Journal of Pattern Recognition Research 1 (2010) 1-9

Received May 26, 2010. Accepted Jul 12, 2010.

A Probabilistic Tri-class Support Vector Machine

Luis Gonzalez-Abril *luisgon@us.es*
Applied Economics I Department
University Seville, Avda Ramon y Cajal, 41018 Seville, Spain

Cecilio Angulo *cangulo@esaii.upc.es*
GREC Research Group
Universitat Politécnica de Catalunya, 08800 Vilanova i la Geltru, Spain

Francisco Velasco *velasco@us.es*
Applied Economics I Department
University Seville, Avda Ramon y Cajal, 41018 Seville, Spain

Juan Antonio Ortega *ortega@lsi.us.es*
Computer Languages and Systems Department
41012 Seville, University Seville, Spain

Abstract

A probabilistic interpretation for the output obtained from a tri-class Support Vector Machine into a multi-classification problem is presented in this paper. Probabilistic outputs are defined when solving a multi-class problem by using an ensemble architecture with tri-class learning machines working in parallel. This architecture enables the definition of an 'interpretation' mapping which works on signed and probabilistic outputs providing more control to the user on the classification problem.

Keywords: Multi-class, Classification, Pairwise training, Kernel.

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 on the basis of a separable binary classification problem with signed outputs ± 1. Roughly speaking, two SVM-based approaches exist to extend binary classification to multi-class classification [19]. The "all the classes at once" approach solves the multi-classification problem by considering all instances from all classes in an unique optimization formulation, whereas the "decomposition-reconstruction" architecture defines an ensemble architecture with learning machines working in parallel. Latter approach is usually preferred since the optimization problem is more manageable. Several architectures [12, 14, 17] have been developed combining parallel SVMs into a multi-classification framework, with binary methods based on one-versus-rest (1-v-r) or pairwise (1-v-1) classes division. The 1-v-1 scheme is usually preferred to the 1-v-r scheme [11] because it takes less training time.

Probabilistic outputs according to the method introduced by Sollich [18] are considered in a multi-classification ensemble architecture with several learning machines working in parallel. The approach taken into consideration for the ℓ-class problem is based on the ℓ-SVCR machine [3] for multi-classification purposes. The ℓ-SVCR machine is especially addressed towards avoiding any loss of information which occurrs in the usual 1-v-1 training, by using a similar two-phase (decomposition, reconstruction) scheme. Furthermore, it is well-known that the comparison of outputs of different SVMs which provide the final

output when bi-classifiers are used in multi-classification problems is deemed an inadequate approach [10, 13, 15]. Therefore, a direct comparison between numeric outputs of different parallel SVMs is avoided by using a probabilistic approach.

The paper is organized as follows: both SVMs and Sollich's approach are briefly introduced in the next section. In Section 3, SVMs are analyzed for multi-class problems when 1-v-1 SVMs are implemented in a two-phase scheme. Sollich's probabilities are generalized for a ℓ-SVCR decomposition and the counterpart reconstruction scheme is determined. The interpretation of the new paradigm is presented in Section 4, and an experimental comparison with other approaches is given in Section 5. Section 6 provides a final discussion and concludes this paper.

2. Probabilities in SVMs

Let $Z = \{(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_n, y_n)\}$ be a training set, with $\{\mathbf{x}_1, \cdots, \mathbf{x}_n\} \subset \mathcal{X}$, and $y_i \in \mathcal{Y} = \{-1, 1\}$ for a binary classification problem. In the general SVM algorithm, inputs $\mathbf{x}$ are firstly mapped onto vectors $\phi(\mathbf{x})$ in some feature space, $\mathcal{F} \subset \mathbb{R}^d$, by a non-linear mapping. Ideally, in the feature space, where an inner product is defined, the problem should be linearly separable and a search procedure is performed in the form of a decision hyperplane $\pi \equiv \omega \cdot \phi(\mathbf{x}) + b = 0$, leading to the SVM optimization problem [6]: to find a vector $\omega \in \mathbb{R}^{d'}$ and a bias $b \in \mathbb{R}$ which minimizes

$$\begin{array}{l} \min\limits_{\omega \in \mathbb{R}^{d'}} \dfrac{1}{2} \|\omega\|^2 + C \sum\limits_{i=1}^{n} \xi_i \\ \text{s.t.} \quad \left\{ \begin{array}{l} y_i \left(\omega \cdot \phi(\mathbf{x}_i) + b\right) - 1 + \xi_i \geq 0, \ \forall i \\ \xi_i \geq 0, \ \forall i \end{array} \right. \end{array} \tag{1}$$

where $\| \ \|$ and $\cdot$ denote the norm and the inner product in $\mathcal{F}$, respectively.

Patterns exactly matching the first set of inequalities verify $\xi_i = 0$, and hence no penalization occurs of the risk function to be minimized. Remaining training vectors do increase the risk function by a quantity $C\,\xi_i = C\,[1 - y_i(\omega \cdot \phi(\mathbf{x}_i) + b)]$ (Karush-Kuhn-Tucker condition)[19]. Hence, a new formulation of the risk function could be considered:

$$\frac{1}{2} \|\omega\|^2 + C \sum_{i=1}^{n} l(y_i(\omega \cdot \phi(\mathbf{x}_i) + b))$$

where $l(z)$ is the 'hinge loss' function: $l(z) = |1 - z|_+$, that is, $l(z) = 1 - z$ if $1 - z \geq 0$, and $l(z) = 0$ otherwise.

From this formulation, a distribution on (X, Y) (considered as a random vector) is derived [18] such that the problem (1) is a maximum likelihood problem. Accordingly, it follows that the probability of y conditioned to $\mathbf{x}$ where $\theta = (\omega, b)$ with $\theta(\mathbf{x}) = \omega \cdot \phi(\mathbf{x}) + b$ is

$$P(y|\theta(\mathbf{x})) = \left\{ \begin{array}{ll} \dfrac{1}{1 + e^{-2Cy\theta(\mathbf{x})}} & \text{if} \quad |\theta(\mathbf{x})| \leq 1 \\ \dfrac{1}{1 + e^{-Cy[\theta(\mathbf{x}) + sign(\theta(\mathbf{x}))]}} & \text{if} \quad |\theta(\mathbf{x})| > 1. \end{array} \right.$$

This generalization is not disturbed by the former considerations: if a new entry $\mathbf{x}$ is $\theta(\mathbf{x}) > 0$ then $P(Y = 1|\theta(\mathbf{x})) > P(Y = -1|\theta(\mathbf{x}))$ and the signed output is $Y = 1$; analogously, for $\theta(\mathbf{x}) < 0$ then the signed output is $Y = -1$.

3. SVMs for Multi-Classification. ℓ-SVCR Machines

A set of possible labels $\{\theta_1, \cdots, \theta_\ell\}$, with $\ell > 2$ is considered. Let $Z = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ be a training set. Subsets $Z_k \in Z$, defined as $Z_k = \{(\mathbf{x}_i, y_i) : y_i = \theta_k\}$ generate a partition in Z, which is denoted as $n_k = \#Z_k$, and hence $n = n_1 + \cdots + n_\ell$. If I_k is the number of index i where $(\mathbf{x}_i, y_i) \in Z_k$, then it follows that $\bigcup_{i \in I_k} \{(\mathbf{x}_i, y_i)\} = Z_k$.

The 1-v-1 SVM is a well-known multi-classification SVM approach which reduces the problem to learning and aggregating preference predictions among the possible labels. In this approach, $L = \frac{\ell \cdot (\ell-1)}{2}$ binary classifiers are trained to generate hyperplanes f_{kh}, $1 \leq k < h \leq \ell$, which separate training vectors Z_k with label θ_k from training vectors in class θ_h, Z_h. If f_{kh} discriminates without error then $sign(f_{kh}(\mathbf{x}_i)) = 1$ for $\mathbf{x}_i \in Z_k$, and $sign(f_{kh}(\mathbf{x}_i)) = -1$ for $\mathbf{x}_i \in Z_h$. Remaining training vectors $Z \setminus \{Z_k \bigcup Z_h\}$ are not considered in the optimization problem. Hence, for a new entry $\mathbf{x}$, the output from the machine $f_{kh}(\mathbf{x})$ is interpreted as:

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

In the reconstruction phase, a scheme is implemented which takes into consideration the label distribution generated by machines in the parallel decomposition $\{(\theta_k, m_k)\}$, where m_k is the number of votes obtained by label θ_k from the machines f_i, $i = 1, \cdots, L$, $\sum_k m_k = L$ and $0 \leq m_k \leq \ell - 1$.

A drawback cited for this approach is that the number of machines to be trained is high in comparison with the 1-v-r approach when ℓ is high. A second problem, considered in the literature, is that data from only two classes is considered in training each machine, and hence, output variance is high, and any information from the remaining classes is ignored and may even be later misinterpreted. Hence, the SVM solution is affected by this loss of training information: if a hyperplane f_{kh} must classify an input $\mathbf{x}_i$ with $i \notin I_k \bigcup I_h$, only output $f_{kh}(\mathbf{x}_i) = 0$ will generate a correct interpretation. One improvement yet to be analyzed is to force every training input from different classes θ_k and θ_h to be contained into the hyperplane $f_{kh}(\mathbf{x}) = 0$. By following this idea, the ℓ-SVCR machine was introduced [3] which is briefly described below.

3.1 ℓ-SVCR Machines

A hyperplane which separate inputs in class θ_1 from class θ_2, in order to simplify notation, is sought. Training vectors are ordered in such a form that the first n_1 vectors belong to class θ_1, followed by the n_2 vectors belonging to class θ_2 and the remaining vectors are from the rest of the classes.

Following the classic SVM approach, the objective is to find a hyperplane $f_{12}(\mathbf{x}) = 0$ which separate classes θ_1 and θ_2. Nevertheless, information in the rest of the classes is now used for the hyperplane construction: $f_{12}(\mathbf{x})$ must allocate entries from class θ_1 into the region $\{\mathbf{x} : f_{12}(\mathbf{x}) \geq 1\}$, entries from class θ_2 must allocate into the region $\{\mathbf{x} : f_{12}(\mathbf{x}) \leq -1\}$, and the remaining vectors must be allocated into a region, depending on a parameter $0 \leq \delta < 1$, $\{\mathbf{x}, : |f_{12}(\mathbf{x})| \leq \delta\}$. Parameter δ allows to a slack zone (a 'tube') to be created around the hyperplane where remaining training vectors are covered.

The most general solution in the form $f_{12}(\mathbf{x}) = \omega \cdot \phi(\mathbf{x}) + b$ of the ℓ-SVCR problem can be obtained if kernel functions are introduced and restrictions are relaxed by using slack variables and, hence, by solving the problem:

$$\min_{\omega \in \mathbb{R}^d} \frac{1}{2} \|\omega\|^2 + C_1 \sum_{i=1}^{n_1+n_2} \xi_i + C_2 \sum_{i=n_1+n_2+1}^{n} (\varphi_i + \varphi_i^*) \tag{2}$$

subject to

$$y_i \left(\omega \cdot \phi(\mathbf{x}_i) + b\right) \geq 1 - \xi_i, \qquad \forall i = 1, 2, \cdots, n_1 + n_2, \tag{3}$$

$$-\delta - \varphi_i^* \leq \omega \cdot \phi(\mathbf{x}_i) + b \leq \delta + \varphi_i, \qquad \forall i = n_1 + n_2 + 1, \cdots, n, \tag{4}$$

$$\begin{array}{rl} \xi_i \geq 0, & \forall i = 1, 2, \cdots, n_1 + n_2, \\ \varphi_i^*, \varphi_i \geq 0, & \forall i = n_1 + n_2 + 1, \cdots, n, \end{array} \tag{5}$$

with $0 \leq \delta < 1$. The new machine assigns a new entry $\mathbf{x}$ to a class in accordance with

$$\Theta(f_{12}(\mathbf{x})) = \begin{cases} \theta_1 & \text{if } f_{12}(\mathbf{x}) > \delta \\ \theta_0 & \text{if } |f_{12}(\mathbf{x})| \leq \delta \\ \theta_2 & \text{if } f_{12}(\mathbf{x}) < -\delta \end{cases} \tag{6}$$

where θ_0 is an artificial label designating a no-label assignment. Furthermore, the solution is presented in the form [3]: $f_{12}(\mathbf{x}) = \sum_i \alpha_i \, k(\mathbf{x}_i, \mathbf{x}) + b$ where α_i are the Lagrange multipliers associated to (2), whereby $\sum_i \alpha_i = 0$ and bias b is obtained from restrictions on the support vectors [7].

3.2 Probabilities in ℓ-SVCR Machines

Let $\theta(\mathbf{x}) = \omega \cdot \mathbf{x} + b$ be a solution of (2) subject to restrictions (3–5), depending on parameters ω and b, with $\omega \in \mathbb{R}^d$ and $b \in \mathbb{R}$. Therefore,

- If vector $\mathbf{x}_i$ is labelled θ_1, then the correct output for the ℓ-SVCR machine is $\theta(\mathbf{x}_i) \geq 1$, because output $y_i = 1$ for the 1-v-1 learning machine $f_{12}(\mathbf{x})$ has been matched with θ_1 in (6). Otherwise, it follows from (3) that $\xi_i = 1 - \theta(\mathbf{x}_i) \geq 0$ is added to the risk function.

- If vector $\mathbf{x}_i$ is labelled θ_2, then a similar study can be developed with $\theta(\mathbf{x}_i) \leq -1$ and $\xi_i = 1 + \theta(\mathbf{x}_i)$.

- If vector $\mathbf{x}_i$ is labelled θ_k with $k \neq \{1, 2\}$ then the correct output for the ℓ-SVCR machine is $|\theta(\mathbf{x}_i)| \leq \delta$, because output $y_i = 0$ has been matched with θ_0. Otherwise, it adds a loss in the risk function $\varphi_i^* = -\theta(\mathbf{x}_i) - \delta$ if $\theta(\mathbf{x}_i) < -\delta$ or $\varphi_i = \theta(\mathbf{x}_i) - \delta$ if $\theta(\mathbf{x}_i) > \delta$.

Following Sollich's approach, when the hinge loss function is used, "probabilities" can be assigned to $y = 1$ and $y = -1$ depending on the new input $\mathbf{x}$, and parameters ω and b:

$$\begin{aligned} Q[y = 1 | \theta(\mathbf{x})] &= exp\left[-C_1 \; l(\theta(\mathbf{x}))\right], \\ Q[y = -1 | \theta(\mathbf{x})] &= exp\left[-C_1 \; l(-\theta(\mathbf{x}))\right]. \end{aligned}$$

Furthermore, by considering the δ-insensitivity function

$$|z|_\delta = \begin{cases} -z-\delta & \text{if} \quad z < -\delta \\ 0 & \text{if} \quad -\delta \leq z \leq \delta \\ z-\delta & \text{if} \quad \delta < z \end{cases}$$

then output $y = 0$ can be assigned with "probability"

$$Q[y=0|\theta(\mathbf{x})] = exp\left[-C_2 \ |\theta(\mathbf{x})|_\delta\right].$$

In order to convert these quantities into effective probabilities, then $v(\theta(\mathbf{x})) = \sum_{y\in\{-1,0,1\}} Q[y|\theta(\mathbf{x})]$ must be considered. Hence, if an adequate distribution is chosen on X, ω and b, the maximum likelihood problem obtained by using probabilities

$$P[Y=i] = P[y=i|\theta(\mathbf{x})] = \frac{1}{v(\theta(\mathbf{x}))} Q[y=i|\theta(\mathbf{x})], \qquad i=-1,0,1,$$

is the same as the ℓ-SVCR problem. An example for these probabilities is displayed in Fig. 1. It can be seen that results on the machine are very intuitive:

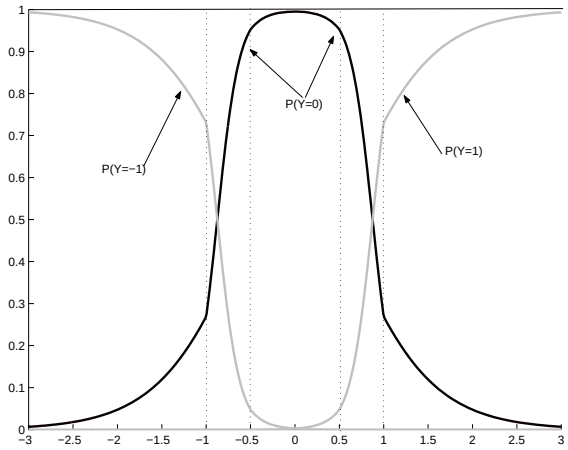

Fig. 1: Probability function for $\delta = 0.5$, $C_1 = 6$ and $C_2 = 2$.

- if $\theta(\mathbf{x}) < -1$, the probability assigning label $y = -1$ is higher than the other two probabilities, and it increases as $\theta(\mathbf{x})$ decreases.
- if $\theta(\mathbf{x}) > 1$, the probability assigning label $y = 1$ is higher than the other two probabilities, and it increases along $\theta(\mathbf{x})$.
- if $-\delta < \theta(\mathbf{x}) < \delta$, the probability assigning label $y = 0$ is higher than the other two probabilities, and it increases the nearer it is to 0.

3.3 Reconstruction Scheme

When probabilities are considered in the models, a new 'interpretation mapping' for ℓ-SVCR outputs, different from (6), is defined:

$$\Theta(f_{12}(\mathbf{x})) = \begin{cases} \theta_1 & \text{if } P[Y=1] > \max\{P[Y=0], P[Y=-1]\} \\ \theta_0 & \text{if } P[Y=0] \geq \max\{P[Y=-1], P[Y=1]\} \\ \theta_2 & \text{if } P[Y=-1] > \max\{P[Y=0], P[Y=1]\}. \end{cases} \tag{7}$$

Furthermore, the reconstruction scheme given in (7) is better than the reconstruction scheme given in (6) in the sense that equalities in the number of votes can be broken by using a mean of probabilities for each class.

It is worth noting that the output scale in a standard SVM is determined such that outputs for the support vectors are ± 1. Hence, output comparison for different SVMs which provide the final output when bi-classifiers are used in multi-classification problems is deemed an inadequate approach [10, 13, 15]. Therefore, a direct comparison between numeric outputs for different parallel SV machines is avoided in our probabilistics approach.

Outputs to be taken into for each implemented ℓ-SVCR are: (i) an assigned label from ℓ-SVCR, and (ii) the probability associated to the labelling. Hence, users have more complete information about outputs from the overall multi-class architecture. This point is illustrated with the example of the four classes given in Table 1 where an equality between two classes,

Table 1: Example of probabilities in ℓ-SVCR Machines.

f_{kh}	1-2	1-3	1-4	2-3	2-4	3-4
Label	θ_1	θ_1	θ_0	θ_2	θ_4	θ_4
Probability	65%	80%	55%	80%	80%	70%

θ_1 and θ_4 can be observed. The machine assigns label θ_4 as the winner since the probability mean of θ_1 (72.5%) is smaller than the probability mean of θ_4 (75.0%). Furthermore, it can be seen that mapping f_{14} introduces an error because the final label output is implied, so an 'a posteriori' study should be considered.

3.4 ℓ-SVCR Parameters

Parameters to be tuned in (2) are: (i) k, the kernel function; (ii) C_1, the associated weight for the sum of errors in the two discriminated classes; (iii) C_2, the associated weight for the sum of errors in the remaining classes; (iv) δ, the insensitivity parameter. As usual, the kernel function is a very relevant choice because it determines the feature space where separation between classes is realized.

The 'interpretation mapping' defined in (7) allows the relationship among C_1, C_2 and δ to be made evident. By using both the definition of the probabilities and the symmetric relationship between regions in (7), the frontier between classes can be evaluated by calculating the value $\delta^* = \theta^*(x)$ such that equation $P[Y = 1/\theta^*(x)] = P[Y = 0/\theta^*(x)]$ is verified, which yields the solution $\theta^*(x) = \delta^* = \frac{C_1 + C_2\,\delta}{C_1 + C_2}$. This solution is a convex combination of the frontiers for the ℓ-SVCR and the SVM standard machines, δ and 1, respectively. If substitution is made, the mapping can be regarded as

$$\Theta(f_{12}(x)) = \begin{cases} \theta_1 & \text{if } \theta(x) > \delta^* \\ \theta_0 & \text{if } |\theta(x)| \leq \delta^* \\ \theta_2 & \text{if } \theta(x) < -\delta^* \end{cases} \tag{8}$$

which is similar to that defined in (6), but with δ^* depending on C_1, C_2 and δ. Hence, as $\delta^* \geq \delta$, it is straightforward to prove that this new mapping is more restrictive than (6) when assigning a label θ_1 or θ_2.

Variations on the frontiers can be studied in this new expression with respect to the parameters. If C_2 and δ are fixed, then increasing C_1 signifies giving more weight to migrations between labels θ_1 and θ_2 because δ^* is approximated towards value 1. Hence, the 'tube' region is wider and the resulting learning machine takes little risk. Similar reasoning can

be carried out if C_1 decreases (δ^* is approximated towards value δ), with a riskier learning machine being generated.

If C_1 and δ are fixed, the increasing C_2 is equivalent to increasing the weight on errors with patterns labelled θ_0, and, therefore, the number of inputs with label θ_1 or θ_2 are increased.

If C_1 and C_2 are fixed, then interpretation of changes in $0 \leq \delta \leq 1$ is the same as in the original configuration problem.

By studying variations on the frontier with respect to joint variations on C_1 and C_2, it can be noted that $\delta^* = 1 - \frac{1-\delta}{1+C_1/C_2}$ and from here, it follows that: if ratio C_1/C_2 increases then the frontier tends towards 1; if ratio C_1/C_2 decreases, then the frontier tends towards δ. As a particular case, if $C_1 = C_2$, then the frontier is the point midway between δ and 1. For an automatic selection of δ and further discussion about this parameter, see [2].

4. An Example on Enterprise Data

A data benchmark problem composed of 474 vectors [16] is considered as an illustrative example. The dimension patterns are grouped into 3 classes with a label dominating the other two labels which implies that the complexity of the classification problem is high.

The labelling distributions of both dataset and training set which is formed by extracting the first 200 vectors, are given in Table 2. Thus, for a random labelling, the probability of

Table 2: Labeling distribution of dataset and training set in the illustrative example.

	dataset			training set		
Label	1	2	3	1	2	3
Number	363	27	84	150	11	69
Percentage	75.68%	5.71%	17.72%	75%	6.5%	19.5%

assigning correct labels is 62.07%. However, if information about label distribution is used, then label "1" can be assigned to any entry $\mathbf{x}$ and the probability of correct output becomes 75.68%. Hence, our overall multi-class machine must improve this baseline percentage. The classification has been developed over normalized data and allocates a higher weight to migrations between outputs "1" and "-1", in f_{ij}, than migrations to or from "0" ($C_1 = 5$ and $C_2 = 3$). In this way, influence from label "1" is reduced. The insensitivity parameter is adjusted to $\delta = 0.1$ and the kernel is a standard Gaussian function with parameter $\sigma = 1$.

Accuracy for the machine evaluated on the training vectors is 95% correct, 5% error and all the training patterns are classified. A low insensitivity parameter $\delta = 0.1$ causes the labelling of all the data, and as a result several errors can be appreciated. Accuracy results obtained on the test vectors are given in Table 3. Overall, the model makes a correct prediction on 247 patterns (90.15%), makes mistakes on 21 (7.66%) and no label is assigned on 6 (2.19%). It can be concluded that SVMs are sensitive to the relative size of the classes, an inherent characteristic on any discriminant analysis.

5. Experimental Results

In this section, experimental results are presented for several datasets (*Iris*, *Wine*, *Glass*, *Vowel*, *Vehicle* and *DNA*) from the UCI Repository of machine learning databases [4].

The results have been obtained by following the experimental framework which was proposed by [9] and was continued in [1], but with some modifications introduced to incorporate the suggestions in [8] and [20]. Hence, training data have been normalized, (that is, mean

Table 3: Results on the test set.

	Prediction label				Percentage		
Label	1	2	3	0	Correct	Error	Unlabelled
1	201	1	7	4	94.37%	03.76%	01.88%
2	6	8	0	2	50.00%	37.50%	12.50%
3	7	0	38	0	84.44%	15.56%	00.00%

zero and standard deviation one), in order to avoid problems with outliers. Test data are normalized accordingly.

The standard 1-v-1 and 1-v-r formulation and the ℓ-SVCR with $C_1 = C_2 = C$ and δ automatically chosen [2] are considered for multi-classification problem. Their performance, (in the form of accuracy rate), has been evaluated on models using the Gaussian kernel. 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: $C = \left[2^4, 2^3, \ldots, 2^{-10}\right]$ and $\sigma^2 = \left[2^{-11}, 2^{-10}, \ldots, 2^1\right]$.

The criteria used to estimate the generalized accuracy is a ten-fold cross-validation (CV) 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 [5]. The best cross-validation mean rate among the several pairs (C, σ^2) is reported in Table 4. where can be observed that similar performance results are obtained by all three approaches, however slight differences can be appreciated.

Table 4: A comparison of the best accuracy rates using the RBF kernel.

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

6. Conclusions

In this paper, a probabilistic version of a multi-class Support Vector Machine is introduced. Multi-classification problems are analyzed by this machine which is also able to provide the user with guidelines for the labelling process. This new procedure, generated by using probabilities, is more complete and reliable than the standard approach and the accuracy rates is improved in some cases.

The δ-insensitivity zone generated for 'no-labelling' allows all the difficult labelling patterns to be covered. In this way, the patterns without any assigned label can be controlled by the δ parameter and the user can specify the level of risk that the machine takes in the labelling process.

Acknowledgements

This work has been partly supported by the project TIN2009-14378-C02-01 from the Spanish Ministry of Science and Technology.

References

[1] D. Anguita, S. Ridella, and D. Sterpi. A new method for multiclass support vector machines. In *Proceedings of the IEEE IJCNN2004*, 2004.

[2] C. Angulo and L. González. 1-v-1 Tri-Class SV Machine. In *Proceedings of the 11th European Symposium on Artificial Neural Networks, ESANN*, pages 355–360, 2003.

[3] C. Angulo, X. Parra, and A. Català. K-svcr. a support vector machine for multi-class classification. *Neurocomputing*, 55(1-2):57–77, 2003.

[4] C.L. Blake and C.J. Merz. UCI repository of machine learning databases, 1998.

[5] S. Canu, Y. Grandvalet, V. Guigue, and A. Rakotomamonjy. Svm and kernel methods matlab toolbox. Perception Systmes et Information, INSA de Rouen, Rouen, France, 2005.

[6] Luis González, Cecilio Angulo, Francisco Velasco, and Andreu Català. Dual unification of bi-class support vector machine formulations. *Pattern Recognition*, 39(7):1325–1332, 2006.

[7] Luis Gonzalez-Abril, Cecilio Angulo, Francisco Velasco, and J.A. Ortega. A note on the bias in SVMs for multi-classification. *IEEE Transactions on Neural Networks*, 19(4):723–725, January 2008.

[8] Chih-Wei Hsu, C.-C. Chang, and Chih-Jen Lin. A practical guide to support vector classification. Technical report, Department of Computer Science and Information Engineering, National Taiwan University, 2003.

[9] C.W. Hsu and C.J. Lin. A comparison of methods for multiclass support vector machine. *IEEE Transactions on Neural Networks*, 13(2):415–425, 2002.

[10] S. S. Keerthi, S. K. Shevade, and C. Bhattacharyya. A fast iterative nearest point algorithm for support vector machine classifier design. *IEEE Transactions on Neural Networks*, 11:124–136, 2000.

[11] U. Kressel. Pairwise classification and support vector machine. *In B. Schólkopf, C. Burgues and A. Smola, editors*, Advances in Kernel Methods: support Vector Learning. MIT Press. Cambridge, MA:255–268, 1999.

[12] J.T.-Y. Kwok. Moderating the outputs of support vector machine classifiers. *IEEE Trans. on Neural Networks*, 10(5):1018–1031, 1999.

[13] Y. Liu and Y. F. Zhang. One-against-all multi-class svm classification using reliability measures. In *Proceedings of the IJCNN '05*, 2005.

[14] A. Madevska-Bogdanova, D. Nikolik, and L. Curfs. Probabilistic svm outputs for pattern recognition using analytical geometry. *Neurocomputing*, 62:293–303, 2004.

[15] Eddy Mayoraz and Ethem Alpaydin. Support vector machines for multi-class classification. In *IWANN (2)*, pages 833–842, 1999.

[16] C. Pérez. *Técnicas Estadísticas con SPSS*. Prentice Hall, 2001.

[17] J. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *In A. Smola, P. Bartlett, B. Scholkopf, and D. Schuurmans, editors, Advances in Large Margin Classifiers, 1999.*, Advances in Kernel Methods: support Vector Learning. MIT Press. Cambridge, MA, 1999.

[18] P. Sollich. Bayesian methods for support vector machines: Evidence and predictive class probabilities. Kluwer Academic Publidhers, 2000.

[19] V.N. Vapnik. *Statistical Learning Theory*. John Wiley & Sons, Inc, 1998.

[20] Jean-Philippe Vert, Koji Tsuda, and Bernhard Schölkopf. *Kernel Methods in Computational Biology*, chapter A Primer on Kernel Methods, pages 35–70. The MIT Press, 2004.