Available at
www.ComputerScienceWeb.com
POWERED BY SCIENCE DIRECT®

NEUROCOMPUTING

Neurocomputing 55 (2003) 57–77

www.elsevier.com/locate/neucom

K-SVCR. A support vector machine for multi-class classification

Cecilio Angulo*, Xavier Parra, Andreu Català

Knowledge Engineering Research Group (GREC), European Associated Laboratory in Intelligent Systems & Advanced Control (LEA-SICA), Vilanova i la Geltrú, E-08800, Spain

Received 8 March 2002; accepted 8 January 2003

Abstract

The problem of multi-class classification is usually solved by a decomposing and reconstruction procedure when two-class decision machines are implied. During the decomposing phase, training data are partitioned into two classes in several manners and two-class learning machines are trained. To assign the class for a new entry, machines' outputs are evaluated in a specific pulling scheme. This article introduces the "Support Vector Classification-Regression" machine for K-class classification purposes (K-SVCR), a new training algorithm with ternary outputs $\{-1, 0, +1\}$ based on Vapnik's Support Vector theory. This new machine evaluates all the training data into a 1-versus-1-versus-rest structure during the decomposing phase by using a mixed classification and regression SV Machine (SVM) formulation. For the reconstruction, a specific pulling scheme considering positive and negative votes has been designed, making the overall learning architecture more fault-tolerant as it will be demonstrated.

Keywords: Multi-classification; Support vector machines; Robustness

1. Introduction

Let

$$\mathscr{T} = \{(\mathbf{x}_p, y_p)\}_{p=1}^{\ell} \subset \Omega \times \mathscr{Y} \tag{1}$$

be a training data set following an unknown probability density function. The problem of multi-class classification from examples addresses the general problem of finding

* Corresponding author.
E-mail address: cangulo@esaii.upc.es (C. Angulo).

0925-2312/03/$ - see front matter
doi:10.1016/S0925-2312(03)00435-1

a decision function $f(\mathbf{x},\omega)$ approximation of an unknown function, defined from an input space $\Omega \subset \mathbb{R}^N$ into an unordered set of classes $\mathcal{Y} = \{\theta_1,\ldots,\theta_K\}$ having the smaller discrepancy with the real system answer.

Support Vector (SV) Machines that learn classification problems—in short SVC—are specific to binary classification problems. The SVC learning procedure constructs a decision function in the form $f(\mathbf{x},\omega) = sign(h(\mathbf{x},\omega))$ with outputs $\{\pm 1\}$, where

$$h(\mathbf{x},\omega) = \langle \omega, \phi(\mathbf{x})\rangle + b \tag{2}$$

is a separating hyperplane in some 'feature space' $\mathcal{F}$, with $\omega \in \mathcal{F}$, $b \in \mathbb{R}$, $\phi : \Omega \to \mathcal{F}$ being a non-linear mapping inserting the original input space Ω into a usually high-dimensional space. The space $\mathcal{F}$ is dotted with an inner product,

$$k(\mathbf{x}',\mathbf{x}) := \langle \phi(\mathbf{x}'), \phi(\mathbf{x})\rangle,$$

accomplishing Mercer's theorem [19].

The large-scale problem, $K > 2$, K being the number of classes to be considered, is typically solved by the combination of two-class decision functions: a decomposition scheme transforms the K-partition into an ensemble of L bi-partitions, $f_1,\ldots,f_L$; then a reconstruction method makes the fusion of the L learning machines' outputs to select one, or none, of the K classes. In this article, a new SV algorithm to be used in the decomposition scheme, K-SVCR, is presented. For the case $K > 2$, learning machines will be designed to assign output $+1$ or -1 if the entry belongs to the classes to be explicitly separated, and output 0 if the pattern's class is not the key matter of this machine. The computed separating hyperplane $h(\mathbf{x},\omega)$ is forced to cover all the training patterns to be '0-labelled', i.e. entries belonging to classes not being explicitly separated. The new machine is associated to a quadratic programming (QP) problem that could be seen in the middle between the SVC method and the SV learning method for regression estimation—in short, SVR—with 0 being the only value to be regressed.

The rest of this article is organized as follows. In the next section, a brief presentation of SVMs for the general no separable case is introduced with the aim to be used in the theoretical development of the new algorithm. Section 3 shows available procedures in the literature to construct multi-class SVCs. In Section 4, we develop the theory of the novel method to be used for multi-class classification, K-SVCR. Examples of K-SVCR classification are depicted in Section 5 and results are discussed. An adequate reconstruction procedure is defined in Section 6 in such a form that the new multi-class classification architecture is more robust against errors than standard procedures. In the last section, concluding remarks and further research to be developed are presented.

2. Support vector machines

In the SVC binary pattern recognition problem, one would construct an hyperplane in the form of Eq. (2) separating two classes, $y \in \{-1,+1\}$, so that the distance between the optimal hyperplane and the nearest training pattern be maximal, to enforce the generalization of the learning machine [6].

The optimal hyperplane, in canonical form, is found by solving the following constrained QP problem:

$$\arg\min W_{\mathrm{SVC}}(\omega, b, \xi) = \tfrac{1}{2}\|\omega\|^2_{\mathscr{F}} + C\sum_{i=1}^{\ell}\xi_i, \tag{3}$$

subject to $y_i \cdot (\langle \omega, \phi(\mathbf{x}_i)\rangle + b) \geqslant 1 - \xi_i$, $i = 1, \ldots, \ell$, with slack variables $\xi_i \geqslant 0$.

To generalize the SV algorithm to the regression estimation problem, an analogue of the margin is constructed in the space of the target values, $y \in \mathbb{R}$, by using Vapnik's ε-insensitive loss function

$$|y - f(\mathbf{x})|_{\varepsilon} = \max\{0, |y - f(\mathbf{x})| - \varepsilon\}. \tag{4}$$

For an 'a priori' chosen $\varepsilon \geqslant 0$, the associated QP problem is

$$\arg\min W_{\mathrm{SVR}}(\omega, b, \varphi, \varphi^*) = \tfrac{1}{2}\|\omega\|^2_{\mathscr{F}} + D\sum_{i=1}^{\ell}(\varphi_i + \varphi_i^*), \tag{5}$$

subject to $-\varepsilon - \varphi_i^* \leqslant \langle \omega, \phi(\mathbf{x}_i)\rangle + b \leqslant \varepsilon + \varphi_i$, $i = 1, \ldots, \ell$, with slack variables $\varphi_i, \varphi_i^* \geqslant 0$.

3. Multi-class SVMs

In the standard decomposing scheme of a multi-classification problem into dichotomies [7,19], K SVCs are trained over all the training patterns: the ith $1 - v - r$ SVC—short for 'one versus the rest'—must assign label $+1$ to the examples in the ith class, and label -1 to all the other patterns.

Another general decomposing method constructs all the $L = K(K-1)/2$ possible binary machines from a K-class training set, each SVC being trained on only two out of all K classes. Trained SVCs by this method will be referred to as $1 - v - 1$ SVCs—short for 'one versus one'.

The usual reconstruction method associated to these decomposing schemes is a parallel structure: when a new entry is presented, each binary learned machine provides one output concerning the classes involved in the training phase, then an algorithm interprets these two-class classifier outputs to determine the label to be assigned to the input. There exist several combinatorial algorithms for the outputs, voting schemes being the more simple ones and the final output not depending on the numerical output range for each classifier. On the other hand, if it could be assumed that all the learning machines' outputs are in the same rank, then numeric values are comparable and the 'winner-takes-all' scheme can be applied. However, the output scale of a SVC is not robust, since it depends on just a few points, the support vectors [12], so voting schemes are the more practical techniques for the reconstruction phase with SVMs. More sophisticated combinations can be made if machines' outputs are considered probabilities [13] or SVCs are adapted to produce posterior class probabilities, as pointed out in [15] and references therein.

All the above schemes can be considered as $\{1-v-r,\ \text{parallel}\}$ or $\{1-v-1,\ \text{parallel}\}$ decomposing–reconstruction architecture approaches, with possible additional output treatment. The $\{1-v-1,\ \text{parallel}\}$ approach is, in general, preferable [13] because outputs generated for $i-v-j$ SVCs make no sense when the entry under evaluation belongs neither to the ith class nor to the jth class (*incompetent classifiers*).

In [2] are drawn the two main drawbacks for an $1-v-1$ decomposition scheme: the number of classifiers is high and only two classes are involved for each classifier, so variance is higher and no information is given for the rest of the classes. By applying some ideas from the *Error Correcting Output Codes* technique [8]—in short, ECOC—in [2,1], the authors propose a combinatorial distribution of the training patterns of different classes in two output classes, ± 1, in order to maximize the Hamming distance between classes, so redundancy in the pattern information is added. Nevertheless, $2^{(K-1)}-1$ different classifiers can be constructed and no algorithm exists to make the best choice between them. For example, if $K=10$ classes, a $1-v-1$ scheme needs 45 binary classifiers, whereas an ECOC decomposition according to [2] must choose between 511 possibilities, or 5110 if the approach in [1] is used.

In addition to the two general parallel decomposing–reconstruction methodologies and the {ECOC, parallel} architecture, it is possible to construct a multi-classification structure by combining $1-v-1$ SVCs into a decision tree, able to handle many classes, generating a $\{1-v-1,\ \text{tree}\}$-like approach [4,17].

Finally, in the last few years, several approaches have been developed considering all the classes at once multi-class SVCs—in short KSVMC [20,9,4]. In any case, the associated QP problem to be solved is very large: at least, $2\ell\cdot(K-1)$ inequality and K equality restrictions need to be considered.

4. The K-SVCR learning machine

Given the training set $\mathscr{T}$ defined in Eq. (1), a decision function $f(\mathbf{x},\omega)$ based on a hyperplane similar to Eq. (2) should be found with outputs $\{-1,0,+1\}$:

$$f(\mathbf{x}_p)=\begin{cases}+1, & p=1,\ldots,\ell_1,\\ -1, & p=\ell_1+1,\ldots,\ell_1+\ell_2,\\ 0, & p=\ell_1+\ell_2+1,\ldots,\ell,\end{cases} \tag{6}$$

where, without loss of generality, $\ell_{12}=\ell_1+\ell_2$ patterns belong to the two classes to be separated, and $\ell_3=\ell-\ell_{12}$ patterns belong to some different class—they will be labelled 0.

Obviously, in general, there does not exist any hyperplane accomplishing Eq. (6) in the input space Ω; hence it is useless to look for a linear solution to the problem in the original space. However, if input space is inserted via a nonlinear map into a high-enough dimensional feature space $\mathscr{F}$, then hyperplanes in this new space have increased capacity to accomplish the imposed constraints, and it will be possible to find a solution. For instance, when the QP problem leading to the SVC solution is

solved, it is common practice to formulate the problem adding the constraint $b = 0$, which is equivalent to requiring that the optimal hyperplane contain the origin. This is considered a mild restriction for high-dimensional spaces, since it is equivalent to reducing the number of degrees of freedom by one [6]. Requirements for a K-SVCR learning machine are higher, the optimal hyperplane must contain all the ℓ_3 0-labelled training patterns.

If a positive parameter δ is introduced for the 0-labelled entries to allow little deviations from the hyperplane, the constrained optimization problem associated to the K-SVCR method, for the most general case, is defined as: for $0 \leqslant \delta < 1$ chosen a priori,

$$\arg\min W_{K\text{-SVCR}}(\omega, b, \xi, \varphi, \varphi^*) = \tfrac{1}{2}\|\omega\|^2_{\mathscr{F}} + C\sum_{i=1}^{\ell}\xi_i + D\sum_{i=1}^{\ell}(\varphi_i + \varphi_i^*), \tag{7}$$

subject to

$$\begin{aligned} &y_i \cdot (\langle \omega, \phi(\mathbf{x}_i)\rangle + b) \geqslant 1 - \xi_i, \quad i = 1, \ldots, \ell_{12}, \\ &-\delta - \varphi_i^* \leqslant \langle \omega, \phi(\mathbf{x}_i)\rangle + b \leqslant \delta + \varphi_i, \quad i = \ell_{12} + 1, \ldots, \ell \end{aligned} \tag{8}$$

with slack variables $\xi_i, \varphi_i, \varphi_i^* \geqslant 0$. The positive parameter δ must be restricted to be lower than 1 to avoid overlapping between classes in Eq. (8) when slack variables are null.

A solution for the problem defined by Eqs. (7) and (8) can be found by locating the saddle point of the Lagrangian

$$\begin{aligned} L_{K\text{-SVCR}} = {} & \tfrac{1}{2}\|\omega\|^2_{\mathscr{F}} + C\sum_{i=1}^{\ell}\xi_i + D\sum_{i=1}^{\ell}(\varphi_i + \varphi_i^*) \\ & - \sum_{i=1}^{\ell_{12}}\alpha_i[y_i \cdot (\langle \omega, \phi(\mathbf{x}_i)\rangle + b) - 1 + \xi_i] - \sum_{i=\ell_{12}+1}^{\ell}\mu_i\xi_i \\ & + \sum_{i=\ell_{12}+1}^{\ell}\beta_i[\langle \omega, \phi(\mathbf{x}_i)\rangle + b - \delta - \varphi_i] - \sum_{i=\ell_{12}+1}^{\ell}\eta_i\varphi_i \\ & - \sum_{i=\ell_{12}+1}^{\ell}\beta_i^*[\langle \omega, \phi(\mathbf{x}_i)\rangle + b + \delta + \varphi_i^*] - \sum_{i=\ell_{12}+1}^{\ell}\eta_i^*\varphi_i^*, \end{aligned}$$

subject to α_i, μ_i, β_i, β_i^*, η_i, $\eta_i^* \geqslant 0$, which has to be maximized with respect to these dual variables and minimized with respect to the primal variables $\omega, b, \xi_i, \varphi_i, \varphi_i^*$.

Using Wolfe's dual formulation of the Lagrangian, saddle point conditions lead to

$$\omega = \sum_{i=1}^{\ell_{12}}\alpha_i y_i \phi(\mathbf{x}_i) - \sum_{i=\ell_{12}+1}^{\ell}(\beta_i - \beta_i^*)\phi(\mathbf{x}_i),$$

$$0 = \sum_{i=1}^{\ell_{12}} \alpha_i y_i - \sum_{i=\ell_{12}+1}^{\ell} (\beta_i - \beta_i^*)$$

and

$$0 \leqslant \alpha_i \leqslant C, \quad i = 1, \ldots, \ell_{12},$$
$$0 \leqslant \beta_i, \quad \beta_i^* \leqslant D, \quad i = \ell_{12} + 1, \ldots, \ell.$$

Finally, if it is defined as

$$\gamma_i = \alpha_i y_i, \quad i = 1, \ldots, \ell_{12},$$
$$\gamma_i = \beta_i, \quad i = \ell_{12} + 1, \ldots, \ell,$$
$$\gamma_i = \beta_{i-\ell_3}^*, \quad i = \ell + 1, \ldots, \ell + \ell_3,$$
$$\mathbf{x}_i = \mathbf{x}_{i-\ell_3}, \quad i = \ell + 1, \ldots, \ell + \ell_3,$$

the dual formulation can be expressed as: for $0 \leqslant \delta < 1$ chosen a priori

$$\arg\min L(\gamma) = \tfrac{1}{2}\gamma^{\mathrm{T}} \cdot \mathbf{H} \cdot \gamma + \mathbf{c}^{\mathrm{T}} \cdot \gamma$$

with

$$\gamma^{\mathrm{T}} = (\gamma_1, \ldots, \gamma_\ell, \gamma_{\ell+1}, \ldots, \gamma_{\ell+\ell_3}) \in \mathbb{R}^{\ell_{12}+\ell_3+\ell_3},$$
$$\mathbf{c}^{\mathrm{T}} = \left(\frac{-1}{y_1}, \ldots, \frac{-1}{y_{\ell_{12}}}, \delta, \ldots, \delta \right) \in \mathbb{R}^{\ell_{12}+\ell_3+\ell_3},$$
$$\mathbf{H} = \begin{pmatrix} (k(\mathbf{x}_i, \mathbf{x}_j)) & -(k(\mathbf{x}_i, \mathbf{x}_j)) & (k(\mathbf{x}_i, \mathbf{x}_j)) \\ -(k(\mathbf{x}_i, \mathbf{x}_j)) & (k(\mathbf{x}_i, \mathbf{x}_j)) & -(k(\mathbf{x}_i, \mathbf{x}_j)) \\ (k(\mathbf{x}_i, \mathbf{x}_j)) & -(k(\mathbf{x}_i, \mathbf{x}_j)) & (k(\mathbf{x}_i, \mathbf{x}_j)) \end{pmatrix},$$
$$\mathbf{H} = \mathbf{H}^{\mathrm{T}} \in \mathscr{M}(\mathbb{R}^{\ell_{12}+\ell_3+\ell_3}, \mathbb{R}^{\ell_{12}+\ell_3+\ell_3}),$$

subject to

$$0 \leqslant \gamma_i \cdot y_i \leqslant C, \quad i = 1, \ldots, \ell_{12},$$
$$0 \leqslant \gamma_i \leqslant D, \quad i = \ell_{12}, \ldots, \ell + \ell_3$$

and

$$\sum_{i=1}^{\ell_{12}} \gamma_i = \sum_{i=\ell_{12}+1}^{\ell} \gamma_i - \sum_{i=\ell+1}^{\ell+\ell_3} \gamma_i. \tag{9}$$

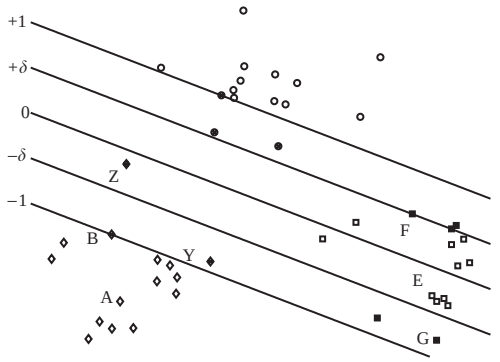

Fig. 1. Working zones for a K-SVCR with linear kernel on a three-classes problem.

The hyperplane decision function can be written as

$$f(\mathbf{x}) = \begin{cases} +1 & \text{if } \sum_{i=1}^{\mathrm{SV}} v_i k(\mathbf{x}_i, \mathbf{x}) + b \geqslant \delta, \\ -1 & \text{if } \sum_{i=1}^{\mathrm{SV}} v_i k(\mathbf{x}_i, \mathbf{x}) + b \leqslant -\delta, \\ 0 & \text{otherwise} \end{cases}$$

or alternatively,

$$f(\mathbf{x}) = sign(g(\mathbf{x}) \cdot |g(\mathbf{x})|_\delta), \tag{10}$$

where

$$g(\mathbf{x}) = \sum_{i=1}^{\mathrm{SV}} v_i k(\mathbf{x}_i, \mathbf{x}) + b,$$

being

$$v_i = \gamma_i, \quad i = 1, \ldots, \ell_{12},$$

$$v_i = \gamma_{i+\ell_3} - \gamma_i, \quad i = \ell_{12} + 1, \ldots, \ell$$

and b is calculated from Eq. (8) on the support vectors. Now, constraint associated to the term b by Eq. (9) can be written as

$$\sum_{i=1}^{\mathrm{SV}} v_i = 0.$$

In Fig. 1 are depicted the working zones for a K-SVCR on a three-class problem. For illustration purposes, the original space has been considered as the feature space, i.e. a linear kernel.

- *Patterns with output* -1: A, well-classified pattern ($v_i = 0$); B, support vector ($0 < v_i < C$); Y, error vector not migrated ($v_i = C$); Z, error vector migrated to 0-labelled class ($v_i = C$); there could exist an error vector migrated to $+1$-labelled class.
- *Patterns with output* $+1$: have a similar discussion.
- *Patterns with output* 0: E, well-classified pattern ($v_i = 0$); F, support vector on the frontier with $+1$-labelled zone ($-D < v_i < 0$); similarly, a support vector on the other frontier accomplishes $0 < v_i < D$; G, error vector migrated to -1-labelled class ($\gamma_i = D$); a similar discussion could be done for a migration to $+1$-labelled class ($\gamma_i = -D$).

The new methodology is consistent with the standard bi-class SVC paradigm because $K = 2$ classes implies $\ell_3 = 0$, so optimization problems defined by Eqs. (3), (7) and (8) are equivalents. Even for the case $K > 2$, if $\delta = 0$, then the decision function based on Eq. (2) is the same as those originally defined by Eq. (10), because

$$|g(\mathbf{x})|_{\delta_{|\delta=0}} = 0 \quad \Leftrightarrow \quad g(\mathbf{x}) = 0.$$

Nonetheless, it must be noted that the imposition $\delta = 0$ involves no generalization for the 0-labelled classes, no sparsity in the 0-label support vectors set [18] and higher computational cost.

Even if the matter of K-SVCR is pattern recognition, the ε-insensitive loss function employed in the SVR method is used in the new formulation for output $y_i = 0$. The K-SVCR general problem formulated by Eqs. (7) and (8) could be named Soft-SVC$_{12}$+ Soft-SVR$_3$ classification problem, indicating the fusion between SVC and SVR style restrictions with both soft margins over the distinguished three classes of patterns for classification purposes. Making null in several ways the slack variables ξ_i, φ_i, φ_i^*, a solution for the other three different $\{\text{Hard/Soft}\}$-SVC$_{12}$ + $\{\text{Hard/Soft}\}$-SVR$_3$ classification problems is straightforward.

The proposed learning machine improves the standard structures employed in the decomposing method of a multi-class classification procedure: in brief, K-SVCR is trained to make a focused partition between two classes, as does a $1-v-1$ decomposing

Table 1
Comparison of several approaches to multi-class architecture complexity of several approaches

	# Machines	# Variables	# Restrictions $\leqslant$
$1-v-r$	$K = 5$	$\ell = 1000$	$2(V+1) = 2002$
$1-v-1$	$\binom{K}{2} = 10$	$2\ell/K = 400$	$2(V+1) = 802$
ECOC [2]	$\leqslant 2\ell \cdot (K-1) = 15$	$\ell = 1000$	$2(V+1) = 2002$
ECOC [1]	$\leqslant K(2\ell \cdot (K-1)) = 75$	$\leqslant \ell = 1000$	$\leqslant 2(V+1) = 2002$
DAGSVM	$\binom{K}{2} = 10$	$2\ell/K = 400$	$2(V+1) = 802$
KSVMC [20]	1	$\ell(K-1) = 4000$	$2(V+K) = 8010$
KSVMC [9]	1	$\ell(K-1) = 4000$	$2(V+K) = 8010$
KSVMC [4]	1	$\ell(K-1) = 4000$	$2(V+K) = 8010$
KSVCR	$\binom{K}{2} = 10$	$\ell = 1000$	$2(V+1) = 2002$

A five-class example, having 200 patterns in each class, has been used for illustration.

scheme, giving always useful outputs because patterns in the other classes are concerned with label 0. From another point of view, each (i,j)-K-SVCR is trained over all the patterns set, as does a $1-v-r$ decomposing scheme, being considered in the same time a 'specialized training' on two classes from the ensemble.

For multi-class SVMs structures considering all the classes at once, it is only necessary to solve one QP problem, but the size of this problem is very large. A comparison of the sizes of the different multi-class approaches is displayed in Table 1 [3].

5. *K*-SVCR data experiments

To evaluate the performance of the new K-SVCR algorithm, several experiments with artificial data in $\mathbb{R}^2$ have been implemented on the MathWorks product, Matlab v5.3, on Intel Pentium III 500 MHz. The QP problem is solved with the standard Matlab routine. The major objectives are:

- to illustrate the influence of the δ parameter on the final classification results and the number of support vectors.
- to justify the new mixed classification and regression option for multi-classification purposes, K-SVCR machine, over a regression machine, SVR, doing classification tasks. It will be shown that mixed restrictions are preferable to pure regression restrictions.
- to show the performance of the proposed algorithm on artificial no separable data.

Parameter δ

The δ parameter influence and the comparison with SVR will be made on separable artificial data. Two training sets, $\mathscr{T}_1$ and $\mathscr{T}_2$, have been generated for the separable case following a gaussian distribution on $\mathbb{R}^2$, with 150 patterns equally distributed in three classes, Fig. 2. Similar examples are used in [11] to show the 'pairwise' classification efficiency.

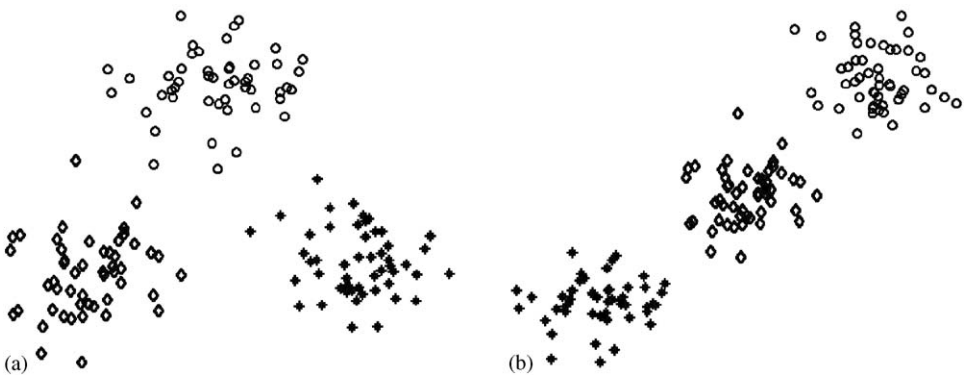

Fig. 2. Training sets for the separable case. (a) Training set $\mathscr{T}_1$; (b) training set $\mathscr{T}_2$.

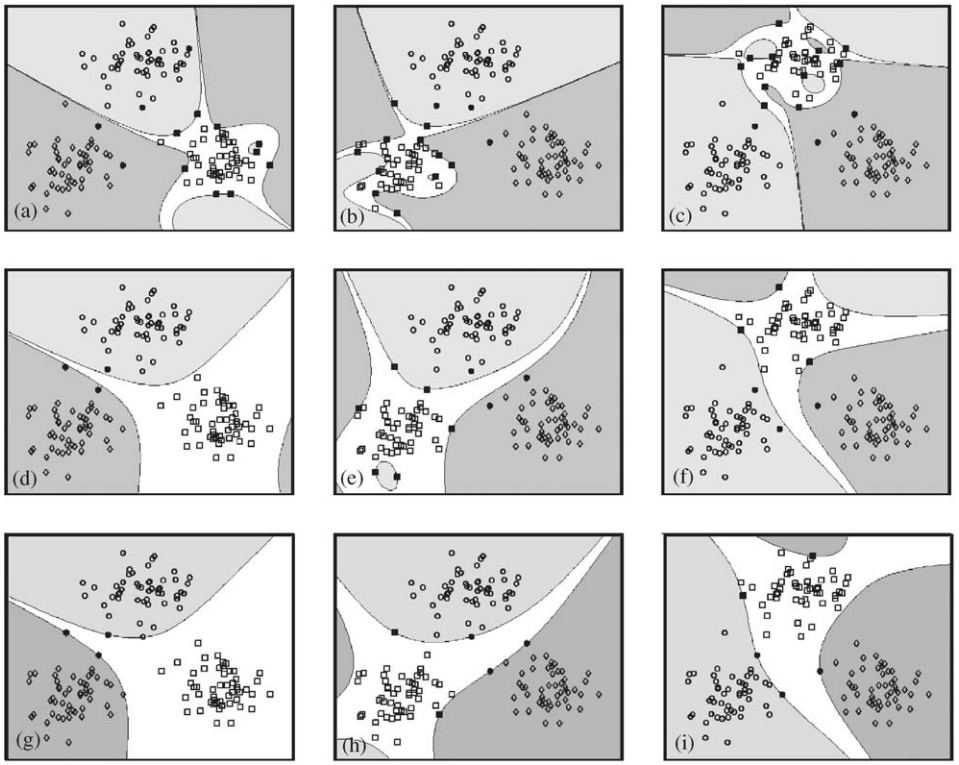

Fig. 3. Different δ insensitivity levels on training set $\mathscr{T}_1$. Displayed quantities are: insensitivity level (δ), training time (t) and number of support vectors (nsv). (a) $\delta = 0.050$, $t = 114.8$ s, $nsv = 13$; (b) $\delta = 0.050$, $t = 121.0$ s, $nsv = 14$; (c) $\delta = 0.050$, $t = 138.2$ s, $nsv = 14$; (d) $\delta = 0.500$, $t = 86.0$ s, $nsv = 3$; (e) $\delta = 0.500$, $t = 98.1$ s, $nsv = 9$; (f) $\delta = 0.500$, $t = 96.2$ s, $nsv = 6$; (g) $\delta = 0.999$, $t = 89.1$ s, $nsv = 3$; (h) $\delta = 0.999$, $t = 87.9$ s, $nsv = 5$; (i) $\delta = 0.999$, $t = 96.8$ s, $nsv = 5$.

Restrictions containing parameter δ introduced by the K-SVCR machine on the ℓ_3 0-labelled patterns have a high influence on the optimal hyperplane determination. In Fig. 3 is displayed the decision function for the three possible $1-v-1-v-r$ K-SVCR when the δ insensitivity level is increased. A polynomial kernel of degree 4 has been employed, avoiding the patterns of gaussian distribution known beforehand. It is evidenced that a lower parameter δ implies a lower generalization on the 0-labelled patterns space, a higher computational time and a higher number of support vectors.

Comparison with SVR

For this purpose, on the separable data set $\mathscr{T}_2$, K-SVCRs and standard SVRs with output values $\{-1, 0, +1\}$ have been trained, with all the output combinations (three

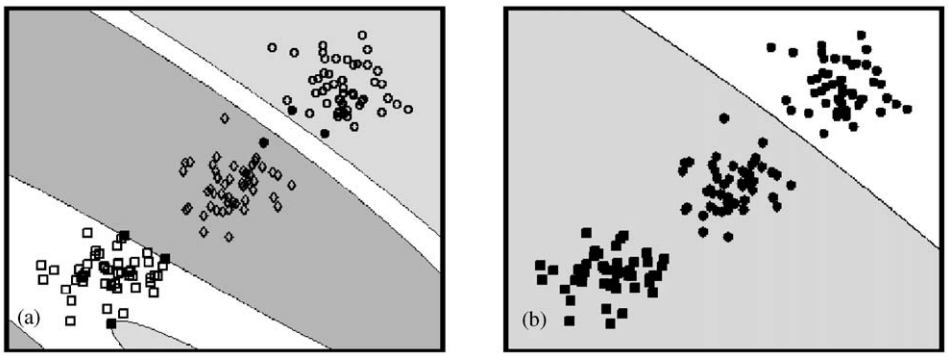

Fig. 4. Training results on the data set $\mathscr{T}_2$. (a) K-SVCR training on $\mathscr{T}_2$; (b) SVR trained on $\mathscr{T}_2$.

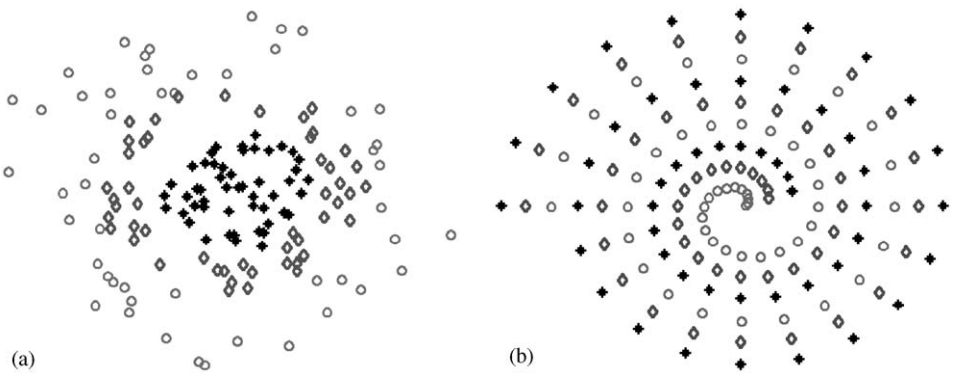

Fig. 5. Training sets for the no separable case. (a) Training set $\mathscr{T}_3$; (b) training set $\mathscr{T}_4$.

machines). For the training procedure has been used a parameter $C = \infty$ (and $D = \infty$) because classes are separable, and a middle-level insensitivity parameter $\delta = 0.5$.

When a polynomial kernel with degree 3 is employed to generate the feature space, the following results are obtained: K-SVCRs take 93.0, 86.3 and 92.2 s to be correctly trained and the optimal hyperplane expands on 6, 4 and 5 support vectors, respectively. Meanwhile, SVRs take 375.8, 365.9 and 367.6 s for training and patterns classified as bad remain, even if all the 150 training patterns are considered as support vectors (Fig. 4). Similar results are obtained for different kernel and δ parameters choices.

SVR exhibits a bad behaviour in the classification because the feature space generated for the polynomial kernel has not enough capacity to match all the restrictions. Conversely, for a K-SVCR the Hilbert space $\mathscr{F}$ is large enough to separate the classes.

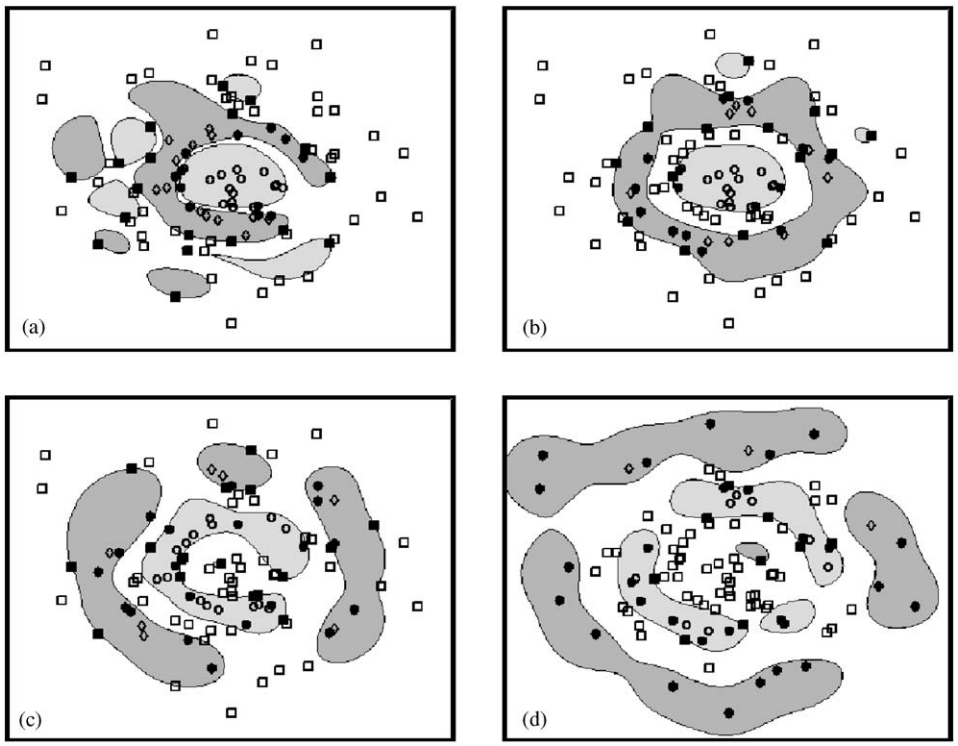

Fig. 6. Training results on data set $\mathscr{T}_3$ by using gaussian kernels when different labels choices are done. (a) 1–2–3,4,5 K-SVCR; (b) 1–3–2,4,5 K-SVCR; (c) 2–4–1,3,5 K-SVCR; (d) 3–5–1,2,4 K-SVCR.

Experiments illustrate that 0-label regression-like restrictions are not high-level exigencies for a K-SVCR.

Artificial no separable data

For the no separable case, training sets $\mathscr{T}_3$ and $\mathscr{T}_4$ have been generated on $\mathbb{R}^2$. Fig. 5(a) shows the 100 patterns data set following a gaussian distribution on five classes and Fig. 5(b) displays the spiral three-class data set.

In Fig. 6 are displayed some trained K-SVCR machines obtained by using a gaussian kernel, $\sigma = 0.45$, and $\delta = 0.75$.

Polynomial kernels have also been considered (degree 10 and $\delta = 0.75$) to prove the machine efficiency on a finite VC-dimension feature space. Results are summarized in Fig. 7 and Table 2.

In training set $\mathscr{T}_4$, three classes are considered on nested spirals. If gaussian kernels with $\sigma = 0.5$ and $\delta = 0.5$ are used, results as those of Fig. 8 are obtained.

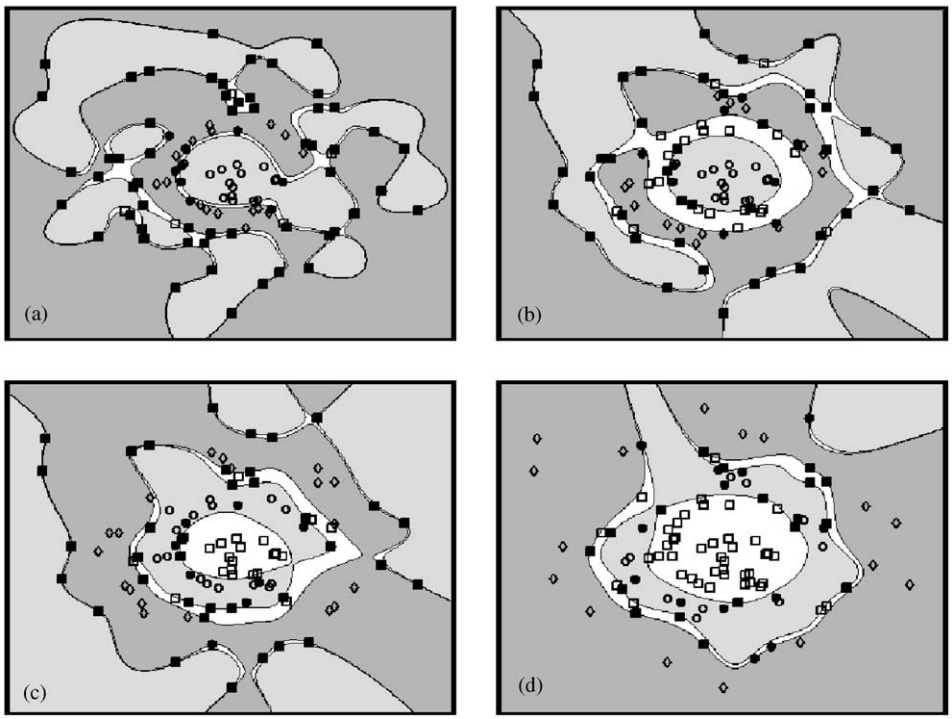

Fig. 7. Training results on data set $\mathscr{T}_3$ by using polynomial kernels when different label choices are done. (a) 1–2–3,4,5 K-SVCR; (b) 1–3–2,4,5 K-SVCR; (c) 2–4–1,3,5; (d) 3–5–1,2,4 K-SVCR.

Table 2
Training results on data set $\mathscr{T}_3$ by using polynomial kernels

Labels	1–2–3,4,5	1–3–2,4,5	1–4–2,3,5	1–5–2,3,4	2–3–1,4,5
Time	181.8	79.1	101.5	71.9	114.7
nsv	64	50	48	33	64
Labels	2–4–1,3,5	2–5–1,3,4	3–4–1,2,5	3–5–1,3,4	4–5–1,2,3
Time	70.5	31.6	78.3	65.4	54.1
nsv	44	100	48	31	20

Time and number of support vectors are displayed for all the label combinations.

6. Robust reconstruction procedure

In the previous section, it has been demonstrated that the K-SVCR machine improves standard algorithms treating two-class classification problems during the decomposing phase of a general multi-class scheme: by focusing the learning on two classes, but using all the disposable information on the patterns. Now, a second principal advantage

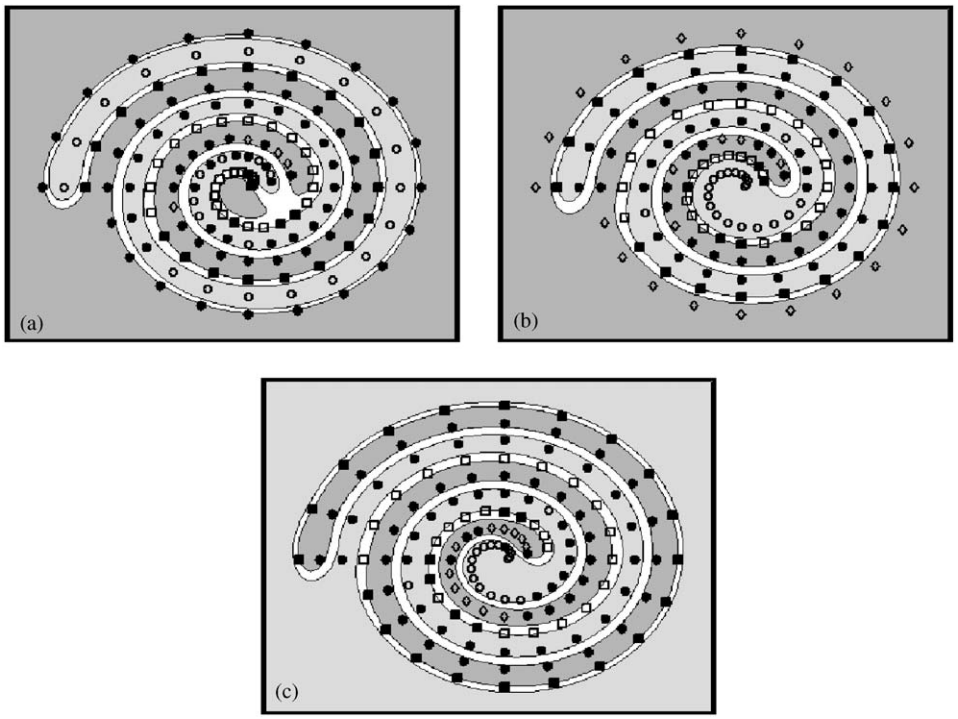

Fig. 8. Training results on data set $\mathscr{T}_4$ by using gaussian kernels when different labels choices are performed. (a) 1–2–3 K-SVCR; (b) 1–3–2 K-SVCR; (c) 2–3–1 K-SVCR.

will be enunciated, the robustness of the reconstruction procedure. To make evident this assertion, a few definitions must be done.

Let f_k be a $k-v-r$, $i-v-j$ or $i-v-j-v-r$ classifier; let $z^k = h_k(\mathbf{x}, \omega)$ be the resulting numerical value for the machine f_k, when an entry $\mathbf{x}$ labelled $\theta_m \in \mathscr{Y}$ is considered; and let $s^k = f_k(\mathbf{x}, \omega, \delta) \in \{-1, 0, +1\}$ be its sign when δ insensitivity is applied. Questions arising in the reconstruction procedure are: How should a 'translate function' $\Theta(s^k)$ interpret the machines' outputs? How have these translations to be combined for a posterior 'combinatorial function' $\Psi(\{\Theta(s^k)\}_{k=1}^{L})$ to determine the final multi-class architecture output for the entry?

Definition 1. Let f_k be a two-class classifier. It is defined as

- *positive precise translation*, the one having as 'translate function':

$$\Theta_{\mathrm{PP}}(s^k) = \begin{cases} \theta_i & \text{if } s^k = +1, \\ \theta_j & \text{if } s^k = -1, \\ Y/\{\theta_i, \theta_j\} & \text{if } s^k = 0, \end{cases}$$

- *negative precise translation*, the one having as 'translate function':

$$\Theta_{\mathrm{NP}}(s^k) = \neg\mathscr{C}(\Theta_{\mathrm{PP}}(s^k)) = \begin{cases} \neg(Y/\theta_i) & \text{if } s^k = +1, \\ \neg(Y/\theta_j) & \text{if } s^k = -1, \\ \neg\{\theta_i, \theta_j\} & \text{if } s^k = 0, \end{cases}$$

where $\mathscr{C}$ is the complementary set and $\neg$ is the negation operator,

- *mixed precise translation*, the one having as 'translate function':

$$\Theta_{\mathrm{MP}}(s^k) = \begin{cases} \theta_i & \text{if } s^k = +1, \\ \theta_j & \text{if } s^k = -1, \\ \neg\{\theta_i, \theta_j\} & \text{if } s^k = 0, \end{cases} \tag{11}$$

- *positive imprecise translation*, the one having as 'translate function':

$$\Theta_{\mathrm{PI}}(s^k) = \begin{cases} \theta_i & \text{if } s^k = +1, \\ \theta_j & \text{if } s^k = -1, \\ Y & \text{if } s^k = 0, \end{cases}$$

- *negative imprecise translation*, the one having as 'translate function':

$$\Theta_{\mathrm{NI}}(s^k) = \neg\mathscr{C}(\Theta_{\mathrm{NP}}(s^k)) = \begin{cases} \neg(Y/\theta_i) & \text{if } s^k = +1 \\ \neg(Y/\theta_j) & \text{if } s^k = -1, \\ \emptyset & \text{if } s^k = 0. \end{cases}$$

What do these translate functions mean? In the first case, $\Theta_{\mathrm{PP}}(s^k)$, if the output is not zero then the entry positively belongs to a certain class, else the entry is in the 'other classes' set. The translate function ensures that separating hyperplanes generated by the learning machines are completely correct. The counterpart negative precise translation $\Theta_{\mathrm{NP}}(s^k)$ has a symmetric sense.

For the third type of translation, mixed precise, if the output is not zero then the function positively assigns the entry to a class, else the translate function is not completely sure for the 0-label zone and the translation is referred to the main two classes concerned in a negative sentence. This translation is precise for ± 1 outputs, but for a 0 output the entry could be considered 'other class', but not necessarily a class in $\mathscr{Y}$.

Finally, the imprecise translations doubt about generated hyperplanes. This sort of translation is fine when some classes are representing a danger or important cost, for example, if clinical analysis about a dangerous illness are being evaluated, then a zero output is preferable to a wrong answer.

The mixed precise translate function on K-SVCR machines has three important features enclosing all the other translations' advantages:

(1) Precision is assumed in the hyperplanes.
(2) It is possible to work with incomplete class sets.

(3) It is possible to control classification generalization on some classes by the δ parameter.

In the following section, it will be demonstrated that additional robustness behaviour is generated when a 'combinatorial function' Ψ is appropriately defined for the reconstruction phase.

6.1. Combinatorial function and robustness analysis

The mixed precise 'translate function' generates positive and negative votes, so the most natural form to combine them is their addition. The assigned class will be the one getting most votes.

Definition 2. Let $\{\Theta(s^k)\}_{k=1}^{L}$ be the interpretation function ensemble output of the $L=\binom{K}{2}$ parallel $1-v-1-v-r$ SVMCs in the form (11) and let Ψ_r be the result of adding positive and negative votes for class θ_r; then it is defined as *combinatorial function* for a $1-v-1-v-r$ SVMC multi-class architecture

$$\Psi(\{\Theta(s^k)\}) = \arg\max_i \Psi_i, \quad i=1,\ldots,L.$$

This definition of the combinatorial function allows to enunciate:

Proposition 3. *Let F be a complete multi-class architecture with $1-v-1-v-r$ SVMC parallel decomposition procedure and mixed precise translation based on combinatorial reconstruction. If all the classifiers' outputs are correct on an entry* $\mathbf{x}$ *with associated label θ_r, then*

$$\Psi_r = K-1,$$

$$\Psi_i = -K+2, \quad \forall i \neq r.$$

Proof. In the proposed multi-class architecture, $K-1$ classifiers are concerned with not zero output on the class θ_r, so Ψ_r adds $K-1$ positive votes when all the hyperplanes are correct. Besides, no negative votes are collected because zero outputs, class θ_r in the 'other classes' group, are interpreted as negative votes for ±1-labelled concerned classes. Each class $\theta_{i\neq r}$ collects only negative votes: no votes are added if θ_r is the opponent class to θ_i, no votes are added if θ_r and θ_i are both in the 0-labelled class group, and negative votes are added if θ_r is in 0-labelled class and θ_i is not. Hence, $\Psi_i = -K+2$, $\forall i \neq r$. □

Lemma 4. *Let $d_{ij}^F = |\Psi_i^F - \Psi_j^F|$ be the distance between two classes pulling results when a multi-class architecture with $1-v-1-v-r$ SVMC parallel decomposition procedure and mixed precise translation based on combinatorial reconstruction F is considered. If some classifier on F is lightly changed and a (new) error takes place, ε,*

then the new multi-class architecture G accomplishes

$$\max_{\varepsilon} d_{ij}^{F} - d_{ij}^{G} = 2.$$

Proof. If the translation function $\Theta_{\mathrm{PP}}(s^k)$ is used, an error on θ_i and θ_j at the most changes a positive vote into a negative one, or deletes a positive vote for the correct class and adds a positive vote for the wrong one. In any case, the distance between total pulls is reduced to two votes. □

In order to study the robustness of multi-class architectures based on two-class classifiers when output signs are considered, a robustness parameter is defined.

Definition 5. Let $\mathbf{x} \in \mathscr{X}$ be an entry having a known output, $\theta_m \in \mathscr{Y}$. Let

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

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

The *robustness parameter*

$$\varepsilon_{\mathrm{rob}}(F) = \arg\min_{\mathbf{x}} \varepsilon_{\mathrm{rob}}(\mathbf{x}, F) \quad \forall \mathbf{x} \in \mathscr{X},$$

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

$$\varepsilon_{\mathrm{rob}}^{1} = \min_{F \in \mathbf{A}_1} \varepsilon_{\mathrm{rob}}^{1}(F) > \min_{F \in \mathbf{A}_2} \varepsilon_{\mathrm{rob}}^{2}(F) = \varepsilon_{\mathrm{rob}}^{2}, \tag{12}$$

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.

Proposition 6. *If K is the number of classes in consideration, the multi-class architecture* $1-v-1-v-r$ *SVMC parallel decomposition procedure and mixed precise translation based on a combinatorial reconstruction F has a robustness parameter*

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

Proof. For this architecture, the number of classifiers concerning a certain class θ_r is $L_m = \binom{K}{2}$. Hence, it is only necessary to probe $\# f_r^{\mathrm{err}} = K-2$.

If Proposition 3 and Lemma 4 are applied, then

$$d_{ir}^{F} = |\Psi_r^F - \Psi_r^F| = 2K-3, \quad \forall i \neq r.$$

If any error, ε, occurs, then the new multi-class machine, G, accomplishes

$$\exists i \neq r: \quad d_{ir}^{G} = d_{ir}^{F} - 2,$$

so, even if

$$\#f_r^{\text{err}} = \left\lfloor \frac{2K-3}{2} \right\rfloor K - 2,$$

the final output is correct. □

In a similar way, the following Proposition can be demonstrated.

Proposition 7. *A standard multi-class architecture based on* $1-v-r$ *two-class classifiers decomposition and pulling reconstruction has a robustness parameter*

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

A standard multi-class architecture based on $1-v-1$ *two-class classifiers decomposition and pulling reconstruction has a robustness parameter*

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

A '*pairwise*' *multi-class architecture* [11] *based on* $1-v-1$ *two-class classifiers decomposition and* '*pairwise*' *pulling reconstruction has a robustness parameter*

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

A DAGSVM architecture [17] *has a robustness parameter*

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

7. Experiments

The theoretical robustness property of the new multi-class architecture based on the K-SVCR machine has been demonstrated in the last section and the good performance of the machine for the decomposition scheme has been illustrated by standard artificial experiments. To validate the new global procedure, typical databases from the 'UCI Repository' [5]—'Iris', 'Glass' and 'Wine'—have been employed. This choice has been taken in order to compare the obtained results with those presented in [20], the most usual multi-class scheme.

In the former illustrative experiments, the number of patterns to be evaluated is reduced, so the QP problem has been exactly solved by using the implemented Matlab's routine. However, for these new 'benchmarks', it is necessary to solve the QP problem by any iterative procedure because solving the QP problem is a critical time-consuming point, prohibitive for PC style computers. New very fast QP-solving algorithms are continuously presented; so it is very difficult to compare them all and make the 'best' choice. Some popular algorithms are based on the work treated in [16,10]. An iterative procedure based on *Iterative Re-Weighted Least Square* [14] has been chosen. This algorithm ensures a fast resolution of the QP problem and the final solution, the support vectors, is identical to those obtained by the QP standard procedure.

Table 3
Performance of the new multi-class architecture based on the K-SVCR machine

	#pts	#atr	#clase	$1-v-r$	$1-v-1$	qp-mc-sv	K-SVCR
Iris	150	3	4	1.33	1.33	1.33	[1.93,3.0]
Wine	178	13	3	5.6	5.6	3.6	[2.29,4.29]
Glass	214	9	7	35.2	36.4	35.6	[30.47,36.35]

The percentage of error on the validation set using the new procedure, standard combinations of SVM machines and the multi-class SVM proposed in [20] are compared.

According to the procedure established in [20], each database is randomly partitioned with a tenth of the data reserved to validate the multi-class architecture. This process is repeated 100 times—in the work of reference it is done only 10 times. Data are normalized to zero mean and unity standard deviation, and several kernels are tested for each 'benchmark': a polynomial kernel, with degree 2 and 3 for the 'Iris' and 'Wine' databases, respectively, and a gaussian kernel for the 'Glass' database. Several C, D and insensitivity parameter δ are used. Table 3 summarizes the results obtained by the new procedure and compares them to those presented in [20]. It also displays the number of training patterns, attributes and classes for each database.

The 10 calculated results for the new machine are expressed in an intervalar form. The inferior bound is the occurred error if all the equality situations between two or more classes are well solved, and the superior bound is the case when the reconstruction procedure chooses always the wrong label.

It can be observed that the new procedure improves the obtained results for other multi-class architectures when database classification is more complicated. In detail, on the 'Iris' set the absolute error is small, but relative performance is poor compared to the rest of the machines. However, a similar experiment has been done considering percentages only every 10 iterations and in three cases performance has been similar to the other machines. For the 'Glass' database, the absolute error is low because the data are unbalanced, but the relative performance is similar to the other procedures. For the 'Wine' benchmark, the new machine shows a better behaviour than the other ones.

8. Conclusions

The "Support Vector Classification-Regression" machine for K-class classification purposes (K-SVCR) introduced in this study is a new training algorithm with ternary outputs $\{-1, 0, +1\}$ based on Vapnik's Support Vector theory for multi-class classification purposes. This learning machine improves standard two-class classifiers in the multi-classification structure because all the data are considered while the centre of attention is a two-class partition. Several experiments on artificial data show the ability of the new machine to treat multi-class problems and they illustrate the insensitivity parameter utility. The advantages compared with SVR performing multi-classification tasks are made evident.

In order to export this excellent behaviour on the overall multi-class architecture, a translate function has been specifically designed. A natural combinatorial function on the translations allows to demonstrate the robustness of the new structure. A robust parameter has been defined to demonstrate that the new architecture is more robust than standard ones. In fact, it has been proven that all the other architectures are not robust at all.

Further research on tuning parameters will be made on traditional benchmark data sets to show the performance of the new multi-class classification architecture.

Acknowledgements

The authors would like to thank the reviewers for their valuable comments. This work was partially supported by the Catalan Government within the framework of the CTP program (MC^3S, ITT-2002) and the Spanish Government grant TIC2002-04371-C02-02.

References

[1] E.L. Allwein, R.E. Schapire, Y. Singer, Reducing multiclass to binary: a unifying approach for margin classifiers, Proceedings of the ICML-00, Stanford, CA, 2000.
[2] E. Alpaydin, E. Mayoraz, Combining linear dichomotizers to construct nonlinear polychotomizers, Technical Report IDIAP-RR 5, IDIAP, Switzerland, 1998.
[3] C. Angulo, Learning with Kernel Machines into Multiclassification Frameworks, Automatic Control Department, Technical University of Catalonia, Barcelona, Spain, 2001.
[4] K.P. Bennett, Combining support vector and mathematical programming methods for classification, in: B. Schölkopf, C.J.C. Burges, A.J. Smola (Eds.), Advances in Kernel Methods: Support Vector Learning, MIT Press, Cambridge, MA, 1999.
[5] C.L. Blake, C.J. Merz, UCI Repository of Machine Learning Databases, University of California, Irvine, 1998. http://www.ics.uci.edu/∼mlearn/MLRepository.html.
[6] C.J.C. Burges, A Tutorial on support vector machines for pattern recognition, Data Mining Knowledge Discov. 2 (1998) 1–47.
[7] C. Cortes, V. Vapnik, Support vector networks, Machine Learning 20 (1995) 273–297.
[8] T.G. Dietterich, G. Bakiri, Solving multiclass learning problems via error-correcting output codes, J. Artif. Intell. Res. 2 (1995) 263–286.
[9] Y. Guermeur, A. Elisseeff, H. Paugam-Moisy, A new multi-class SVM based on a uniform convergence result, Proceedings of the IJCNN-00, Como, Italy, 2000.
[10] S.S. Keerthi, S.K. Shevade, C. Bhattacharyya, K.R.K. Murthy, Improvements to Platt's SMO algorithm for SVM classifier design, Technical Report 99-14, National University of Singapore, Singapore, 1998.
[11] U. Kressel, Pairwise classification and support vector machines, in: B. Schölkopf, C.J.C. Burges, A.J. Smola (Eds.), Advances in Kernel Methods: Support Vector Learning, MIT Press, Cambridge, MA, 1999.
[12] E. Mayoraz, E. Alpaydin, Support vector machines for multi-class classification, Proceedings of the IWANN-99, Alicante, Spain, 1999, pp. 833–842.
[13] M. Moreira, E. Mayoraz, Improved pairwise coupling classification with correcting classifiers, in: C. Nédellec, C. Rouveirol (Eds.), Proceedings of the ECML-98, Chemnitz, Germany, 1998, pp. 160–171.
[14] F. Pérez-Cruz, P.L. Alarcón-Diana, A. Navia-Vázquez, A. Artés-Rodríguez, Fast training of support vector classifiers, in: S.A. Solla, T.K. Leen, K.-R. Müller (Eds.), Advances in Neural Information Processing Systems, Vol. 12, MIT Press, Cambridge, MA, 2000.
[15] J. Platt, Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods, in: A.J. Smola, P. Bartlett, B. Schölkopf, D. Schuurmans (Eds.), Advances in Large Margin Classifiers, MIT Press, Cambridge, MA, 1999.

[16] J. Platt, Fast training of support vector machines using sequential minimal optimization, in: B. Schölkopf, C.J.C. Burges, A.J. Smola (Eds.), Advances in Kernel Methods: Support Vector Learning, MIT Press, Cambridge, MA, 1999.
[17] J. Platt, N. Cristianini, J. Shawe-Taylor, Large margin DAGs for multiclass classification, in: S.A. Solla, T.K. Leen, K.-R. Müller (Eds.), Advances in Neural Information Processing Systems, Vol. 12, MIT Press, Cambridge, MA, 2000.
[18] A.J. Smola, Learning with Kernels, Computer Science Department, Technical University Berlin, Berlin, Germany, 1998.
[19] V. Vapnik, The Nature of Statistical Learning Theory, Springer, New York, 1995.
[20] J. Weston, C. Watkins, Multi-class support vector machines, CSD-TR-98-04 Royal Holloway, University of London, Egham, UK, 1998.

Cecilio Angulo received his M.Sc. degree in Mathematics in 1993 from the University of Barcelona and the Ph.D. degree in 2001 from the Technical University of Catalonia. He is currently an Aggregated Professor at the Technical University of Catalonia and a member of the Knowledge Engineering Research Group, GREC, a partner of the European Research Laboratory LEA-SICA. His research interests include machine learning, intelligent control and financial data modelling.

Xavier Parra received his M.E. degree in Computer Engineering in 1992 and the Ph.D. degree in 2001 from the Technical University of Catalonia. He is currently an Associate Professor at the Technical University of Catalonia and a member of the Knowledge Engineering Research Group, GREC, a partner of the European Research Laboratory LEA-SICA. His main research interests include neural net-works, qualitative reasoning and financial data modelling.

Andreu Català received his M.Sc. degree in Physics in 1980 and the Ph.D. degree in 1993 from the Technical University of Catalonia. He is currently an Associate Professor at the Technical University of Catalonia and Director of the Knowledge Engineering Research Group, GREC, a partner of the European Research Laboratory LEA-SICA. His main research interests include neural networks, uncertainty information treatment and sustainability analysis.