Optimization Methods and Software
Vol. 00, No. 00, Month 200x, 1–18

A NEW MULTI-CLASS SUPPORT VECTOR ALGORITHM[†]

PING ZHONG[a] and MASAO FUKUSHIMA[b,*]

[a] *Faculty of Science, China Agricultural University, Beijing, 100083, China;*
[b] *Department of Applied Mathematics and Physics, Graduate School of Informatics, Kyoto University, Kyoto, 606-8501, Japan*

(Received 18 June 2004; Revised 22 November 2004; In final form 10 February 2005)

Multi-class classification is an important and on-going research subject in machine learning. In this article, we propose a new support vector algorithm, called ν-K-SVCR, for multi-class classification based on ν-SVM. ν-K-SVCR has parameters that enable us to control the numbers of support vectors and margin errors effectively, which is helpful in improving the accuracy of each classifier. We give some theoretical results concerning the significance of the parameters and show the robustness of classifiers. In addition, we have examined the proposed algorithm on several benchmark data sets and artificial data sets, and our preliminary experiments confirm our theoretical conclusions.

Keywords: Machine learning; Multi-class classification; ν-SVM

1 INTRODUCTION

Multi-class classification, as an important problem in data mining and machine learning, refers to the construction of an approximation $\hat{F}$ of an unknown function F defined from an input space $\mathcal{X} \subset R^N$ onto an unordered set of classes $\mathcal{Y} = \{\Theta_1, \cdots, \Theta_K\}$ based on independently and identically distributed (i.i.d.) data

$$\mathcal{T} = \{(\mathbf{x}_p, \theta_p)\}_{p=1}^{l} \subset \mathcal{X} \times \mathcal{Y}. \quad (1)$$

[†] This work was supported in part by the Scientific Research Grant-in-Aid from Japan Society for the Promotion of Science and the National Science Foundation of China Grant No. 10371131.

[*] Corresponding author. Email: fuku@amp.i.kyoto-u.ac.jp

ISSN 1055-6788 print/ ISSN 1029-4937 online ©200x Taylor & Francis Ltd
DOI: doi

Support vector machines (SVMs) is a useful tool for classification. At present, most existing SVMs are restricted to binary classification. However, in general, real world learning problems require examples to be mapped into one of several possible classes. How to effectively extend binary SVMs to multi-class classification is still an on-going research issue. Currently there are roughly two types of SVM-based approaches to solve the multi-class classification problems. One is the "decomposition-reconstruction" architecture approach [1, 5, 7, 9–12, 22] that makes direct use of binary SVMs to tackle the tasks of multi-class classification, while the other is the "all-together" approach [3, 6, 14, 22, 23] that solves the multi-class classification problems by considering all examples from all classes in one optimization formulation. In this article, we propose a new algorithm for multi-class classification with decomposition-reconstruction architecture.

The decomposition-reconstruction architecture approach first uses a decomposition scheme to transform a K-partition $F : \mathcal{X} \rightarrow \mathcal{Y}$ into a series of dichotomizers $f_1, \cdots, f_L$, and then uses a reconstruction scheme to fuse the outputs of all classifiers for a particular example and assign it to one of the K classes.

Among the decomposition schemes frequently used, there are the 'one-against-all (1-a-a)' method [5,22], the 'one-against-one (1-a-1)' method [11,12] and the 'error-correcting output code (ECOC)' method [1, 7, 9, 10]. The 1-a-a method constructs K binary SVM models with the ith one being trained by labelling $+1$ to the examples in the ith class and -1 to all the other examples. The 1-a-1 method requires $K(K-1)/2$ binary class machines, one for each pair of classes. That is, the machine associated with the pair (i, j) concentrates on the separation of class Θ_i and class Θ_j while ignoring all the other examples. The ECOC method applies some ideas from the theory of error correcting code to choose a collection of binary classifiers for training . Then the ECOC method aggregates the results obtained by the binary classifiers to assign each example to one of the K classes. The 1-a-1 method is reported to offer better performance than the 1-a-a method empirically [1, 10, 13]. However, the 1-a-a method has recently been pointed out to have as good performance as other approaches [19].

The usual reconstruction schemes consist of voting, winner-takes-all and tree-structure [18]. Other combinations are made by considering the estimates of a posteriori probabilities for machines' outputs or by adapting the SVM to produce a posteriori class probabilities [16, 17].

Recently, K-SVCR algorithm has been proposed by combining SV classification and SV regression in [2]. The algorithm has the 1-versus-1-versus-rest structure during the decomposition by using the mixture of the formulations of SV classification and regression. In this article, we propose a new algorithm called ν-K-SVCR which is based on ν-SV classification and ν-SV regression.

Like K-SVCR, the new algorithm also requires $K(K-1)/2$ binary classifiers, and each makes the fusion of 1-a-1 and 1-a-a. However, in K-SVCR, the accuracy parameter δ is chosen a priori (cf. (7)–(8) in [2]), while ν-K-SVCR automatically minimizes the accuracy parameter ε (cf. (13)–(18)). In addition, the parameters ν_1 and ν_2 in ν-K-SVCR allow us to control the numbers of support vectors and margin errors effectively, which is helpful in improving the accuracy of each classifier. We give some theoretical results concerning the meaning of the parameters and show the robustness of classifiers. Also, we show that ν-K-SVCR will result in the same classifiers as those of K-SVCR by choosing the parameters appropriately. Finally, we conduct numerical experiments on several artificial data sets and benchmark data sets to demonstrate the theoretical results and the good performance of ν-K-SVCR.

The rest of this article is organized as follows. We first give a brief account of ν-SVM in Section 2. In Section 3, we present ν-K-SVCR algorithm and then show two theoretical results on ν-K-SVCR concerning the meaning of parameters ν_1 and ν_2 and the outlier resistance property of classifiers. In addition, we discuss the connection to K-SVCR. Section 4 gives numerical results to verify the theoretical results and show the performance of ν-K-SVCR. Section 5 concludes the article.

2 ν-SVM

Support vector machines consist of a new class of learning algorithms that are motivated by statistical theory [21]. To describe the basic idea of SVMs, we consider the binary classification problem. Let $\{(\mathbf{x}_i, \mathrm{y}_i),\ i = 1, \cdots, l\}$ be a training example set with the ith example $\mathbf{x}_i \in \mathcal{X} \subseteq R^N$ belonging to one of the two classes labelled by $\mathrm{y}_i \in \{+1, -1\}$. SVMs for classification are used to construct a separating hyperplane $f(\mathbf{x})$ in a high-dimensional feature space $\mathcal{F}$:

$$f(\mathbf{x}) = \mathrm{sgn}((\mathbf{w} \cdot \phi(\mathbf{x})) + b), \tag{2}$$

where $\phi : \mathcal{X} \to \mathcal{F}$ is a nonlinear mapping transforming the examples in the input space into the feature space. This separating hyperplane corresponds to a linear separator in the feature space $\mathcal{F}$ which is equipped with the inner product defined by

$$k(\mathbf{x}_i, \mathbf{x}_j) = (\phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j)), \tag{3}$$

where k is a function called a kernel. When k satisfies the Mercer's theorem, we call it a Mercer kernel [15], and its typical choices include polynomial kernels

$k(\mathbf{x},\mathbf{y}) = (\mathbf{x}\cdot\mathbf{y})^d$ and Gaussian kernels $k(\mathbf{x},\mathbf{y}) = \exp(-\|\mathbf{x}-\mathbf{y}\|^2/(2\sigma^2))$, where $d \in N, \sigma > 0$. The hypothesis space with kernel k is a reproducing kernel Hilbert space (RKHS) $\mathcal{H}$ of functions defined over the domain $\mathcal{X} \subset R^N$ with k being the reproducing kernel, and $\mathcal{H}$ is a closed and bounded set [8]. Hence, it has the finite covering numbers. The kernels used in this article are Mercer kernels.

An interesting type of SVM is ν-SVM developed in [20]. One of its main features is that it has an adjustable regularization parameter ν which has the following significance: ν is both an upper bound on the fraction of errors[1] and a lower bound on the fraction of support vectors. Additionally, when the example size goes to infinity, both fractions converge almost surely to ν under general assumptions on the learning problems and the kernels used [20]. A quadratic programming (QP) formulation of ν-SV classification is given as follows: For $\nu \in (0,1]$ and $C > 0$ chosen a priori,

$$\min \ \tau(\mathbf{w}, b, \xi, \rho) := \frac{1}{2}\|\mathbf{w}\|^2 + C(-\nu\rho + \frac{1}{l}\sum_{i=1}^{l}\xi_i) \tag{4}$$

$$\text{s.t.} \ \ \mathrm{y}_i((\mathbf{w}\cdot\phi(\mathbf{x}_i)) + b) \geq \rho - \xi_i, \tag{5}$$

$$\xi_i \geq 0, \ i = 1, \cdots, l, \tag{6}$$

$$\rho \geq 0. \tag{7}$$

As to ν-SV regression, the estimate function is constructed by using Vapnik's ε-insensitive loss function

$$|\mathrm{y} - f(\mathbf{x})|_\varepsilon = \max\{0, |\mathrm{y} - f(\mathbf{x})| - \varepsilon\}, \tag{8}$$

where $\mathrm{y} \in R$ is a target value. The associated QP problem to be solved is written as

$$\min \ \tau(\mathbf{w}, b, \varphi, \tilde{\varphi}, \varepsilon) := \frac{1}{2}\|\mathbf{w}\|^2 + D(\nu\varepsilon + \frac{1}{l}\sum_{i=1}^{l}(\varphi_i + \tilde{\varphi}_i)) \tag{9}$$

$$\text{s.t.} \ \ -\varepsilon - \tilde{\varphi}_i \leq ((\mathbf{w}\cdot\phi(\mathbf{x}_i)) + b) - \mathrm{y}_i \leq \varepsilon + \varphi_i, \tag{10}$$

$$\varphi_i, \tilde{\varphi}_i \geq 0, \ i = 1, \cdots, l, \tag{11}$$

where $\nu \in (0,1]$ and $D > 0$ are parameters chosen a priori.

[1] In ν-SV regression, errors refer to the training examples lying outside the ε-tube. In ν-SV classification, errors refer to margin errors including examples lying within the margin.

3 THE ν-K-SVCR LEARNING MACHINE

3.1 *The Formulation of ν-K-SVCR*

Let the training set $\mathcal{T}$ be given by (1). For an arbitrary pair $(\Theta_j, \Theta_k) \in \mathcal{Y} \times \mathcal{Y}$ of classes, we wish to construct a decision function $f(\mathbf{x})$ based on a hyperplane similar to (2) which separates the two classes Θ_j and Θ_k as well as the remaining classes. Without loss of generality, let examples $\mathbf{x}_i,\ i = 1, \cdots, l_1$, and $\mathbf{x}_i,\ i = l_1+1, \cdots, l_1+l_2$ belong to the two classes Θ_j and Θ_k which will be labelled $+1$ and -1, respectively, and the remaining examples belong to the other classes which will be labelled 0. Specifically, we wish to find a function $f(\mathbf{x})$ such that

$$f(\mathbf{x}_i) = \begin{cases} +1, i = 1, \cdots, l_1, \\ -1, i = l_1 + 1, \cdots, l_1 + l_2, \\ 0, \quad i = l_1 + l_2 + 1, \cdots, l. \end{cases} \tag{12}$$

In the following, we denote $l_{12} = l_1 + l_2$ and $l_3 = l - l_{12}$.

For $\nu_1, \nu_2 \in (0, 1]$ and $C, D > 0$ chosen a priori, combining ν-SV classification and ν-SV regression, the ν-K-SVCR method solves the following optimization problem:

$$\begin{aligned} \min\ & \tau(\mathbf{w}, b, \xi, \varphi, \tilde{\varphi}, \rho, \varepsilon) \\ & := \frac{1}{2}\|\mathbf{w}\|^2 + C(\frac{1}{l_{12}}\sum_{i=1}^{l_{12}} \xi_i - \nu_1\rho) + D(\frac{1}{l_3}\sum_{i=l_{12}+1}^{l} (\varphi_i + \tilde{\varphi}_i) + \nu_2\varepsilon) && (13) \\ \text{s.t. } & \mathrm{y}_i \cdot ((\mathbf{w} \cdot \phi(\mathbf{x}_i)) + b) \geq \rho - \xi_i,\ i = 1, \cdots, l_{12}, && (14) \\ & (\mathbf{w} \cdot \phi(\mathbf{x}_i)) + b \leq \varepsilon + \varphi_i,\ i = l_{12} + 1, \cdots, l, && (15) \\ & (\mathbf{w} \cdot \phi(\mathbf{x}_i)) + b \geq -\varepsilon - \tilde{\varphi}_i,\ i = l_{12} + 1, \cdots, l, && (16) \\ & \xi_i, \varphi_i, \tilde{\varphi}_i, \varepsilon \geq 0, && (17) \\ & \rho \geq \varepsilon. && (18) \end{aligned}$$

The ν-K-SVCR can be considered to include the ν-SV classification with $\mathrm{y}_i = \pm 1$ (cf.(14)) and the ν-SV regression with 0 being the only target value (cf.(15) and (16)).

Introducing the dual variables $\alpha_i \geq 0,\ i = 1, \cdots, l_{12},\ \beta_i,\ \tilde{\beta}_i \geq 0,\ i = l_{12}+1, \cdots, l,\ \mu_i \geq 0,\ i = 1, \cdots, l_{12},\ \eta_i,\ \tilde{\eta}_i \geq 0,\ i = l_{12}+1, \cdots, l,\ \zeta \geq 0,\ \varsigma \geq 0$, we obtain the following formulation of Karush-Kuhn-Tucker (KKT) conditions for problem (13)–(18):

$$\mathbf{w} = \sum_{i=1}^{l_{12}} \alpha_i \mathrm{y}_i \phi(\mathbf{x}_i) - \sum_{i=l_{12}+1}^{l} (\beta_i - \tilde{\beta}_i)\phi(\mathbf{x}_i), \tag{19}$$

$$\sum_{i=1}^{l_{12}} \alpha_i \mathrm{y}_i - \sum_{i=l_{12}+1}^{l} (\beta_i - \tilde{\beta}_i) = 0, \tag{20}$$

$$\frac{C}{l_{12}} - \mu_i - \alpha_i = 0,\ i = 1, \cdots, l_{12}, \tag{21}$$

$$\frac{D}{l_3} - \beta_i - \eta_i = 0,\ i = l_{12}+1, \cdots, l, \tag{22}$$

$$\frac{D}{l_3} - \tilde{\beta}_i - \tilde{\eta}_i = 0,\ i = l_{12}+1, \cdots, l, \tag{23}$$

$$\sum_{i=1}^{l_{12}} \alpha_i - C\nu_1 - \varsigma = 0, \tag{24}$$

$$\sum_{i=l_{12}+1}^{l} (\beta_i + \tilde{\beta}_i) - D\nu_2 + \zeta - \varsigma = 0, \tag{25}$$

$$\alpha_i \left(\mathrm{y}_i \cdot ((\mathbf{w} \cdot \phi(\mathbf{x}_i)) + b) - \rho + \xi_i\right) = 0,\ i = 1, \cdots, l_{12}, \tag{26}$$

$$\beta_i \left((\mathbf{w} \cdot \phi(\mathbf{x}_i)) + b - \varepsilon - \varphi_i\right) = 0,\ i = l_{12}+1, \cdots, l, \tag{27}$$

$$\tilde{\beta}_i \left((\mathbf{w} \cdot \phi(\mathbf{x}_i)) + b + \varepsilon + \tilde{\varphi}_i\right) = 0,\ i = l_{12}+1, \cdots, l, \tag{28}$$

$$\mu_i \xi_i = 0,\ i = 1, \cdots, l_{12}, \tag{29}$$

$$\eta_i \varphi_i = 0,\ i = l_{12}+1, \cdots, l, \tag{30}$$

$$\tilde{\eta}_i \tilde{\varphi}_i = 0,\ i = l_{12}+1, \cdots, l, \tag{31}$$

$$\zeta\varepsilon = 0, \tag{32}$$

$$\varsigma(\rho - \varepsilon) = 0. \tag{33}$$

By (19)–(33), the dual of problem (13)–(18) can be expressed as follows: For $\nu_1, \nu_2 \in (0, 1]$ chosen a priori,

$$\begin{aligned} &\min\ \ W(\mathbf{r}) := \frac{1}{2}\mathbf{r}^T\mathbf{H}\mathbf{r} \\ &\ \text{s.t.}\ \ 0 \le r_i \mathrm{y}_i \le \frac{C}{l_{12}},\ i = 1, \cdots, l_{12}, \end{aligned}$$

$$
\begin{aligned}
&0 \leq r_i \leq \frac{D}{l_3},\ i = l_{12}+1, \cdots, l+l_3,\\
&\sum_{i=1}^{l_{12}} r_i = \sum_{i=l_{12}+1}^{l} r_i - \sum_{i=l+1}^{l+l_3} r_i,\\
&\sum_{i=1}^{l_{12}} r_i \mathrm{y}_i \geq C\nu_1,\\
&\sum_{i=l_{12}+1}^{l+l_3} r_i \leq D\nu_2,
\end{aligned}
$$

where

$$
\begin{aligned}
&\mathbf{r} := (\alpha_1 \mathrm{y}_1, \cdots, \alpha_{l_{12}} \mathrm{y}_{l_{12}}, \beta_{l_{12}+1}, \cdots, \beta_l, \tilde{\beta}_{l_{12}+1}, \cdots, \tilde{\beta}_l)^T \in R^{l_{12}+l_3+l_3},\\
&\mathbf{H} = \mathbf{H}^T := \begin{pmatrix} (k(\mathbf{x}_i,\mathbf{x}_j)) & -(k(\mathbf{x}_i,\mathbf{x}_j)) & (k(\mathbf{x}_i,\mathbf{x}_j)) \\ -(k(\mathbf{x}_i,\mathbf{x}_j)) & (k(\mathbf{x}_i,\mathbf{x}_j)) & -(k(\mathbf{x}_i,\mathbf{x}_j)) \\ (k(\mathbf{x}_i,\mathbf{x}_j)) & -(k(\mathbf{x}_i,\mathbf{x}_j)) & (k(\mathbf{x}_i,\mathbf{x}_j)) \end{pmatrix} \in R^{(l_{12}+l_3+l_3)\times(l_{12}+l_3+l_3)},
\end{aligned}
$$

with

$$\mathbf{x}_i = \mathbf{x}_{i-l_3},\ i = l+1, \cdots, l+l_3.$$

Denote

$$
v_i = \begin{cases} \alpha_i \mathrm{y}_i,\ i = 1, \cdots, l_{12}, \\ \tilde{\beta}_i - \beta_i,\ i = l_{12}+1, \cdots, l. \end{cases} \tag{34}
$$

Then, by (19), $\mathbf{w}$ is written as

$$
\mathbf{w} = \sum_{i=1}^{l} v_i \phi(\mathbf{x}_i), \tag{35}
$$

and the hyperplane decision function is given by

$$
f(\mathbf{x}) = \begin{cases} +1, \text{ if } \displaystyle\sum_{v_i \in SV} v_i k(\mathbf{x}_i, \mathbf{x}) + b \geq \varepsilon, \\ -1, \text{ if } \displaystyle\sum_{v_i \in SV} v_i k(\mathbf{x}_i, \mathbf{x}) + b \leq -\varepsilon, \\ 0, \quad \text{otherwise,} \end{cases} \tag{36}
$$

where SV $= \{v_i | v_i \neq 0\}$, or alternatively

$$f(\mathbf{x}) = \mathrm{sgn}(g(\mathbf{x}) \cdot |g(\mathbf{x})|_\varepsilon), \tag{37}$$

where

$$g(\mathbf{x}) = \sum_{v_i \in SV} v_i k(\mathbf{x}_i, \mathbf{x}) + b \tag{38}$$

and $|\cdot|_\varepsilon$ is defined by (8). In (35), v_i will be nonzero only if the corresponding constraint (14) or (15) or (16) is satisfied as an equality; the examples corresponding to those v_i's are called *support vectors (SVs).*

To compute b and ρ, we consider two sets $S_\pm$. S_+ contains the SVs $\mathbf{x}_i$ with $\mathrm{y}_i = 1$ and $0 < v_i < \frac{C}{l_{12}}$, $i = 1, \cdots, l_{12}$, and S_- contains the SVs $\mathbf{x}_i$ with $\mathrm{y}_i = -1$ and $-\frac{C}{l_{12}} < v_i < 0$, $i = 1, \cdots, l_{12}$. For these examples in $S_\pm$, we get $\mu_i > 0$ by (21) and then $\xi_i = 0$ by (29). Hence (14) becomes equality with $\xi_i = 0$, and we get

$$\sum_{\mathbf{x} \in S_+} (\mathbf{w} \cdot \phi(\mathbf{x})) + |S_+| b - |S_+| \rho = 0,$$

$$-\sum_{\mathbf{x} \in S_-} (\mathbf{w} \cdot \phi(\mathbf{x})) - |S_-| b - |S_-| \rho = 0,$$

where $|S_+|$ and $|S_-|$ denote the number of examples in S_+ and S_-, respectively. Solving these equations for b and ρ, and using (35) and (3), we obtain

$$b = -\frac{1}{2}[\frac{1}{|S_+|} \sum_{\mathbf{x} \in S_+} \sum_{v_i \in SV} v_i k(\mathbf{x}_i, \mathbf{x}) + \frac{1}{|S_-|} \sum_{\mathbf{x} \in S_-} \sum_{v_i \in SV} v_i k(\mathbf{x}_i, \mathbf{x})], \tag{39}$$

$$\rho = \frac{1}{2}[\frac{1}{|S_+|} \sum_{\mathbf{x} \in S_+} \sum_{v_i \in SV} v_i k(\mathbf{x}_i, \mathbf{x}) - \frac{1}{|S_-|} \sum_{\mathbf{x} \in S_-} \sum_{v_i \in SV} v_i k(\mathbf{x}_i, \mathbf{x})]. \tag{40}$$

The value of ε can be calculated from (15) or (16) on the support vectors.

3.2 *The Properties of ν-K-SVCR*

In this subsection, we first show that the parameter ν in ν-K-SVCR has a similar significance to that in ν-SVM [20]. Note that, since the ν-SVM in [20] only aims at classifying examples into two classes, a margin error in [20] refers

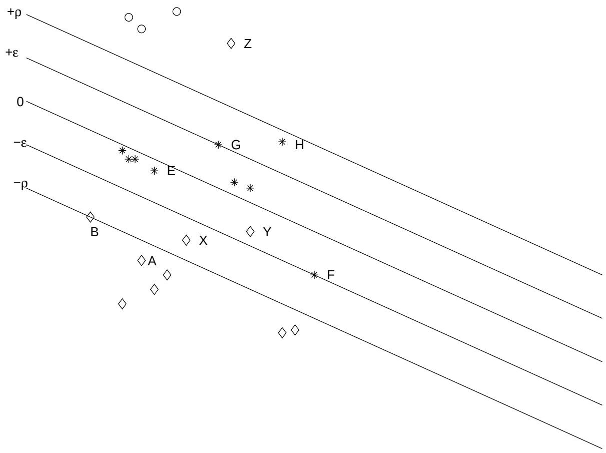

Figure 1. Working zones for a ν-K-SVCR in the feature space on a three-class problem.

to an example that is either misclassified or lying within the ρ-tube. Since our aim is to classify examples into multiple classes, we will modify the definition of margin errors. Namely, throughout this paper, the *margin errors* refer to two kinds of training examples: For the examples in the two classes to be separated, margin errors refer to the examples $\mathbf{x}_i$ with $\xi_i > 0$, that is, the examples that are either misclassified or lying within the ρ-tube; for the examples that are not in either of the two classes, margin errors refer to the examples lying outside the ε-tube. Specifically, the margin errors are the training examples that belong to the set

$$\{i : \mathrm{y}_i g(\mathbf{x}_i) < \rho, i = 1, \cdots, l_{12}\} \cup \{i : |g(\mathbf{x}_i)| > \varepsilon, i = l_{12} + 1, \cdots, l\}, \qquad (41)$$

where $g(\cdot)$ is defined by (38).

Figure 1 depicts the working zones for a ν-K-SVCR in the feature space on a three-class problem: The '$\diamond$', '$\circ$' and '$*$' represent the classes labelled by -1, $+1$ and 0 respectively.

Examples with output -1: A, well-classified example ($v_i = 0$); B, support vector ($v_i \in (-\frac{C}{l_{12}}, 0)$); X, margin error with label -1 ($v_i = -\frac{C}{l_{12}}$) (X is regarded as a margin error since we have $\mathrm{y}_i g(\mathbf{x}_i) \geq \rho$ for this example, even if it is labelled as -1, cf. (41)); Y, margin error with label 0 ($v_i = -\frac{C}{l_{12}}$); Z, margin error with label $+1$ ($v_i = \frac{C}{l_{12}}$).

Examples with output $+1$: Similar to the above.

Examples with output 0: E, well-classified example ($v_i = 0$); F, support vector ($v_i \in (0, \frac{D}{l_3})$); G, support vector ($v_i \in (-\frac{D}{l_3}, 0)$); H, margin error ($v_i = -\frac{D}{l_3}$).

The fraction of margin errors is defined as the number of margin errors divided by the total number of examples, i.e.,

$$\mathrm{FME} := \frac{1}{l}\left\{|\{i : \mathrm{y}_i g(\mathbf{x}_i) < \rho,\ i = 1, \cdots, l_{12}\}| + |\{i : |g(\mathbf{x}_i)| > \varepsilon,\ i = l_{12}+1, \cdots, l\}|\right\}, \tag{42}$$

where $g(\cdot)$ is defined by (38).

THEOREM 3.1 *Suppose ν-K-SVCR is applied to some data set, and the resulting ε and ρ satisfy $\rho > \varepsilon > 0$. Then the following statements hold:*

(i). $\frac{\nu_1 l_{12} + \nu_2 l_3}{l}$ is an upper bound on the fraction of margin errors.

(ii). $\frac{\nu_1 l_{12} + \nu_2 l_3}{l}$ is a lower bound on the fraction of SVs.

(iii). Suppose the data are i.i.d. from a distribution $P(\mathbf{x}, \theta) = P(\mathbf{x})P(\theta|\mathbf{x})$ such that none of $P(\mathbf{x}, \theta = \Theta_1), \cdots, P(\mathbf{x}, \theta = \Theta_K)$ contain any discrete component. Suppose that the kernel is analytic and non-constant. With probability 1, asymptotically, $\frac{\nu_1 l_{12} + \nu_2 l_3}{l}$ equals both the fraction of SVs and the fraction of margin errors.

Proof (i). Suppose the numbers of margin errors in examples $\{(\mathbf{x}_i, \theta_i)\}_{i=1}^{l_{12}}$ and $\{(\mathbf{x}_i, \theta_i)\}_{i=l_{12}+1}^{l}$ are m_1 and m_2, respectively. If an example belonging to $\{(\mathbf{x}_i, \theta_i)\}_{i=1}^{l_{12}}$ is a margin error, then the corresponding slack variable satisfies $\xi_i > 0$, which along with (29) implies $\mu_i = 0$. Hence by (21), the examples with $\xi_i > 0$ satisfy $\alpha_i = \frac{C}{l_{12}}$. On the other hand, since $\rho > \varepsilon$, (33) yields $\varsigma = 0$. Hence (24) becomes

$$\sum_{i=1}^{l_{12}} \alpha_i = C\nu_1. \tag{43}$$

Therefore, at most $\nu_1 l_{12}$ examples in $\{(\mathbf{x}_i, \theta_i)\}_{i=1}^{l_{12}}$ can have $\alpha_i = \frac{C}{l_{12}}$. Thus we obtain

$$m_1 \leq \nu_1 l_{12}. \tag{44}$$

If an example belonging to $\{(\mathbf{x}_i, \theta_i)\}_{i=l_{12}+1}^{l}$ is a margin error, then the corresponding dual variables satisfy $\varphi_i > 0$ or $\tilde{\varphi}_i > 0$. By (30) or (31), we have $\eta_i = 0$ or $\tilde{\eta}_i = 0$ correspondingly. By (22) and (23), we get $\beta_i = \frac{D}{l_3}$ or $\tilde{\beta}_i = \frac{D}{l_3}$. On the other hand, $\rho > \varepsilon > 0$ implies $\zeta = 0$ and $\varsigma = 0$. So (25) becomes

$$\sum_{i=l_{12}+1}^{l} (\beta_i + \tilde{\beta}_i) = D\nu_2. \tag{45}$$

Since $\beta_i \tilde{\beta}_i = 0$, (45) implies that at most $\nu_2 l_3$ examples in $\{(\mathbf{x}_i, \theta_i)\}_{i=l_{12}+1}^{l}$ can have $\beta_i = \frac{D}{l_3}$ or $\tilde{\beta}_i = \frac{D}{l_3}$. Thus we obtain

$$m_2 \leq \nu_2 l_3. \tag{46}$$

Combining (44) and (46) shows that $\frac{\nu_1 l_{12} + \nu_2 l_3}{l}$ is an upper bound on the fraction of margin errors.

(ii). If an example belonging to $\{(\mathbf{x}_i, \theta_i)\}_{i=1}^{l_{12}}$ is a support vector, it can contribute at most $\frac{C}{l_{12}}$ to the left-hand side of (43). Hence there must be at least $\nu_1 l_{12}$ SVs. Similarly, for the examples in $\{(\mathbf{x}_i, \theta_i)\}_{i=l_{12}+1}^{l}$, by (45) and noting $\beta_i \tilde{\beta}_i = 0$, we can deduce that there must be at least $\nu_2 l_3$ SVs. Therefore, $\frac{\nu_1 l_{12} + \nu_2 l_3}{l}$ is an lower bound on the fraction of SVs.

(iii). We prove it by showing that, asymptotically, the probability of examples on the edges of ρ-tube or ε-tube (that is, the examples that satisfy $\mathrm{y}_i g(\mathbf{x}_i) = \rho, \ i = 1, \cdots, l_{12}$, or $g(\mathbf{x}_i) = \pm\varepsilon, \ i = l_{12} + 1, \cdots, l$) vanishes. It follows from the condition on $P(\mathbf{x}, \theta)$ that, apart from some set of measure zero, the K class distributions are absolutely continuous. Since the kernel is analytic and non-constant, it cannot be constant on any open set. Hence all the functions g constituting the argument of the sign in the decision function (cf. (37)) cannot be constant on any open set. Therefore, the distribution over $\mathbf{x}$ can be transformed into the distributions such that for all $t \in R$, $\lim_{\gamma \to 0} P(|g(\mathbf{x}) + t| < \gamma) = 0$. On the other hand, since the class of these functions has well-behaved covering numbers, we get uniform convergence, i.e., for all $\gamma > 0$ and $t \in R$,

$$\sup_g |P(|g(\mathbf{x}) + t| < \gamma) - \hat{P}_l(|g(\mathbf{x}) + t| < \gamma)| \xrightarrow{P} 0, \ l \to \infty,$$

where $\hat{P}_l$ is the sample-based estimate of P with l being the sample size (that is, the proportion of examples that satisfy $|g(\mathbf{x}) + t| < \gamma$). Then for any $\sigma > 0$ and any $t \in R$, we have

$$\lim_{\gamma \to 0} \lim_{l \to \infty} P(\sup_g \hat{P}_l(|g(\mathbf{x}) + t| < \gamma) > \sigma) = 0,$$

and hence

$$\sup_g \hat{P}_l(|g(\mathbf{x}) + t| = 0) \xrightarrow{P} 0, l \to \infty.$$

Setting $t = \pm\rho$ or $t = \pm\varepsilon$ shows that almost surely the fraction of examples exactly on the edges of ρ-tube or ε-tube tends to zero. Since SVs include the examples on the edges of ρ-tube or ε-tube and margin errors, the fraction of

SVs equals that of margin errors. It then follows from (i) and (ii) that both fractions converge almost surely to $\frac{\nu_1 l_{12}+\nu_2 l_3}{l}$. □

Theorem 3.1 gives a theoretical bound on the fraction of margin errors. In fact, we will show in Section 4 by numerical experiments that this theoretical bound is practically useful in estimating the number of margin errors.

For each classifier of ν-K-SVCR, the following theorem shows the 'outlier' resistance property. We use shorthand $\mathbf{z}_i$ for $\phi(\mathbf{x}_i)$ below.

THEOREM 3.2 *Suppose* $\mathbf{w}$ *can be expressed in terms of the SVs that are not margin errors, that is,*

$$\mathbf{w} = \sum_{i=1}^{l} \lambda_i \mathbf{z}_i, \tag{47}$$

where $\lambda_i = 0$ *for all* $i \notin \mathcal{U} := \{i \in \{1, \cdots, l_{12}\} : -\frac{C}{l_{12}} < v_i < 0$ *or* $0 < v_i < \frac{C}{l_{12}}\} \bigcup \{i \in \{l_{12}+1, \cdots, l\} : 0 < v_i < \frac{D}{l_3}$ *or* $-\frac{D}{l_3} < v_i < 0\}$. *Then a sufficiently small perturbation of any margin error* $\mathbf{z}_m$ *along the direction* $\mathbf{w}$ *does not change the hyperplane.*

Proof We first consider the case $m \in \{1, \cdots, l_{12}\}$. Since the slack variable corresponding to $\mathbf{z}_m$ satisfies $\xi_m > 0$, it follows from (29) that $\mu_m = 0$. Then by (21) and (34), we get $v_m = \frac{C}{l_{12}}$ or $-\frac{C}{l_{12}}$. Let $\mathbf{z}'_m := \mathbf{z}_m + \vartheta \mathbf{w}$, where $|\vartheta|$ is sufficiently small. Then the slack variable corresponding to $\mathbf{z}'_m$ will still satisfy $\xi'_m > 0$. Hence we have $v'_m = v_m$. Replacing ξ_m by ξ'_m and keeping all other primal variables unchanged, we obtain an updated vector of primal variables that is still feasible.

In order to keep $\mathbf{w}$ unchanged, from the expression (35) of $\mathbf{w}$, we need to let v'_i, $i \neq m$, satisfy

$$\sum_{i=1}^{l} v_i \mathbf{z}_i = \sum_{i \neq m} v'_i \mathbf{z}_i + v_m \mathbf{z}'_m. \tag{48}$$

Substituting $\mathbf{z}'_m = \mathbf{z}_m + \vartheta \mathbf{w}$ and (47) into (48), and noting that $m \notin \mathcal{U}$, we get a sufficient condition for (48) to hold is that for all $i \neq m$

$$v'_i = v_i - \vartheta \lambda_i v_m. \tag{49}$$

We next show that b is also kept unchanged. Recall that by assumption $\lambda_i = 0$ for any $i \notin \mathcal{U}$, and so $\lambda_m = 0$. Hence by (34) and $v'_m = v_m$, (49) yields

$$\alpha'_i = \alpha_i - \vartheta \lambda_i \mathrm{y}_i \alpha_m \mathrm{y}_m, \qquad i = 1, \cdots, l_{12}.$$

If $\alpha_i \in \left(0, \frac{C}{l_{12}}\right)$, then α'_i will be in $\left(0, \frac{C}{l_{12}}\right)$, provided $|\vartheta|$ is sufficiently small. If $\alpha_i = \frac{C}{l_{12}}$, then by assumption we have $\lambda_i = 0$, and hence α'_i will equal $\frac{C}{l_{12}}$. Thus a small perturbation $\mathbf{z}_m$ along the direction $\mathbf{w}$ does not change the status of examples $(\mathbf{x}_i, \theta_i),\ i = 1, \cdots, l_{12}$. By (39), b does not change. Thus $(\mathbf{w}, b)$ remains to be the hyperplane parameters.

In the case $m \in \{l_{12}+1, \cdots, l\}$, the result can be obtained in a similar way. □.

The new learning machine makes the fusion of the standard structures, one-against-one and one-against-all, employed in the decomposition scheme of a multi-class classification procedure. In brief, a ν-K-SVCR classifier is trained to focus on the separation between two classes, as one-against-one does. At the same time, it also gives useful information about the other classes that are labelled 0, as one-against-all does. In addition, for each classifier, we can use the parameters ν_1 and ν_2 to control the number of margin errors, which is helpful in improving the accuracy of each classifier. Theorem 3.2 indicates that every classifier is robust.

3.3 *The Connection to K-SVCR*

Angulo et al. [2] propose K-SVCR method for the multi-class classification. The following theorem shows that ν-K-SVCR formulation will result in the same classifier as that of K-SVCR by selecting the parameters properly.

THEOREM 3.3 *If $\mathbf{w}^\nu$, b^ν, ξ_i^ν, φ_i^ν, $\tilde{\varphi}_i^\nu$, ε^ν and $\rho^\nu > 0$ constitute an optimal solution to a ν-K-SVCR with given ν_1, ν_2, then $\mathbf{w}^C = \frac{\mathbf{w}^\nu}{\rho^\nu}$, $b^C = \frac{b^\nu}{\rho^\nu}$, $\xi_i^C = \frac{\xi_i^\nu}{\rho^\nu}$, $\varphi_i^C = \frac{\varphi_i^\nu}{\rho^\nu}$, $\tilde{\varphi}_i^C = \frac{\tilde{\varphi}_i^\nu}{\rho^\nu}$ constitute an optimal solution to the corresponding K-SVCR, with $C^C = \frac{C^\nu}{l_{12}\rho^\nu}$, $D^C = \frac{D^\nu}{l_3\rho^\nu}$ and $\delta^C = \frac{\varepsilon^\nu}{\rho^\nu}$ being a priori chosen parameters in K-SVCR.*

Proof Consider the primal formulation of ν-K-SVCR. Let an optimal solution be given by $\mathbf{w}^\nu$, b^ν, ξ_i^ν, φ_i^ν, $\tilde{\varphi}_i^\nu$, ε^ν and ρ^ν. Substituting $\mathbf{w}' = \frac{\mathbf{w}}{\rho^\nu}$, $b' = \frac{b}{\rho^\nu}$, $\xi'_i = \frac{\xi_i}{\rho^\nu}$, $\varphi'_i = \frac{\varphi_i}{\rho^\nu}$ and $\tilde{\varphi}'_i = \frac{\tilde{\varphi}_i}{\rho^\nu}$ in the ν-K-SVCR (13)–(18), we have the optimization problem

$$\min_{\{\mathbf{w}', b', \xi', \varphi', \tilde{\varphi}'\}} \frac{1}{2}\|\mathbf{w}'\|^2 + C^C \sum_{i=1}^{l} \xi'_i + D^C (\sum_{i=1}^{l} (\varphi'_i + \tilde{\varphi}'_i)) \tag{50}$$

$$\text{s.t.} \quad \mathrm{y}_i((\mathbf{w}' \cdot \phi(\mathbf{x}_i)) + b') \geq 1 - \xi'_i,\ i = 1, \cdots, l_{12}, \tag{51}$$

$$-\delta^C - \tilde{\varphi}'_i \leq (\mathbf{w}' \cdot \phi(\mathbf{x}_i)) + b' \leq \delta^C + \tilde{\varphi}'_i,\ i = l_{12}+1, \cdots, l, \tag{52}$$

$$\xi'_i, \varphi'_i, \tilde{\varphi}'_i \geq 0. \tag{53}$$

Note that this has the same form as K-SVCR. It is not difficult to see that $\mathbf{w}^C$, b^C, ξ_i^C, φ_i^C, $\tilde{\varphi}_i^C$ constitute an optimal solution to problem (50)–(53). □

3.4 *The Combination Method*

After the decomposition scheme, we need a reconstruction scheme to fuse the outputs of all classifiers for each example and assign it to one of the classes. In a K-class problem, for each pair, say Θ_j and Θ_k, we have a classifier $f_{j,k}(\cdot)$ to separate them as well as the other classes (cf.(12)). So we have $K(K-1)/2$ classifiers in total. Hence, for an example $\mathbf{x}_p$, we get $K(K-1)/2$ outputs. We translate the outputs as follows: When $f_{j,k}(\mathbf{x}_p) = +1$, a positive vote is added on Θ_j, and no votes are added on the other classes; when $f_{j,k}(\mathbf{x}_p) = -1$, a positive vote is added on Θ_k, and no votes are added on the other classes; when $f_{j,k}(\mathbf{x}_p) = 0$, a negative vote is added on both Θ_j and Θ_k, and no votes are added on the other classes. After we translate all of the $K(K-1)/2$ outputs, we will get the total votes of each class by adding the positive and negative votes on this class. Finally, $\mathbf{x}_p$ will be assigned to the class that gets the most votes.

4 EXPERIMENTS

In this section, we present two types of experiments demonstrating the performance of ν-K-SVCR: Experiments on artificial data sets which are used to verify the theoretical results, and experiments on benchmark data sets.

4.1 *Experiments on Artificial Data Sets*

In this subsection, several experiments with artificial data in R^2 are carried out using Matlab v6.5 on Intel Pentium IV 3.00GHz PC with 1GB of RAM. The QP problems are solved with the standard Matlab routine. For simplicity, we set $\nu_1 = \nu_2$, which is denoted ν. The main tasks in the experiments are the following:

(1) To investigate the effect of parameter ν on the fraction of margin errors, the fraction of SVs, and the values of ε and ρ;

(2) To observe an asymptotic behavior of the fractions of margin errors and SVs.

The effect of parameter ν

The influence of parameter ν is investigated on the training set $\mathcal{T}$ generated from a Gaussian distribution on R^2. The set $\mathcal{T}$ contains 150 examples in three classes, each of which has 50 examples, as shown in Figure 2.

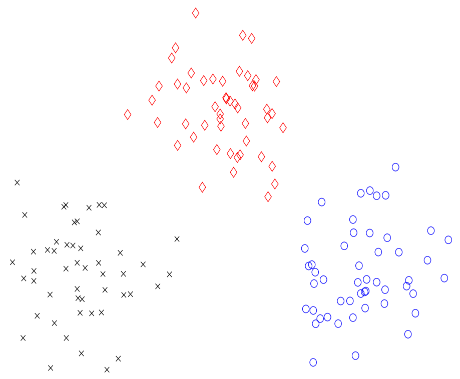

Figure 2. Training set $\mathcal{T}$

Table 1. Fractions of margin errors and SVs and the values of ε and ρ for $\mathcal{T}$.

ν	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8
FME[a]	0	0.0667	0.1667	0.2667	0.3533	0.4600	0.5600	0.6400
FSV[b]	0.3867	0.4067	0.5333	0.6133	0.6867	0.7867	0.8333	0.8933
ε	0.0246	0.0112	0.0085	0.0045	0.0041	0.0009	0.0005	0
ρ	14.2287	29.1982	49.2638	72.9279	100.2379	131.1074	166.9139	213.3048

[a] Fraction of margin errors.

[b] Fraction of support vectors.

In the experiment, a Gaussian kernel with $\sigma = 0.2236$ is employed. We choose $C = 2000$ and $D = 300$. Table 1 shows that ν provides an upper bound of the fraction of margin errors and a lower bound of the fraction of SVs, and that increasing ν allows more margin errors, increases the value of ρ, and decreases the value of ε. It is worth noticing from Table 1 that the assumption $\rho > \varepsilon$ in Theorem 3.1 does hold in practice.

The tendency of the fractions of margin errors and SVs

In this experiment, we show the tendency of the fractions of margin errors and SVs when the number of training examples increases. We choose $\nu = 0.2$. Ten training sets are generated from the same Gaussian distribution. In each training set, there are three classes. The total number of examples in each training set is l, and each class has $l/3$ examples. Hence we have $\frac{\nu_1 l_{12} + \nu_2 l_3}{l} = 0.2$. We use a polynomial kernel with degree 4, and set $C = 100$, $D = 50$. Table 2 shows that both the fraction of margin errors and the fraction of SVs tend to 0.2 from below and above, respectively, when the total number of examples l increases. This confirms the results established in Theorem 3.1.

Table 2. Asymptotic behavior of the fractions of margin errors and SVs.

l	30	60	90	120	150	180	210	240	270	300
FME	0.13	0.13	0.18	0.17	0.18	0.19	0.19	0.19	0.19	0.20
FSV	0.30	0.25	0.23	0.22	0.22	0.23	0.22	0.22	0.21	0.20

Table 3. The percentage of error by 1-a-a, 1-a-1, qp-mc-sv, lp-mc-sv, K-SVCR and ν-K-SVCR.

name	#pts[a]	#att[b]	#class	1-a-a	1-a-1	qp-mc-sv	lp-mc-sv	K-SVCR	ν-K-SVCR
Iris	150	4	3	1.33	1.33	1.33	2.0	[1.93, 3.0]	1.33
Wine	178	13	3	5.6	5.6	3.6	10.8	[2.29, 4.29]	3.3
Glass	214	9	6	35.2	36.4	35.6	37.2	[30.47, 36.35]	[32.38, 36.19]

[a] The number of training data.

[b] The attributes of examples.

4.2 *Experiments on Benchmark Data Sets*

In this subsection, we test ν-K-SVCR on a collection of three benchmark data sets from the UCI machine learning repository [4], 'Iris', 'Wine' and 'Glass'. In problem 'Glass', there is one missing class. Since no test data sets are provided in the three benchmark data sets, we use ten-fold cross validation to evaluate the performance of the algorithms. That is, each data set is split randomly into ten subsets and one of those sets is reserved as a test set; this process is repeated ten times. For 'Iris' and 'Wine', the polynomial kernels with degree 4 and 3 are employed, respectively. For 'Glass', the Gaussian kernel with $\sigma = 0.2236$ is employed. We choose $\nu = 0.01$ in each algorithm. We compare the obtained results with one-against-all (1-a-a), one-against-one (1-a-1), quadratic multi-class SVM (qp-mc-sv) and linear multi-class SVM (lp-mc-sv) proposed in [23], and K-SVCR proposed in [2]. The results are summarized in Table 3. In the table, $[\cdot, \cdot]$ refers to the case where two or more classes get the most votes in a tie. The first and second numbers in the brackets are the percentage of error when examples are assigned to the right and the wrong classes, respectively, among those with the most votes.

It can be observed that the performance of the new algorithm is generally comparable to the other ones. Specifically, for the 'Iris' set, ν-K-SVCR outperforms the lp-mc-sv and K-SVCR methods; and it is comparable to the others. For the 'Wine' set, ν-K-SVCR is comparable to K-SVCR; and it outperforms the others. For the 'Glass' set, ν-K-SVCR shows a similar performance to all the others. Note that the new algorithm is not fundamentally different from the K-SVCR algorithm. Indeed, we have shown that for a certain parameter setting, both algorithms will produce the same results. In practice, however, it may sometimes be desirable to specify a fraction of points that are allowed to become errors.

5 CONCLUSION

We have proposed a new algorithm, ν-K-SVCR, for the multi-class classification based on ν-SV classification and ν-SV regression. By redefining the concept of margin errors, we have clarified the theoretical meaning of the parameters in ν-K-SVCR. We have also shown the robustness of the classifiers and connection to K-SVCR. We have confirmed the established theoretical results and good behavior of the algorithm through experiments on several artificial data and benchmark data sets. Future research subjects include more comprehensive testing of the algorithm and application to real-world problems.

References

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

[2] C. Angulo, X. Parra and A. Català (2003). K-SVCR. A support vector machine for multi-class classification. *Neurocomputing*, **55**, 57–77.

[3] K.P. Bennett (1999). Combining support vector and mathematical programming methods for classification. in: B. Schölkopf, C. J. C. Burges and A. J. Smola (Eds.), *Advances in Kernel Methods: Support Vector Learning*, pp. 307–326. MIT Press, Cambridge, MA.

[4] C. L. Blake and C. J. Merz (1998). UCI repository of machine learning databases. University of California. [www http://www.ics.uci.edu/∼mlearn/MLRepository.html]

[5] L. Bottou, C. Cortes, J. S. Denker, H. Drucker, I. Guyon, L. D. Jackel, Y. LeCun, U. A. Müller, E. Sackinger, P. Simard and V. Vapnik (1994). Comparison of classifier methods: a case study in handwriting digit recognition. in: IAPR (Ed.), *Proceedings of the International Conference on Pattern Recognition*, pp. 77–82. IEEE Computer Society Press.

[6] K. Crammer and Y. Singer (2002). On the algorithmic implementation of multiclass kernel-basd vector machines. *Journal of Machine Learning Research*, **2**, 265–292.

[7] K. Crammer and Y. Singer (2002). On the learnability and design of output codes for multiclass problems. *Machine Learning*, **47**, 201–233.

[8] N. Cristianini and J. Shawe-Taylor (2000). *An Introduction to Support Vector Machines.* Cambridge University Press, Cambridge, UK.

[9] T.G. Dietterich and G. Bakiri (1995). Solving multi-class learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research*, **2**, 263–286.

[10] J. Fürnkranz (2002). Round robin classification. *Journal of Machine Learning Research*, **2**, 721–747.

[11] T. J. Hastie and R. J. Tibshirani (1998). Classification by pairwise coupling. in: M. I. Jordan, M. J. Kearns and S. A. Solla (Eds.), *Advances in Neural Information Processing Systems*, **10**, 507–513. MIT Press, Cambridge, MA.

[12] U. Kreßel (1999). Pairwise classification and support vector machines. in: B. Schölkopf, C. J. C. Burges and A. J. Smola (Eds.), *Advances in Kernel Methods: Support Vector Learning*, pp. 255–268. MIT Press, Cambridge, MA.

[13] C. W. Hsu and C. J. Lin (2002). A comparison of methods for multiclass support vector machines. *IEEE Transactions on Neural Networks*, **13**, 415–425.

[14] Y. Lee, Y. Lin and G. Wahba (2001). Multicategory support vector machines. *Computing Science and Statistics*, **33**, 498–512.

[15] J. Mercer (1909). Functions of positive and negative type and their connection with the theory of integral equations. *Philosophical Transactions of the Royal Society, London*, **A 209**, 415–446.

[16] M. Moreira and E. Mayoraz (1998). Improved pairwise coupling classification with correcting classifiers. *Lecture Notes in Computer Science: Proceedings of the 10th European conference on Machine Learning*, pp. 160–171.

[17] J. Platt (1999). Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. in: A.J. Smola, P. Bartlett, B. Schölkopf and D. Schuurmans (Eds.), *Advances in Large Margin Classifiers*, pp. 61–74. MIT Press, Cambridge, MA.

[18] J. Platt, N. Cristianini and J. Shawe-Taylor (2000). Large margin DAGs for multiclass classifi-

cation. in: S. A. Solla, T. K. Leen and K. -R. Müller (Eds.), *Advances in Neural Information Processing Systems*, **12**, 547–553. MIT Press, Cambridge, MA.

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

[20] B. Schölkopf, A. J. Smola, R. C. Williamson and P. L. Bartlett (2000). New support vector algorithms. *Neural Computation*, **12**, 1207–1245.

[21] V. Vapnik (1995). *The Nature of Statistical Learning Theory.* Springer-Verlag, New York.

[22] V. Vapnik (1998). *Statistical Learning Theory.* Wiley, New York.

[23] J. Weston and C. Watkins (1998). Multi-class support vector machines. CSD-TR-98-04 Royal Holloway, University of London, Egham, UK.