Multi-Classification by Using Tri-Class SVM

CECILIO ANGULO[1,*], FRANCISCO J. RUIZ[1], LUIS GONZÁLEZ[2], and JUAN ANTONIO ORTEGA[3]

[1] *Grup de Recerca en Enginyeria del Coneixement, Universitat Politècnica de Catalunya, Av. Víctor Balaguer s/n. 08800 – Vilanova i la Geltrú, Spain. e-mail: cecilio.angulo@upc.edu*

[2] *Departamento de Economía Aplicada I, Universidad de Sevilla, Avenida Ramón y Cajal, 1. 41018 – Sevilla, Spain*

[3] *Escuela Técnica Superior de Ingenieŕia Informática, Universidad de Sevilla, Avenida Reina Mercedes, s/n. 41012 – Sevilla, Spain*

Abstract. The standard form for dealing with multi-class classification problems when bi-classifiers are used is to consider a two-phase (decomposition, reconstruction) training scheme. The most popular decomposition procedures are pairwise coupling (one versus one, 1-v-1), which considers a learning machine for each Pair of classes, and the one-versus-all scheme (one versus all, 1-v-r), which takes into consideration each class versus the remaining classes. In this article a 1-v-1 tri-class Support Vector Machine (SVM) is presented. The expansion of the architecture of this machine into three categories specifically addresses the decomposition problem of how to prevent the loss of information which occurs in the usual 1-v-1 training procedure. The proposed machine, by means of a third class, allows all the information to be incorporated into the remaining training patterns when a multi-class problem is considered in the form of a 1-v-1 decomposition. Three general structures are presented where each improves some features from the precedent structure. In order to deal with multi-classification problems, it is demonstrated that the final machine proposed allows ordinal regression as a form of decomposition procedure. Examples and experimental results are presented which illustrate the performance of the new tri-class SV machine.

Key words. bi-classifier, multi-classification, ordinal regression, Support Vector Machine

Abbreviations. 1-v-1 – one versus one; all versus all; pairwise coupling; 1-v-r – one versus the rest; one versus all; s.t. – subject to; SV – Support Vector; SVM – Support Vector Machine

1. Introduction

Support Vector Machines (SVMs) are learning machines which implement the structural risk minimization inductive principle to obtain good generalization on a limited number of learning patterns. This theory was originally developed by Vapnik on the basis of a separable binary classification problem with signed outputs ±1 [21].

The SVM presents good theoretical properties and behaviour in problems of binary classification [9]. There are several papers which generalize the original

[*] Corresponding author.

bi-class approach to multi-classification problems [16, 17, 1] through different algorithms, such as 1-v-r SVM or 1-v-1 SVM (see [15] for a comparison of SVM multi-class methods). In this work it is assumed that problems with more than 2 classes will be considered, hence the original bi-class SVM is extended to a more general tri-class SVM approach. The proposed final tri-class machine is presented in a three-stage procedure: first the original idea of a third class is introduced which was developed by Angulo and Català [3, 2]; secondly a more specific machine, as proposed by Angulo and González [5] is presented; finally, the proposed novel tri-class SVM is explained, which implies a huge computational cost reduction with respect to the former proposals, and a meeting point for both classification and ordinal regression techniques.

The rest of the article is organized as follows: in Section 2, the standard SVM classification learning paradigm is briefly presented in order to introduce some notation. Section 3 is devoted to a short introduction about SVMs for multi-classification. In Section 4, the 1-v-1 tri-class SV Machine is presented, and its faster computational counterpart is derived in Section 5. Examples and experimental results are displayed in Section 6 to illustrate its behaviour and strengths. Finally, some conclusions are drawn and future research suggested.

2. Bi-Class SV Machine Learning

The SV Machine is an implementation of a more general regularization principle known as the large margin principle. Let

$$\mathcal{Z} = \{(x_1, y_1), \ldots, (x_n, y_n)\} = \{z_1, \ldots, z_n\} \in (\mathcal{X} \times \mathcal{Y})^n \tag{1}$$

be a training set, where $\mathcal{X}$ is the input space and

$$\mathcal{Y} = \{\theta_1, \theta_2\} = \{-1, +1\} \tag{2}$$

the output space. Let

$$\phi : \mathcal{X} \to \mathcal{F} \subseteq \mathbb{R}^d \tag{3}$$

be a feature mapping, with $\phi = (\phi_1, \ldots, \phi_d)$, for the usual 'kernel trick'. $\mathcal{F}$ is named *feature space*. Let

$$\mathbf{x} \overset{\text{def}}{=} \phi(x) \in \mathcal{F} \tag{4}$$

be the *representation* of $x \in \mathcal{X}$. A binary linear classifier,

$$f_{\mathbf{w}}(x) = \langle \phi(x), \mathbf{w} \rangle + b = \langle \mathbf{x}, \mathbf{w} \rangle + b \tag{5}$$

is sought in the space $\mathcal{F}$, with $f_{\mathbf{w}}: \mathcal{X} \to \mathcal{F} \to \mathbb{R}$, $b \in \mathbb{R}$, and where outputs are obtained by thresholding the classifier, $h_{\mathbf{w}}(x) = \text{sign}(f_{\mathbf{w}}(x))$. According to [12], the classifier **w** with the largest geometrical margin on a given training sample $\mathcal{Z}$ can be written as

$$\mathbf{w}_{\mathrm{SVM}} \overset{\mathrm{def}}{=} \arg\max_{\mathbf{w}\in\mathcal{F}} \frac{1}{\|\mathbf{w}\|} \cdot \min_{z_i\in\mathcal{Z}} y_i \langle \mathbf{x_i}, \mathbf{w}\rangle. \tag{6}$$

One practical method of dealing with the problem is to minimize the norm $\|\mathbf{w}\|$ in (6) with the geometrical margin fixed to unity

$$\begin{aligned} &\min_{\mathbf{w}\in\mathcal{F}} \tfrac{1}{2}\|\mathbf{w}\|^2 \\ &\text{s.t. } y_i \langle \mathbf{x_i}, \mathbf{w}\rangle \geqslant 1 \qquad z_i \in \mathcal{Z}. \end{aligned} \tag{7}$$

The solution can be expressed in the form

$$\mathbf{w}_{\mathrm{SVM}} = \sum_i \alpha_i\, y_i\, \mathbf{x_i}; \qquad f_{\mathbf{w}_{\mathrm{SVM}}}(x) = \sum_i \alpha_i\, y_i\, k(x_i, x), \tag{8}$$

where $k(x, x') = \langle \phi(x), \phi(x')\rangle = \langle \mathbf{x}, \mathbf{x}'\rangle$ is the kernel function, and only a few α_i are not zero; those associated to the so-called *support vectors*.

3. SV Machine for Multi-Classification

Let $\mathcal{Z}$ be a training set. Now, a set of possible labels $\{\theta_1, \dots, \theta_\ell\}$, with $\ell > 2$ will be considered. Subsets $\mathcal{Z}_k \in \mathcal{Z}$, defined as

$$\mathcal{Z}_k = \{z_i = (x_i, y_i) \colon y_i = \theta_k\} \tag{9}$$

generate a partition in $\mathcal{Z}$, and $n_k = \#\mathcal{Z}_k$, hence $n = n_1 + n_2 + \cdots + n_\ell$. If I_k is defined as the set of indexes i where $z_i \in \mathcal{Z}_k$, it follows that,

$$\bigcup_{i\in I_k} \{(\mathbf{x_i}, y_i)\} = \mathcal{Z}_k. \tag{10}$$

A very common decomposition procedure for multi-classification when SVMs are considered is 1-v-1 SVM: a first decomposition phase generates several learning machines in parallel, whereby each machine takes only two classes into consideration. A reconstruction scheme then allows the calculation of the overall output by merging outputs from the decomposition phase. In this approach, $\frac{\ell\cdot(\ell-1)}{2}$ binary classifiers are trained to generate hyperplanes f_{kh}, $1 \leqslant k < h \leqslant \ell$, by separating training vectors $\mathcal{Z}_k$ with label θ_k from training vectors in class θ_h, $\mathcal{Z}_h$. If f_{kh} discriminates without error then $\mathrm{sign}(f_{kh}(x_i)) = 1$, for $z_i \in \mathcal{Z}_k$ and $\mathrm{sign}(f_{kh}(x_i)) = -1$, for $z_i \in \mathcal{Z}_h$. Remaining training vectors $\mathcal{Z} \setminus \{\mathcal{Z}_k \cup \mathcal{Z}_h\}$ are not considered in the optimization problem. Hence, for a new entry x, the numeric output from each machine $f_{kh}(x)$ is interpreted as,

$$\Theta(f_{kh}(x)) = \begin{cases} \theta_k \text{ if } \mathrm{sign}(f_{kh}(x)) = 1 \\ \theta_h \text{ if } \mathrm{sign}(f_{kh}(x)) = -1. \end{cases} \tag{11}$$

In the reconstruction phase, the label distribution generated by the trained machines in the parallel decomposition is considered through a merging scheme.

The 1-v-1 multi-classification approach is usually preferred to the 1-v-r scheme [16] because it takes less training time, despite studies such as [19]. Moreover, according to [15] it would be difficult to say which one gives better accuracy. The main drawback for this approach is that only data from two classes is considered for the training of each machine, therefore output variance is high and any information from the rest of the classes is ignored.

If a hyperplane f_{kh} must classify an input x_i with $i \notin I_k \bigcup I_h$, only output $f_{kh}(x_i) = 0$ will be translated into a correct interpretation. The natural improvement to be analysed is the enforcement of every training input in a different class from θ_k and θ_h to be contained in the separating hyperplane $f_{kh}(x) = 0$.

3.1. THE K-SVCR MACHINE. A FIRST APPROACH

In [2], the first tri-class procedure, the K-SVCR machine, was presented where remaining training vectors are forced to be encapsulated in a δ-tube, $0 \leqslant \delta < 1$, along the separation hyperplane. Parameter δ allows the creation of a slack zone (a 'tube') around the hyperplane where remaining training vectors are covered. The separating hyperplane must solve the optimization problem,

$$\begin{array}{ll} \min\limits_{\mathbf{w} \in \mathcal{F}} & \frac{1}{2} \|\mathbf{w}\|^2 + C_1 \cdot \sum\limits_i \xi_i + C_2 \cdot \sum\limits_j (\varphi_j + \varphi_j^*) \\ \text{s.t.} & \left\{ \begin{array}{rl} y_i \langle \mathbf{w}, \mathbf{x_i} \rangle \geqslant 1 - \xi_i & z_i \in \mathcal{Z}_{1,3} \\ -\delta - \varphi_j^* \leqslant \langle \mathbf{w}, \mathbf{x_j} \rangle \leqslant \delta + \varphi_j & z_j \in \mathcal{Z}_2 \\ \xi_i \geqslant 0 & z_i \in \mathcal{Z}_{1,3} \\ \varphi_j^*, \varphi_j \geqslant 0 & z_j \in \mathcal{Z}_2, \end{array} \right. \end{array} \tag{12}$$

where $\mathcal{Z}_{1,3}$ are the patterns belonging to the classes labelled as $\{-1, +1\}$ and $\mathcal{Z}_2$ are those labelled with 0. The solution has a similar form to (8), where α_i are the multipliers associated to the problem, such that $\sum_i \alpha_i = 0$. For a new entry x, the numeric output from the machine $f_{\mathbf{w}}(x)$ is interpreted as

$$\Theta(f_{\mathbf{w}}(x)) = \left\{ \begin{array}{rl} 1 & \text{if } f_{\mathbf{w}}(x) > \delta \\ -1 & \text{if } f_{\mathbf{w}}(x) < -\delta \\ 0 & \text{if } |f_{\mathbf{w}}(x)| \leqslant \delta. \end{array} \right. \tag{13}$$

This approach has demonstrated good results on standard 'benchmarks' [2], however, for the general case, it is necessary to select many parameters,[1] such as: (i) k, kernel function; (ii) C_1, associated weight for the sum of errors in the two discriminated classes; (iii) C_2, associated weight for the sum of errors in the remaining classes; (iv) δ, insensitivity parameter.

[1] An extended study can be found in [10].

3.2. ROBUST DECOMPOSITION – RECONSTRUCTION PROCEDURE

The K-SVCR machine improves standard algorithms treating 2-class classification problems during the decomposing phase of a general multi-class scheme: by focusing the learning on 2 classes, but using all the disposable information on the patterns. Now, a second theoretical advantage of the 'third-class approach' will be enunciated, the robustness of the reconstruction procedure [6]. To make evident this assertion, a definition must be done.

DEFINITION 1. Let $x \in \mathcal{X}$ be an entry having a known output, θ_m. Let

$$\varepsilon_{\text{rob}}(x, F) = \frac{\# f_m^{\text{err}}}{L_m}$$

be the rate between the number of classifiers concerning class θ_m producing a wrong output, $\# f_m^{\text{err}}$, and the total number of concerned classifiers with class θ_m, L_m, being correct the final multi-class architecture output, $F(x) = \theta_m$. The *robustness parameter*

$$\varepsilon_{\text{rob}}(F) = \arg\min_{x} \varepsilon_{\text{rob}}(x, F) \qquad \forall x \in \mathcal{X}$$

determines that a general decomposition and reconstruction multi-class architecture $\mathbf{A}_1$ is more *robust* than $\mathbf{A}_2$ if

$$\varepsilon_{\text{rob}}^1 = \min_{F \in \mathbf{A}_1} \varepsilon_{\text{rob}}^1(F) > \min_{F \in \mathbf{A}_2} \varepsilon_{\text{rob}}^2(F) = \varepsilon_{\text{rob}}^2, \tag{14}$$

where superscripts refer to the global architecture being considered.

Basically, the robustness parameter specifies, for the worst case, how many classifiers concerned with the class of the entry could be wrong while the multi-class architecture output is still correct.

Now, it can be enunciated the following Proposition [6].

PROPOSITION 2. *If K is the number of classes in consideration, the multi-class architecture based on a three-classes machine, like K-SVCR machine, with a voting reconstruction scheme F has a robustness parameter*

$$\varepsilon_{\text{rob}} = \frac{2(K-2)}{K(K-1)}.$$

In a similar way the following Proposition can be demonstrated [6].

PROPOSITION 3. *A standard multi-class architecture based on 1-v-r 2-class classifiers decomposition and a voting reconstruction scheme has a robustness parameter*

$$\varepsilon_{\text{rob}} = 0.$$

A standard multi-class architecture based on 1-v-1 2-class classifiers decomposition and voting reconstruction scheme has a robustness parameter

$$\varepsilon_{\text{rob}} = 0.$$

A 'pairwise' multi-class architecture [16] based on 1-v-1 2-class classifiers decomposition and 'pairwise' voting reconstruction scheme has a robustness parameter

$$\varepsilon_{\text{rob}} = 0.$$

A DAGSVM architecture [18] has a robustness parameter

$$\varepsilon_{\text{rob}} = 0.$$

4. 1-v-1 Tri-Class SVM. A Second Approach

The number of tuning parameters can be reduced if the margin to be maximized in (7) is that defined between the patterns assigned with output $\{-1, +1\}$, and the entries labelled with 0, which are the remaining patterns. In this case, the width of the 'decision tube' along the decision hyperplane where 0-labelled patterns are allocated is not considered 'a priori' and the δ parameter is eliminated. A classifier with this characteristic must accomplish

$$\mathbf{w}_{SV3} \overset{\text{def}}{=} \arg\max_{\mathbf{w}\in\mathcal{F}} \frac{1}{\|\mathbf{w}\|} \cdot \left\{ \min_{z_i \in \mathcal{Z}_{1,3}} y_i \langle \mathbf{x_i}, \mathbf{w} \rangle - \max_{z_i \in \mathcal{Z}_2} |\langle \mathbf{x_i}, \mathbf{w} \rangle| \right\}. \tag{15}$$

When $\|\mathbf{w}\|$ is minimized while the rest of the product is fixed to the unitary distance, (15) can be translated into the more manageable[2]

$$\begin{array}{ll} \min\limits_{\mathbf{w}\in\mathcal{F}} & \frac{1}{2}\|\mathbf{w}\|^2 \\ \text{s.t.} & y_i \langle \mathbf{x_i}, \mathbf{w} \rangle \geqslant 1 + \left|\langle \mathbf{x_j}, \mathbf{w} \rangle\right| \qquad z_i \in \mathcal{Z}_{1,3}; \quad z_j \in \mathcal{Z}_2. \end{array} \tag{16}$$

This optimization problem is consistent with the standard formulation since if all the 0-labelled training patterns are exactly on the decision hyperplane, (i.e. no incorrect interpretation is possible), or these patterns are not considered in the problem, then the novel machine would be similar to the 1-v-1 SVM machine.

Restrictions can be relaxed to allow some degree of noise on the ±1-labelled training patterns by using 'slack' variables

$$\xi_i = 1 + \max_{z_j \in \mathcal{Z}_2} \left|\langle \mathbf{x_j}, \mathbf{w} \rangle\right| - y_i \langle \mathbf{x_i}, \mathbf{w} \rangle \geqslant 0 \qquad z_i \in \mathcal{Z}_{1,3} \tag{17}$$

[2] Constraints are slightly stricter than (15).

and restrictions in (16) can be manipulated to obtain the optimization problem [5]

$$\begin{array}{ll} \min\limits_{\mathbf{w}\in\mathcal{F}} & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum\limits_i \xi_i \\ \text{s.t.} & \begin{cases} y_i\langle \mathbf{x_i}-\mathbf{x_j}, \mathbf{w}\rangle - 1 + \xi_i \geqslant 0 & z_i \in \mathcal{Z}_{1,3};\ z_j \in \mathcal{Z}_2 \\ \xi_i \geqslant 0 & z_i \in \mathcal{Z}_{1,3}. \end{cases} \end{array} \tag{18}$$

When Lagrange multipliers are applied to the original optimization problem, it is obtained

$$\mathcal{L} = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_i \xi_i + \sum_{ij}\alpha_{ij}\left(1 - \xi_i - y_i\langle \mathbf{x_i}-\mathbf{x_j}, \mathbf{w}\rangle\right) - \sum_i \mu_i\xi_i \tag{19}$$

with

$$0 \leqslant \sum_j \alpha_{ij} \leqslant C, \quad z_i \in \mathcal{Z}_{1,3}; \qquad \mathbf{w} = \sum_{ij} y_i\,\alpha_{ij}\,(\mathbf{x_i}-\mathbf{x_j}). \tag{20}$$

The dual problem is therefore,

$$\begin{array}{ll} \max\limits_{\alpha} & \sum\limits_{i,j}\alpha_{ij} - \sum\limits_{ijkl} y_i y_k \alpha_{ij}\alpha_{kl}\langle \mathbf{x_i}-\mathbf{x_j}, \mathbf{x_k}-\mathbf{x_l}\rangle \\ \text{s.t.} & \begin{cases} 0 \leqslant \sum\limits_j \alpha_{ij} \leqslant C \\ \alpha_{ij}, \alpha_{kl} \geqslant 0, \quad z_i, z_k \in \mathcal{Z}_{1,3}; \quad z_j, z_l \in \mathcal{Z}_2 \end{cases} \end{array} \tag{21}$$

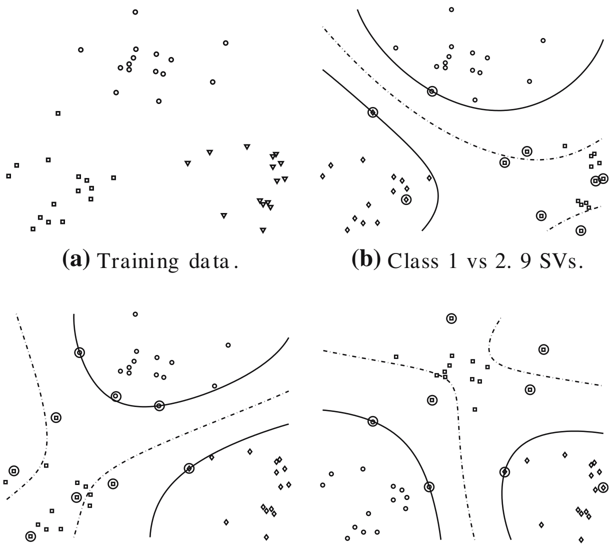
and the solution function can be written,

$$f_{\mathbf{w}}(x) = \sum_{ij}\alpha_{ij} y_i \left(k(x_i, x) - k(x_j, x)\right). \tag{22}$$

For a new entry x, output is interpreted in accordance with (13), where

$$\delta = \max_{z_j\in\mathcal{Z}_2}\left|f_{\mathbf{w}}(x_j)\right| = \max_{z_j\in\mathcal{Z}_2}\left|\langle \mathbf{w}, x_j\rangle\right|. \tag{23}$$

In Figure 1, the behaviour of the 1-v-1 tri-class machine is illustrated by using a simple linearly separable problem with a Gaussian kernel. Support vectors (SVs) are those patterns with associated null parameters, i.e. a null row or column in the parameter matrix. As expected, the number of support vectors is limited and they lie in the margin. Solid lines indicate the δ-tube for the ‘remaining vectors’ belonging to the ‘third class’ and the dotted line represents the separating hyperplane. It must be noted that values for $0 < \delta < 1$ are very low, in this example 0.1126, 0.1750 and 0.2159.



(a) Training data. **(b)** Class 1 vs 2. 9 SVs.

(c) Class 1 vs 3. 9 SVs. **(d)** Class 2 vs 3. 8 SVs.

Figure 1. Results of the 1-v-1 tri-class machine applied to a simple separable problem with 45 patterns.

5. 1-v-1 Tri-Class SVM Revised. A Third Approach

By means of a tri-class scheme, both the K-SVCR and the 1-v-1 tri-class SVM allow the incorporation of all the information contained in the training patterns when a multi-class problem is considered. For the 1-v-1 tri-class SVM, information from 'remaining patterns' is captured in a δ-tube, where δ is an optimal parameter which is automatically obtained by maximizing the margin between classes. However, this automatic tuning of the parameter leads to a computationally more expensive optimization problem. This computational effort must be reduced.

By observing the nature of the constraints in the optimization problem (18), an almost direct relation with respect to ordinal regression problems could be investigated. In this sense, Shashua and Levin [20] have recently developed a fixed margin strategy to deal with ordinal regression problems by means of large margin algorithms such as SVMs. This strategy considers all the classes at once, but without squaring the size of the training data. Hence, the procedure seeks parallel hyperplanes by separating consecutive classes through the optimization problem

$$\begin{array}{ll} \min\limits_{\mathbf{w}\in\mathcal{F};\, b_j\in\mathbb{R}} & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum\limits_i\sum\limits_j\left(\xi_i^j + \xi_i^{*j+1}\right) \\ \text{s.t.} & \left\{\begin{array}{ll} \langle\mathbf{x_i},\mathbf{w}\rangle - b_j \leqslant -1 + \xi_i^j & z_i\in\mathcal{Z}_j \\ \langle\mathbf{x_i},\mathbf{w}\rangle - b_j \geqslant 1 - \xi_i^{*j+1} & z_i\in\mathcal{Z}_{j+1} \\ \xi_i^j,\, \xi_i^{*j+1} \geqslant 0 & \end{array}\right. \end{array} \tag{24}$$

where $j = 1, \ldots, \ell - 1$.

When comparing the 1-v-1 tri-class approach (18) and the formulation in (24), it follows that (18) can be obtained from (24) when the number of categories to be considered is three, $\ell = 3$, if the constraints which have similar bias b_j are subtracted , and a double value for the margin is considered. Hence,

$$\begin{array}{ll} \min\limits_{\mathbf{w}\in\mathcal{F};b_1,b_2\in\mathbb{R}} & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum\limits_i \left(\xi_i^1 + \xi_i^2 + \xi_i^{*2} + \xi_i^{*3}\right) \\ \text{s.t.} & \begin{cases} \langle \mathbf{x_i}, \mathbf{w}\rangle - b_1 \leqslant -1 + \xi_i^1 & z_i \in \mathcal{Z}_1 \\ \langle \mathbf{x_i}, \mathbf{w}\rangle - b_1 \geqslant 1 - \xi_i^{*2} & z_i \in \mathcal{Z}_2 \\ \langle \mathbf{x_i}, \mathbf{w}\rangle - b_2 \leqslant -1 + \xi_i^2 & z_i \in \mathcal{Z}_2 \\ \langle \mathbf{x_i}, \mathbf{w}\rangle - b_2 \geqslant 1 - \xi_i^{*3} & z_i \in \mathcal{Z}_3 \\ \xi_i^1, \xi_i^2, \xi_i^{*2}, \xi_i^{*3} \geqslant 0 \end{cases} \end{array} \tag{25}$$

leads to

$$\begin{array}{ll} \min\limits_{\mathbf{w}\in\mathcal{F};b_1,b_2\in\mathbb{R}} & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum\limits_i \left(\xi_i^1 + \xi_i^2 + \xi_i^{*2} + \xi_i^{*3}\right) \\ \text{s.t.} & \begin{cases} \langle \mathbf{x_j} - \mathbf{x_i}, \mathbf{w}\rangle \geqslant 2 - \xi_i^1 - \xi_j^{*2} & z_i \in \mathcal{Z}_1;\ z_j \in \mathcal{Z}_2 \\ \langle \mathbf{x_i} - \mathbf{x_j}, \mathbf{w}\rangle \geqslant 2 - \xi_j^2 - \xi_i^{*3} & z_i \in \mathcal{Z}_3;\ z_j \in \mathcal{Z}_2 \\ \quad \xi_i^1 + \xi_i^{*2}, \xi_i^2 + \xi_i^{*3} \geqslant 0 \end{cases} \end{array} \tag{26}$$

which is the same problem as (18) but with a double margin.

Indeed, it has been demonstrated that this ordinal regression approach can be used in a similar way to the tri-class SVM in the decomposition – reconstruction multi-classification procedure established in previous sections, by separating all the patterns into three ensembles, labelled $\{-1, 0, 1\}$.

The size of the optimization problem associated to the 1-v-1 tri-class machine has been drastically reduced. Hence, if a multi-classification problem of ℓ classes is considered, where each class has the same number of patterns, (i.e. $\frac{n}{\ell}$ patterns for classes labelled ±1 and $\frac{(\ell-2)n}{\ell}$ patterns for the 0-labelled class), the first optimization problem has to fulfil a number of restrictions of $O(n^2)$, while the new version has an order of $O(n)$. When all the necessary 1-v-1 tri-class machines are considered in the multi-classification schema, $\frac{\ell(\ell-1)}{2}$, then the global number of constraints is $O(\ell^2 n)$.

In Figure 2, the performance of the novel machine is shown when it is applied with a Gaussian kernel on a non linearly separable multiclass problem. Classifiers are combined by a majority voting scheme to produce the final multiclass classification. It can be observed that a little band between classes remains unclassified since the outputs from the parallel decomposition phase assign this zone to different classes.

6. Experimental Results

In this section, experimental results are presented for several problems from the usual UCI Repository of machine learning databases [7]. A summary of the

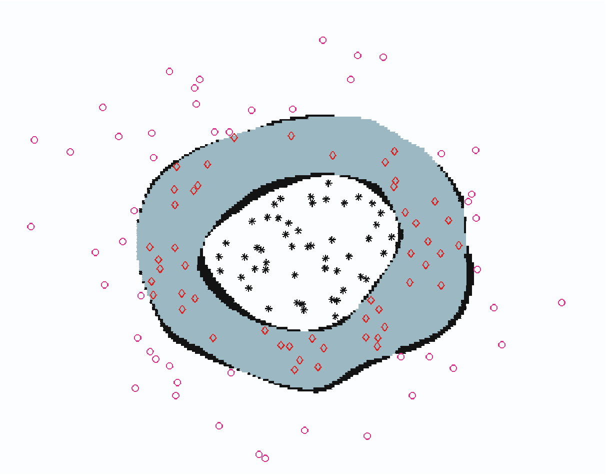


Figure 2. Results of the 1-v-1 tri-class machine applied to a simple separable problem with 45 patterns.

Table 1. Characteristics of the selected datasets from the UCI repository.

Dataset	Patterns	Classes	Features
Iris	150	3	4
Wine	178	3	13
Glass	214	6	9
Vowel	528	11	10
Vehicle	846	4	18
DNA	2000 (1186)	3	180

characteristics of the selected datasets (*Iris*, *Wine*, *Glass*, *Vowel*, *Vehicle* and *DNA*) is in Table 1. DNA dataset contains training and testing data.

The results have been obtained by following the experimental framework which was proposed by [15] and was continued in [1], but with some modifications introduced to incorporate the suggestions in [14] and [22]. Hence, training data have not been scaled for their inclusion in $[-1, +1]$, but have been normalized, (that is, mean zero and standard deviation one), in order to avoid problems with outliers. Test data are normalized accordingly.

The algorithms considered are the standard 1-v-1 and 1-v-r formulation and the 1-v-1 Tri-Class SVM in its final revised form for multi-classification. Their performance, (in the form of accuracy rate), has been evaluated on models using the Gaussian kernel,

$$k(x_i, x_j) = \exp^{-\|x_i - x_j\|^2 / 2\sigma^2} \tag{27}$$

therefore two hyperparameters must be set: the regularization term C and the width of the kernel σ. This space is explored on a two-dimensional grid with the following values: $C = [2^4, 2^3, \ldots, 2^{-10}]$ and $\gamma = [2^{12}, 2^{11}, \ldots, 2^{-2}]$, where $\gamma = \frac{1}{2\sigma^2}$.

Table 2. A comparison of the best accuracy rates using the RBF kernel.

Dataset	CV	1-v-1 (C,γ)	1-v-r (C,γ)	Tri-class (C,γ)
Iris	30	96.73 $(2^0, 2^3)$	96.00 $(2^6, 2^4)$	95.49 $(2^8, 2^{-2})$
Wine	25	98.39 $(2^{11}, 2^3)$	97.86 $(2^2, 2^3)$	97.06 $(2^7, 2^3)$
Glass	10	70.91 $(2^3, 2^1)$	71.11 $(2^9, 2^4)$	71.81 $(2^{-1}, 2^{-7})$
Vowel	10	98.95 $(2^3, 2^0)$	98.48 $(2^3, 2^{-1})$	99.36 $(2^3, 2^0)$
Vehicle	3	84.17 $(2^8, 2^4)$	86.21 $(2^8, 2^4)$	88.18 $(2^6, 2^2)$
DNA	–	95.45 $(2^3, 2^{-5})$	95.78 $(2^1, 2^{-6})$	95.86 $(2^2, 2^{-7})$

The criteria used to estimate the generalized accuracy is a ten-fold cross-validation on the whole training data, except for the DNA dataset. This procedure is repeated between 3 and 30 times, according to the size of the dataset, in order to ensure good statistical behaviour. The optimization algorithm used is the exact quadratic program-solver provided by Matlab, except for the Vowel and DNA datasets, that an iterative solver has been employed [8]. The best cross-validation mean rate among the several pairs (C, γ) is reported in Table 2.

It can be observed that similar performance results are obtained by all three approaches, however slight differences can be appreciated.

7. Conclusions and Future Work

In this paper, a new kernel machine has been designed to solve multi-classification problems. Initially, it has been proved that, by means of a tri-class scheme, the machine allows the incorporation of all the information contained in the training patterns when a multi-class problem is considered. Information from ‘remaining patterns’ is captured in a δ-tube, where δ is an optimal parameter which can be automatically obtained by maximizing the margin between classes.

New formulation with automatic tuning of parameter δ is very time-consuming, since comparisons between classes must be realized, similar to an ordinal regression procedure. An algorithm in [20] avoids making comparisons between classes when a preference learning task is performed, which speeds up the computation time considerably. However, all the hyperplanes considered must be parallel, hence the explanation power of the machine is reduced, and the use of the machine is restricted to ordinal regression. Our approach is an improvement on the machine in [20], since it is possible that hyperplanes are not parallel, which improves their explanation power, and our approach can therefore be used for multi-classification tasks.

By observing the constraints in the optimization problem, a more direct extension to ordinal regression problems is under investigation. A first natural choice would be to use a 1-v-1 tri-class SVM to solve preference learning problems, in the same way as the K-SVCR machine was developed for this utilization in [4], in accordance with the approach presented in [13]. However, it is still necessary to use constraints on the differences between patterns of different classes.

When hyperplanes are merged to obtain the final multi-class solution, only signed outputs are considered in the voting scheme, so ties between classes are considered as errors. An initiated research line is the probabilistic interpretation of the outputs in accordance with their value [11].

Acknowledgements

This study was partially supported by Junta de Andalucía grant ACPAI-2003/014, and Spanish MCyT grant TIC2002-04371-C02-01.

References

1. Anguita, D., Ridella, S. and Sterpi, D.: A New Method for Multiclass Support Vector Machines. In: *Proceedings of the IEEE IJCNN2004*. Budapest (Hungary), 2004.
2. Angulo, C.: Learning with Kernel Machines into a Multi-Class Environment. Doctoral thesis, Technical University of Catalonia. In Spanish, 2001.
3. Angulo, C. and Català, A.: A Multi-class Support Vector Machine. *Lecture Notes in Computer Science*, **1810** (2000), 55–64.
4. Angulo, C. and Català, A.: Ordinal regression with K-SVCR machines. In: J. Mira and A. Prieto (eds.), *Proceedings of IWANN 2001, Part I*, Vol. 2084 of *Lecture Notes in Computer Science*. pp. 661–668, 2001.
5. Angulo, C. and González, L.: 1-v-1 tri-class SV machine. In: *Proceedings of the 11th European Symposium on Artificial Neural Networks*. Bruges (Belgium), pp. 355–360, 2003.
6. Angulo, C., Parra, X. and Català, A.: K-SVCR. A support vector machine for multi-class classification. *Neurocomputing*, **55**(1–2), (2003) 57–77.
7. Blake, C. and Merz, C.: UCI Repository of Machine Learning Databases, 1998.
8. Canu, S., Grandvalet, Y. and Rakotomamonjy, A.: *SVM and Kernel Methods Matlab Toolbox*. Perception Systèmes et Information. INSA de Rouen, Rouen, France, 2003.
9. Cristianini, N. and Shawe-Taylor, J.: *An Introduction to Support Vector Machines and other Kernel-based Learning Methods*. Cambridge University press, 2000.
10. González, L.: Discriminative analysis using kernel vector machines support. The similarity kernel function. Doctoral thesis, University of Seville. In Spanish, 2002.
11. González, L., Angulo, C., Velasco, F., and Vílchez, M.: Máquina ℓ-SVCR con salidas probabilísticas (ℓ-SVCR machine with probabilistic outputs). *Inteligencia Artificial. Revista Iberoamericana de IA* (17) (2002) 72–82. In Spanish.
12. Hebrich, R.: *Learning Kernel Classifiers. Theory and Algorithms*. The MIT Press, 2002.
13. Herbrich, R., Graepel, T. and Obermayer, K.: *Advances in Large Margin Classifiers*, Chapt. Large Margin Rank Boundaries for Ordinal Regression, pp. 115–132. Cambridge, MA: MIT Press, 2000.
14. Hsu, C.-W., Chang, C.-C. and Lin, C.-J.: A practical guide to support vector classification. Technical report, Department of Computer Science and Information Engineering, National Taiwan University, 2003.
15. Hsu, C.-W. and Lin, C.-J.: A Comparison of methods for multiclass support vector machine. *IEEE Transactions on Neural Networks*, **13**(2), (2002) 415–425.
16. Kressel, U.: Pairwise classification and support vector machine. In B. Schölkopf, C. Burgues and A. Smola (eds.) *Advances in Kernel Methods: Support Vector Learning*, pp. 255–268, Cambridge, MA: MIT Press, 1999.

17. Mayoraz, E. and Alpaydin, E.: Support vector machines for multi-class classification. In: J. Mira and J. V. Sánchez-Andrés (eds.), *Proceedings of IWANN 1999, Part II*, Vol. 1607 of *Lecture Notes in Computer Science*, 1999.
18. Platt, J., Cristianini, N. and Shawe-Taylor, J.: Large margin DAGs for multiclass classification. *Neural Information Processing Systems*, **12** (2000).
19. Rifkin, R. and Klautau, A.: In defense of one-vs-all classification. *Journal of Machine Learning Research*, **5** (2004), 101–141.
20. Shashua, A. and Levin, A.: Taxonomy of large margin principle algorithms for ordinal regression problems. *Neural Information Processing Systems*, **16** (2002).
21. Vapnik, V.: *Statistical Learning Theory*. John Wiley & Sons, Inc., 1998.
22. Vert, J.-P., Tsuda, K. and Schölkopf, B.: *Kernel Methods in Computational Biology*, Chapt. A Primer on Kernel Methods, pp. 35–70. The MIT Press, 2004.