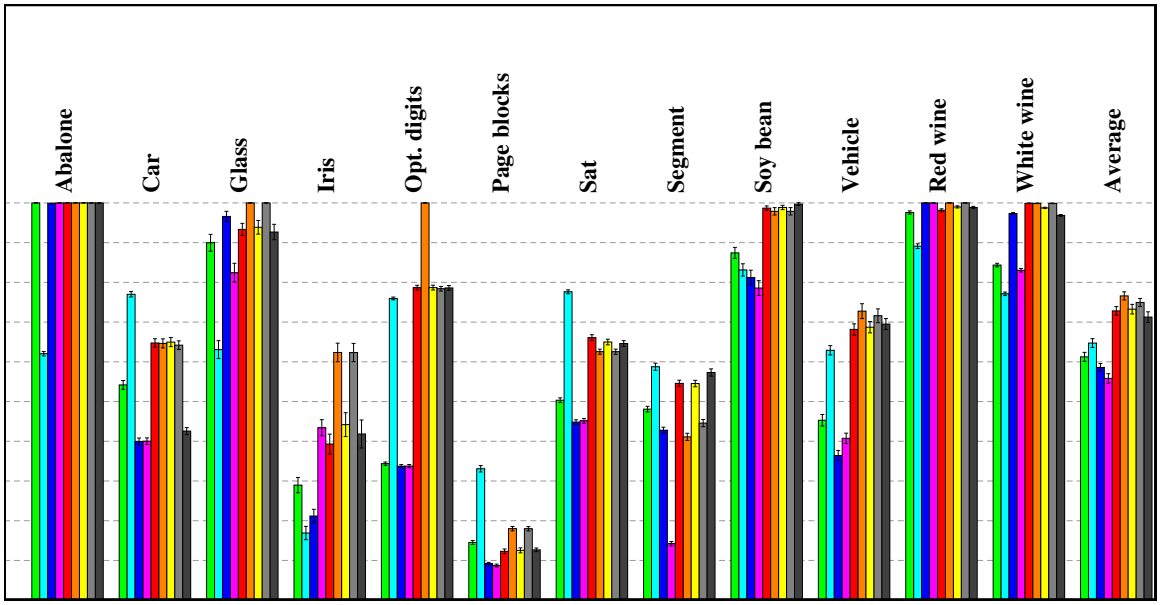

Figure 5: Average fraction of support vectors for all nine machines (lower is better). A bar hitting the top indicates that all points are support vectors, while a bar vanishing at the bottom indicates that no points were used as support vectors. Colors represent models as follows: green (OVA), cyan (MMR), blue (WW), pink (CS), red (LLW), orange (AMO), yellow (ATS), light gray (ATM), and dark gray (RM). The average fraction of support vectors across all 12 problems is found on the right.

calibrated LLW and RM approaches and the other absolute margin machines worked about as well as in the non-linear case. Most other problems have rather low-dimensional feature spaces. Thus, the setting comes conceptually closer to the synthetic circle problems presented in Section 3. This explains the complete breakdown of some methods on some of the problems, for example, all absolute margin machines on the letter data. In accordance with the findings in Section 3, WW and CS outperformed the classification calibrated LLW, ATS and RM machines. The linear SVM results also clearly display the problems of the OVA and MMR approaches. While OVA works well at least in some cases, we have to dismiss the MMR machine completely for its consistently inferior performance. The WW and CS machines clearly performed best, again with a slight but insignificant edge for WW.

The secondary goals of training and test times give a yet completely different picture. The rightmost column in Figure 4 summarizes the training times. The measurements are in accordance with the asymptotical results in Table 2. Not surprisingly, the MMR machine was clearly fastest to train. It is followed by OVA, which is, however, closely followed by WW and CS. It is clear from the asymptotical considerations that the differences between OVA and the all-in-one machines will get more pronounced with increasing number of classes. The WW machine was not slower to train than the popular CS approach, actually

	OVA	MMR	WW	CS	LLW	AMO	ATS	ATM	RM
Covertype	50.59 ± 5.49	23.55 ± 0.19	**70.55** ± 0.09	45.73 ± 5.88	21.87 ± 23.19	47.31 ± 6.79	18.44 ± 17.71	51.19 ± 7.52	69.61 ± 0.40
Letter	63.69 ± 0.48	21.66 ± 1.21	69.39 ± 0.63	**76.59** ± 0.61	12.78 ± 0.40	6.54 ± 1.88	17.44 ± 0.79	6.78 ± 2.04	21.39 ± 1.84
News-20	85.36 ± 0.32	78.10 ± 0.53	85.13 ± 0.15	85.17 ± 0.32	**86.71** ± 0.39	85.65 ± 0.32	**86.69** ± 0.37	85.61 ± 0.35	**86.61** ± 0.37
Sector	94.53 ± 0.22	91.97 ± 0.30	94.10 ± 0.33	94.80 ± 0.29	94.82 ± 0.28	**95.09** ± 0.30	94.82 ± 0.28	**95.09** ± 0.30	94.74 ± 0.37
Usps	94.50 ± 0.39	87.84 ± 0.75	94.46 ± 0.57	**95.26** ± 0.46	78.18 ± 5.27	40.57 ± 2.15	80.60 ± 0.65	40.27 ± 3.27	88.08 ± 0.69
Abalone	18.95 ± 0.86	15.80 ± 0.90	**21.70** ± 1.30	14.12 ± 1.64	16.56 ± 1.17	8.36 ± 2.34	18.33 ± 0.76	8.78 ± 2.31	18.86 ± 1.38
Car	71.69 ± 1.73	70.47 ± 1.86	**73.76** ± 1.68	**73.15** ± 2.02	65.34 ± 12.17	70.43 ± 1.83	72.14 ± 1.71	70.49 ± 1.94	72.52 ± 1.58
Glass	**56.98** ± 6.44	43.87 ± 7.74	**61.93** ± 6.63	**61.93** ± 6.04	46.78 ± 6.77	28.43 ± 11.98	49.51 ± 5.78	32.13 ± 12.04	51.50 ± 6.77
Iris	91.11 ± 4.85	65.34 ± 6.07	**95.88** ± 1.71	**91.76** ± 7.18	74.65 ± 7.52	67.32 ± 6.66	82.05 ± 5.94	67.32 ± 6.66	85.14 ± 4.96
Opt. Digits	95.98 ± 0.60	89.08 ± 1.08	96.03 ± 0.37	**96.42** ± 0.37	73.56 ± 2.11	18.45 ± 4.77	81.32 ± 1.62	19.11 ± 4.41	92.10 ± 0.66
Page Blocks	70.44 ± 21.20	48.99 ± 24.39	91.14 ± 5.41	**94.20** ± 2.34	93.22 ± 1.02	**93.87** ± 1.58	**93.81** ± 1.44	**93.74** ± 1.71	**93.58** ± 0.75
Sat	75.04 ± 0.96	31.04 ± 0.95	**77.40** ± 3.00	66.87 ± 9.90	51.47 ± 9.01	47.59 ± 6.89	61.21 ± 0.95	45.49 ± 6.97	63.47 ± 1.29
Segment	**92.54** ± 0.75	50.90 ± 2.52	**92.43** ± 2.13	**92.43** ± 2.13	74.50 ± 1.32	67.58 ± 12.21	76.30 ± 4.26	67.58 ± 12.21	81.94 ± 2.48
Soy Bean	**90.65** ± 3.03	61.41 ± 7.80	87.75 ± 3.16	83.49 ± 5.80	77.95 ± 9.97	38.87 ± 6.42	66.74 ± 4.52	39.86 ± 5.84	79.20 ± 4.23
Vehicle	52.02 ± 11.98	52.33 ± 2.98	72.75 ± 4.13	72.75 ± 4.13	63.21 ± 10.63	63.21 ± 10.63	63.21 ± 10.63	63.21 ± 10.63	**76.42** ± 2.38
Red wine	53.38 ± 2.63	23.63 ± 5.68	**58.37** ± 1.69	55.61 ± 2.47	57.26 ± 2.02	36.58 ± 10.95	56.29 ± 1.64	36.27 ± 9.67	**57.81** ± 1.68
White wine	50.73 ± 1.27	19.20 ± 9.68	**51.78** ± 1.24	50.85 ± 1.12	46.44 ± 1.74	30.03 ± 8.21	51.16 ± 0.98	27.06 ± 7.21	51.41 ± 1.27

Table 5: Classification accuracies of the linear machines in percent of correctly classified test examples. The table lists mean values and standard deviations. In each row, bold numbers indicate that the result is not significantly worse than the best one (paired U-test, $\alpha = 0.01$).

it was slightly faster on average in our experiments.[7] Finally, the various absolute margin machines took at least an order of magnitude longer to train than WW and CS.

7. Here we only present our results for training without bias parameters. However, this statement also holds true for training with bias parameters.

At least for expensive kernel functions prediction speed is governed by the sparsity of the resulting model. Figure 5 reveals that sparsity is primarily a property of the problem, while the chosen loss function has only a minor impact. Overall, multi-class models are expected to be less sparse than, for instance, binary SVMs since a training point can be dropped only if it does not appear in any of the d weight vectors. We observed significant sparsity only for the page blocks data. In comparison relative margin machines gave slightly more sparse models than absolute margin machines. The MMR self-margin machine did not stand out in this category. However, only few problems had sparse solutions at all, and inter-problem spread was much higher than differences between methods. Hence we do not see good reasons to reject any machine based on these data.

In summary, the WW and CS machines showed robust performance across all disciplines, with WW slightly ahead of CS. The MMR machine performed worst in terms of accuracy, but was fastest to train. In most cases there are no good reasons for absolute margin machines.

The results have an interesting interpretation in the unified view presented in Section 2. Differences between self, relative and absolute machine machines were much more pronounced than differences between aggregation operators. Relying only on the single self-margin component (MMR) is computationally attractive but results in rather poor prediction accuracy. Relative margins give strong machines. Consistent absolute margin machines with sum-aggregation are a viable alternative in sufficiently high-dimensional feature spaces, although they are usually costly to train.

It is questionable whether a connection between asymptotic consistency and practical performance exists. This is partially observable in sum-aggregation absolute margin machines performing well only in high-dimensional feature spaces. However, this condition does not seem to be necessary, since the other absolute margin machines show the same trend and also the inconsistent CS machine performs well, except for the noisy circle problem (see Section 3.3). Overall, prediction performance seems to be more closely related to the applied margin concept than to uniform consistency.

Based on our empirical findings, we can recommend relative margin machines for almost all applications. We prefer the WW machine over the CS approach for its slightly more stable performance and its theoretical properties (consistency of WW with Gaussian kernel, see Theorem 8, and breakdown of CS on the noisy circle problem).

5. Conclusion

This article provides a novel unified view on multi-class SVMs, showing that the various approaches are not as different as they seem. All popular algorithms reduce to the standard SVM for binary classification problems, however, they differ along several dimensions when applied to more than two classes. The machines can be formulated based on a vector-valued decision function with as many components as classes. We have highlighted margin concepts, named relative and absolute margins (including the self-margin component), that describe whether the components of the decision function are optimized relative to each other or not. Independent of the margin concept, the machines can employ either sum- or maximum-based aggregation operators for combining margin violations into a scalar loss. A further distinction can be made based on whether a machine fulfills (or even enforces) that

the components of the decision function sum to zero or not. All machines can be formulated with and without bias term (i.e., for linear and affine hypotheses in feature space), and we derived a unified formulation of the corresponding dual optimization problems in terms of margin concepts and aggregation operators.

Our unifying view pointed at three combinations of these features that had not been investigated. These missing machines were derived and evaluated. The new classifiers, named AMO-, ATS- and ATM-SVM, all use the absolute margin concept but consider different aggregates of the margin violations. The ATS machine, which can be viewed as one-vs-all classification with imposed sum-to-zero constraint and is closely related to RM SVMs, turns out to be Fisher consistent.

The LLW, RM, and ATS SVMs rely on a classification calibrated loss function. However, we have shown that a non-classification calibrated training loss function does not need to be a principal problem for an SVM, since the overall *regularized* empirical risk minimizer can still be consistent. This is the case for the maximum margin regression (MMR), one-vs-all (OVA), and Weston & Watkins (WW) machines. One may argue that for large enough amounts of data one should prefer consistent models over inconsistent large margins. It is provably impossible to "repair" the inconsistency of the Crammer & Singer (CS) multi-class SVM, AMO and ATM losses by means of SVM-style regularization.[8] The MMR and WW machines, and even the OVA scheme (lacking the sum-to-zero property), turn out to be consistent classifiers when regularized properly. We argue that these results are best understood in the context of our unified view. All inconsistent machines apply the maximum operator for aggregation, while none of the consistent machines relies on this operator. Thus, our results hint at a fundamental difference between the sum and maximum aggregation operators.

Our unified learning problems lend themselves to efficient optimization algorithms. These allowed us to perform a thorough empirical comparison of the various multi-class SVMs and to draw conclusions about training times and generalization performance. We considered the LLW and RM machines in our experiments, which are typically omitted in comparison studies for training time reasons (e.g., Ding and Dubchak, 2001; Rifkin and Klautau, 2004; Statnikov et al., 2005), and we could perform proper model selection. We compared nine different multi-class SVM with Gaussian and linear kernel on a large collection of benchmark problems. The unified view enables a meaningful interpretation of the results by relating them to margin concepts and aggregation operators. We find that relative margin machines are superior for the linear kernel, and that sum-based aggregation generally outperforms the maximum approach. Our empirical and theoretical results support the following recommendations:

The standard all-in-one multi-class WW SVM should be used as the default since it gives robust performance at moderate training times. It can be trained at least as fast as the popular CS SVM while giving at least as good generalization results and having nicer theoretical properties (which can matter in practice as shown on artificial test problems and when using a linear kernel). The simplistic OVA method can be a viable alternative; it delivers slightly worse results on average while being a little faster to train. If training time is a major concern then the MMR machine may seem like the best option. However,

8. In practice, the performance of the CS machine is very similar to the WW machine. Thus, more often than not this effect does not seem to be dominant.

its performance can be so poor that even sub-sampling the data in combination with WW or OVA may be a better approach.

With universal kernels and in general in extremely high-dimensional feature spaces the LLW and RM machines with their classification calibrated losses are usually on par with WW and CS and thus promising candidates. However, training may be infeasible for large data sets, for its more than ten times longer training times compared to WW and CS.

Acknowledgments

Tobias Glasmachers acknowledges support by the Mercator Research Center Ruhr (MERCUR), project Pr-2013-0015, and Christian Igel from The Danish Council for Independent Research (DFF) through the project Surveying the sky using machine learning (SkyML).

References

E. L. Allwein, R. E. Schapire, and Y. Singer. Reducing multiclass to binary: A unifying approach for margin classifiers. *Journal of Machine Learning Research*, 1:113–141, 2001.

A. Bordes, N. Usunier, and L. Bottou. Sequence labelling SVMs trained in one pass. In W. Daelemans, B. Goethals, and K. Morik, editors, *Machine Learning and Knowledge Discovery in Databases*, volume 5211 of *LNCS*, pages 146–161. Springer, 2008.

B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory (COLT 1992)*, pages 144–152. ACM, 1992.

L. Bottou and C. J. Lin. Support vector machine solvers. In L. Bottou, O. Chapelle, D. DeCoste, and J. Weston, editors, *Large Scale Kernel Machines*, pages 1–28. MIT Press, 2007.

E. J. Bredensteiner and K. P. Bennett. Multicategory classification by support vector machines. *Computational Optimization and Applications*, 12(1):53–79, 1999.

C.-C. Chang and C.-J. Lin. LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3):27, 2011.

C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

K. Crammer and Y. Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of Machine Learning Research*, 2:265–292, 2002.

L. Devroye, L. Gyorfi, and G. Lugosi. *A probabilistic theory of pattern recognition*. Springer Verlag, 1996.

T. G. Dietterich and G. Bakiri. Solving multiclass learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research*, 2(263):286, 1995.

C. H. Q. Ding and I. Dubchak. Multi-class protein fold recognition using support vector machines and neural networks. *Bioinformatics*, 17(4):349, 2001.

Ü. Doğan, T. Glasmachers, and C. Igel. A note on extending generalization bounds for binary large-margin classifiers to multiple classes. In P. A. Flach, T. De Bie, and N. Cristianini, editors, *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD 2012)*, volume 7523 of *LNCS*, pages 122–129. Springer, 2012.

R. E. Fan, K.W. Chang, C. J. Hsieh, X.R. Wang, and C. J. Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008.

T. Glasmachers and Ü. Doğan. Accelerated coordinate descent with adaptive coordinate frequencies. In *Proceedings of the Fifth Asian Conference on Machine Learning (ACML)*, JMLR W&CP, 2013.

T. Glasmachers and Ü. Doğan. Coordinate Descent with Online Adaptation of Coordinate Frequencies. Technical Report arXiv:1401.3737, arxiv.org, 2014.

Y. Guermeur. VC theory for large margin multi-category classifiers. *Journal of Machine Learning Research*, 8:2551–2594, 2007.

Y. Guermeur. A generic model of multi-class support vector machine. *International Journal of Intelligent Information and Database Systems (IJIIDS)*, 6:555–577, 2012.

T. Hastie and R. Tibshirani. Classification by pairwise coupling. *Annals of Statistics*, 26 (2):451–471, 1998.

S. I. Hill and A. Doucet. A framework for kernel-based multi-category classification. *Journal of Artificial Intelligence Research*, 30(1):525–564, 2007.

C. J. Hsieh, K.W. Chang, C. J. Lin, S.S. Keerthi, and S. Sundararajan. A dual coordinate descent method for large-scale linear SVM. In *Proceedings of the 25th International Conference on Machine learning (ICML)*, pages 408–415. ACM, 2008.

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.

C. Igel, T. Glasmachers, and V. Heidrich-Meisner. Shark. *Journal of Machine Learning Research*, 9:993–996, 2008.

T. Joachims. Making large-scale SVM learning practical. In B. Schölkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods – Support Vector Learning*, chapter 11, pages 169–184. MIT Press, 1998.

W. Kienzle and B. Schölkopf. Training support vector machines with multiple equality constraints. In *Machine Learning: ECML 2005*, volume 3720 of *LNCS*, pages 182–193. Springer, 2005.

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*, 99(465):67–82, 2004.

Y. Liu. Fisher consistency of multicategory support vector machines. In M. Meila and X. Shen, editors, *Eleventh International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 2 of *JMLR W&P*, pages 289–296, 2007.

Y. Liu and M. Yuan. Reinforced multicategory support vector machines. *Journal of Computational and Graphical Statistics*, 20(4):901–919, 2011.

A. Passerini, M. Pontil, and P. Frasconi. New results on error correcting output codes of kernel machines. *IEEE Transations on Neural Networks*, 15(1):45–54, 2004.

J. C. Platt, N. Cristianini, and J. Shawe-Taylor. Large margin DAGs for multiclass classification. In S. A. Solla, T. K. Leen, and K.-R. Müller, editors, *Advances in Neural Information Processing Systems*, volume 12, pages 547–553. MIT Press, 2000.

R. Rifkin and A. Klautau. In defense of one-vs-all classification. *Journal of Machine Learning Research*, 5:101–141, 2004.

A. Statnikov, C. F. Aliferis, I. Tsamardinos, D. Hardin, and S. Levy. A comprehensive evaluation of multicategory classification methods for microarray gene expression cancer diagnosis. *Bioinformatics*, 21(5):631, 2005.

I. Steinwart. On the influence of the kernel on the consistency of support vector machines. *Journal of Machine Learning Research*, 2:67–93, 2002a.

I. Steinwart. Support vector machines are universally consistent. *Journal of Complexity*, pages 768–791, 2002b.

I. Steinwart and A. Christmann. *Support vector machines*. Springer, 2008.

S. Szedmak, J. Shawe-Taylor, and E. Parado-Hernandez. Learning via linear operators: Maximum margin regression. Technical report, PASCAL, Southampton, UK, 2006.

A. Tewari and P. L. Bartlett. On the consistency of multiclass classification methods. *Journal of Machine Learning Research*, 8:1007–1025, 2007.

V. Vapnik. *Statistical Learning Theory*. John Wiley and Sons, 1998.

J. Weston and C. Watkins. Support vector machines for multi-class pattern recognition. In M. Verleysen, editor, *Proceedings of the Seventh European Symposium On Artificial Neural Networks (ESANN)*, pages 219–224. Evere, Belgium: d-side publications, 1999.

Y. Wu and Y. Liu. Robust truncated hinge loss support vector machines. *Journal of the American Statistical Association*, 102(479):974–983, 2007.

H. Zou, J. Zhu, and T. Hastie. The margin vector, admissible loss and multi-class margin-based classifiers. *Annals of Applied Statistics*, 2:1290–1306, 2008.