Probabilistic outputs for a new multi-class Support Vector Machine

Luis González[1] and Cecilio Angulo[2]

[1] Universidad de Sevilla, Applied Economics Dept.,
41018 Sevilla, Spain,
luisgon@us.es
[2] Universitat Politècnica de Catalunya, GREC Research Group,
08800 Vilanova i la Geltrú, Spain,
cangulo@esaii.upc.es

Abstract Support Vector Machines are learning machines originally developed on the basis of a binary classification problem with signed outputs ± 1. The aim of this work is to give a probabilistic interpretation to the numerical output values into a multi-classification learning problem framework. For this purpose, a recent SV Machine, called ℓ-SVCR, addressed to avoid the lose of information occurred in the usual 1-v-1 training is implemented. On this structure, a certain class of probabilistic outputs are considered in an ensemble architecture with learning machines working in parallel. New architecture allows to define a 'interpretation mapping' working on signed and probabilistic outputs giving more control to the user on the classification problem.

1 Introduction

Support Vector Machines are learning machines implementing 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 [Vap98]. Standard SVMs outputs have not a probabilistic interpretation, in the sense to estimate the conditional probability $P[Y|X = \mathbf{x}]$ to quantify uncertainty associated to a prediction. From different perspectives [Kwo99], [MA99], [Pla99], [Sol00], several probabilistic approaches have been developed to set the 'tunable' parameters of the SVM algorithm.

In this work, probabilistic outputs, according the method explained in [Sol00], are considered in a multi-classification ensemble architecture with several learning machines working in parallel. The approach took in consideration for the ℓ-class problem is based on a new SV Machine [Ang01], called ℓ-SVCR, introduced for multi-classification purposes. The ℓ-SVCR machine is specially addressed to avoid the lose of information occurred in the usual 1-v-1 training, by using a similar two-phases (decomposition, reconstruction) scheme.

The paper is organized as follows: Sollich's approach is shortly 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-phases scheme. Drawbacks from this standard approach leads to the definition of a new SV machine specifically designed for multi-classification problems, the ℓ-SVCR machine. Sollich's probabilities are defined for a ℓ-SVCRs decomposition and the counterpart reconstruction scheme is determined. Performance of the new paradigm is evaluated on a benchmark problem. Finally, some concluding remarks are presented.

2 Probabilities in SVMs

A interesting probabilistic method have been elaborated by Peter Sollich in [Sol00] to be applied on standards SVMs. This algorithm and standard SV machines are briefly introduced: let $Z = \{(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_n, y_n)\}$ be a training set, with $\{\mathbf{x}_1, \cdots, \mathbf{x}_n\} \subset \mathcal{X} \subset \mathbb{R}^d$, 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, $\mathcal{F}$, by a non linear mapping. Ideally, in the feature space, where a 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 in the form: to find a parameter vector $\omega \in \mathbb{R}^{d'}$ and a bias $b \in \mathbb{R}$ minimizing

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

Patterns $(\mathbf{x}_i, y_i)$ in the training set being $y_i(\omega \cdot \phi(\mathbf{x}_i) + b) \geq 1$ verify $\xi_i = 0$, so risk function in (1) is not penalized by them. Remaining training vectors do increase risk function in a quantity

$$C\,\xi_i = C\left[1 - y_i(\omega \cdot \phi(\mathbf{x}_i) + b)\right]$$

because $\alpha_i \neq 0$ and $y_i(\omega \cdot \phi(\mathbf{x}_i) + b) - 1 + \xi_i = 0$ (Karush-Kuhn-Tucker condition).
A new formulation of the risk function can be considered:

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

where $l(z)$ is the 'hinge loss' function:

$$l(z) = |1 - z|_+ \tag{3}$$

From this formulation, in [Sol00] is derived a distribution on (X, Y), in such a form that problem (1) is the same as maximum likelihood problem. Accordingly, it follows that probability of y conditioned to $\mathbf{x}$ and $\theta = (\omega, b)$, with $\theta(\mathbf{x}) = \omega \cdot \phi(\mathbf{x}) + b$, is

$$P(y|\mathbf{x}, \theta) = \begin{cases} \dfrac{1}{1 + e^{-2Cy\theta(\mathbf{x})}} & \text{if} \quad |\theta(\mathbf{x})| \leq 1 \\ \dfrac{1}{1 + e^{-Cy[\theta(\mathbf{x}) + sign(\theta(\mathbf{x}))]}} & \text{if} \quad |\theta(\mathbf{x})| > 1. \end{cases} \tag{4}$$

Generalization process 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 output machine is $Y = 1$; analogously, if $\theta^*(\mathbf{x}) < 0$ then output is $Y = -1$. Moreover, if probabilistic outputs are considered in a multi-classification ensemble architecture with several learning machines working in parallel, outputs can be separately interpreted and they can be compared among them because probabilities introduce output normalization.

3 SVMs for Multi-Classification

A set of possible labels $\{\theta_1, \cdots, \theta_\ell\}$, with $\ell > 2$ will be 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, i.e. $Z = \bigcup_{k=1}^{\ell} Z_k$ and $Z_k \bigcap Z_h = \emptyset$, $\forall k \neq h$. It will be denoted $n_k = \#Z_k$, so $n = n_1 + n_2 + \cdots + n_\ell$. If I_k is the number of index i being $(\mathbf{x}_i, y_i) \in Z_k$, it follows $\bigcup_{i \in I_k} \{(\mathbf{x}_i, y_i)\} = Z_k$.
A very usual multi-classification SVM approach is 1-v-1 SVMs: a decomposition phase generates several learning machines in parallel, having in consideration only two classes, and a reconstruction scheme allows to obtain the overall output by merging outputs from the decomposition phase.
In the next, main features of these 1-v-1 machines will be displayed. Improvements will be obtained by incorporating ℓ-SVCR ternary machines with a probabilistic interpretation.

3.1 1-v-1 SV Machines

In this approach, $L = \frac{\ell \cdot (\ell - 1)}{2}$ binary classifiers are trained to generate hyperplanes f_{kh}, $1 \leq k < h \leq \ell$, separating 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 $\mathbf{x}$, numeric 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, some pulling scheme is implemented having in consideration labels distribution generated by machines in the parallel decomposition

Labels	θ_1	...	θ_k	...	θ_ℓ	$\mathcal{Y}$
Votes	m_1	...	m_k	...	m_ℓ	$\frac{\ell \cdot (\ell - 1)}{2}$

where m_k is the number of votes obtained by label θ_k from the machines f_i, $i = 1, \cdots, \frac{\ell \cdot (\ell-1)}{2}$.
The 1-v-1 multi-classification approach is characterized by: (i) $\frac{\ell \cdot (\ell-1)}{2}$ SVMs must be trained on a reduced training set, and (ii) this procedure is usually preferred to the 1-v-r scheme [Kre99]. Main drawbacks for this approach are: (i) only data from two classes is considered for training each machine, so output variance is high and any information from the rest of classes is ignored, and (ii) the number of machines to be trained is high in comparison with the 1-v-r approach when ℓ is high.
SVM solution is affected by this lose of training information because only two classes are considered in each machine. Hence, if a hyperplane f_{kh} must classify a input $\mathbf{x}_i$ with $i \notin I_k \bigcup I_h$, only output $f_{kh}(\mathbf{x}_i) = 0$ will do not generate a incorrect interpretation. The first improvement to be analyzed is to force every training input in different classes to θ_k and θ_h to be contained into the hyperplane $f_{kh}(\mathbf{x}) = 0$.

4 ℓ-SVCR Machines for Multi-Classification

In [Ang01] a new SV Machine is introduced into a similar two-phases scheme (decomposition, reconstruction) for multi-classification, called ℓ-SVCR, addressed to avoid the lose of information in the 1-v-1 training.
In order to simplify notation, let suppose we are looking for a hyperplane separating inputs in class θ_1 from class θ_2. Training vectors are ordered in such a form that the first n_1 vectors belong to class θ_1, next n_2 vectors belong to class θ_2 and remaining $n - n_1 - n_2$ vectors are from the rest of the classes, $\{\theta_3, \cdots, \theta_\ell\}$.
Following the classical SV approach, the objective is looking for a hyperplane $f_{12}(\mathbf{x}) = 0$ separating classes θ_1 and θ_2. Nevertheless, information into the rest of the classes will be now used for the hyperplane construction: $f_{12}(\mathbf{x})$ must allocate entries from class θ_1 in the region $\left\{\mathbf{x} \in \mathbb{R}^d : f_{12}(\mathbf{x}) \geq 1\right\}$, entries from class θ_2 must be similarly allocated in the region $\left\{\mathbf{x} \in \mathbb{R}^d : f_{12}(\mathbf{x}) \leq -1\right\}$, and remaining vectors must be allocated into a region, depending on a parameter $0 \leq \delta < 1$, $\left\{\mathbf{x} \in \mathbb{R}^d, : |f_{12}(\mathbf{x})| \leq \delta\right\}$. Parameter δ allows to create a slack zone (a 'tube') around the hyperplane where remaining training vectors are covered.
If a hyperplane solution exists in the form $f_{12}(\mathbf{x}) = \omega \cdot \mathbf{x} + b$, then it is possible to solve the following ℓ-SVCR problem in exact form:

$$\min_{\omega \in \mathbb{R}^d} \frac{1}{2} \|\omega\|^2 \tag{5}$$

subject to

$$y_i \left(\omega \cdot \mathbf{x}_i + b\right) - 1 \geq 0, \ \forall i = 1, 2, \cdots, n_1 + n_2, \tag{6}$$

$$-\delta \leq \omega \cdot \mathbf{x}_i + b \leq \delta, \ \forall i = n_1 + n_2 + 1, \cdots, n, \tag{7}$$

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

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

where θ_0 is a artificial label designating a no-label assignment.
Usually, no solution exists for the this problem in the original space. A more general solution can be obtained if kernel functions are introduced and restrictions (6–7) are relaxed by using slack variables. A solution hyperplane in the form $f_{12}(\mathbf{x}) = \omega \cdot \phi(\mathbf{x}) + b$ must solve the ℓ-SVCR problem:

$$\min_{\omega \in \mathbb{R}^d} \frac{1}{2} \|\omega\|^2 + C_1 \cdot \sum_{i=1}^{n_1+n_2} \xi_i + C_2 \cdot \sum_{i=n_1+n_2+1}^{n} (\varphi_i + \varphi_i^*) \tag{9}$$

subject to

$$y_i \left(\omega \cdot \phi(\mathbf{x}_i) + b\right) \geq 1 - \xi_i, \ \ \forall i = 1, 2, \cdots, n_1 + n_2, \tag{10}$$

$$-\delta - \varphi_i^* \leq \omega \cdot \phi(\mathbf{x}_i) + b \leq \delta + \varphi_i, \ \ \forall i = n_1 + n_2 + 1, \cdots, n, \tag{11}$$

$$\begin{array}{r} \xi_i \geq 0, \ \ \forall i = 1, 2, \cdots, n_1 + n_2, \\ \varphi_i^*, \varphi_i \geq 0, \ \ \forall i = n_1 + n_2 + 1, \cdots, n. \end{array} \tag{12}$$

A solution to this problem is presented in [Ang01] in the form:

$$f_{12}(\mathbf{x}) = \sum_{i=1}^{N_{sv}} \alpha_i \, k(\mathbf{x}_i, \mathbf{x}) + b \tag{13}$$

where α_i are the Lagrange multipliers associated to (9) accomplishing

$$\sum_{i=1}^{N_{sv}} \alpha_i = 0$$

and bias b is obtained from restrictions on the support vectors.
Parameters to be tuned in the ℓ-SV optimization problem are: (i) k, kernel function; (ii) C_1, associated weight for the sum of errors into the two discriminated classes; (iii) C_2, associated weight for the sum of errors into the remaining classes; (iv) δ, insensitivity parameter. Kernel function is a very relevant choice because it determines the feature space where separation between classes will be realized. A high dimension of this space is necessary because all the training vectors labeled with θ_k, $k = 3, \cdots, \ell$, must be covered by a small 'tube'. However, it has been empirically demonstrated that restrictions associated to a ℓ-SVCR optimization problem are less powerful than those associated to a SVM for regression problems [Ang01].
Parameters C_1 and C_2 are the tradeoff between fitness and smoothing of the solution. To obtain efficient rules determining adequate values is a topic of current research.
Insensitivity parameter δ must remain in the range $[0, 1]$ to avoid decision regions overlapping. If δ decreases, generalization capability of the learning machine decreases on patterns labeled θ_0 and the number of support vectors increases. This parameter is similar to that used in the ε-insensitivity Vapnik's function for SV machines for regression problems.

4.1 Probabilities in ℓ-SVCR Machines

Problem (9) subject to restrictions (10–12) is considered to be solved. Let $\theta(\mathbf{x}) = \omega \cdot \mathbf{x} + b$ be a possible solution, depending on parameters ω and b, with $\omega \in \mathbb{R}^d$ and $b \in \mathbb{R}$. It follows,

- If vector $\mathbf{x}_i$ is labeled θ_1, then 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 (8). Otherwise, it follows from (10) that $\xi_i = 1 - \theta(\mathbf{x}_i) \geq 0$ is added to the risk function.
- If vector $\mathbf{x}_i$ is labeled 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 labeled θ_k with $k \neq \{1, 2\}$ then 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 $\theta(\mathbf{x}_i) > \delta$.

When the hinge loss function is used, according [Sol00], "probabilities" can be assigned to $y = 1$ and $y = -1$ depending on the new input $\mathbf{x}$, and parameters ω and b:

$$Q[y = 1|\theta(\mathbf{x})] = \kappa(C_1, C_2)\, exp\left[-C_1\ l(\theta(\mathbf{x}))\right],$$
$$Q[y = -1|\theta(\mathbf{x})] = \kappa(C_1, C_2)\, exp\left[-C_1\ l(-\theta(\mathbf{x}))\right],$$

with $\kappa(C_1, C_2)$ to be determined.
By considering the δ-insensitivity function

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

then output $y = 0$ from the ℓ-SVCR machine can be assigned with "probability"

$$Q[y = 0|\theta(\mathbf{x})] = \kappa(C_1, C_2)\, exp\left[-C_2\ \ |\theta(\mathbf{x})|_\delta\right].$$

In order to convert these quantities in effective probabilities, it will be defined $\kappa(C_1, C_2)$ as inverse of

$$v(\theta(\mathbf{x})) = \sum_{y \in \{-1,0,1\}} Q[y|\theta(\mathbf{x})].$$

If an adequate distribution is chosen on X, ω and b, the maximum likelihood problem obtained by using probabilities

$$P[y = 1|\theta(\mathbf{x})] = exp\left[-C_1\ l(\theta(\mathbf{x}))\right]/v(\theta(\mathbf{x})),$$
$$P[y = -1|\theta(\mathbf{x})] = exp\left[-C_1\ l(-\theta(\mathbf{x}))\right]/v(\theta(\mathbf{x})),$$
$$P[y = 0|\theta(\mathbf{x})] = exp\left[-C_2\ \ |\theta(\mathbf{x})|_\delta\right]/v(\theta(\mathbf{x}))$$

is the same as ℓ-SVCR problem.
In Fig. 1 is displayed a example for these probabilities. Results on the machine are very intuitive:

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

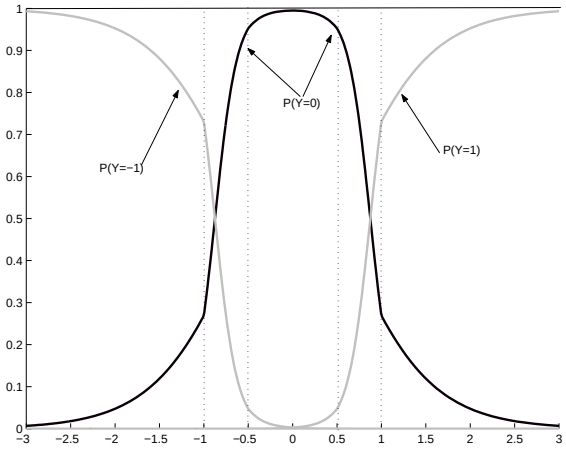

Figure1. Probability function for $\delta = 0.5$, $C_1 = 6$ and $C_2 = 2$.

4.2 Reconstruction Scheme

When probabilities are considered into the models, a new 'interpretation mapping' for ℓ-SVCR outputs, different from (8), can be defined:

$$\Theta(f_{12}(\mathbf{x})) = \begin{cases} \theta_1 \text{ if } P[Y=1] > \max\{P[Y=0], P[Y=-1]\} \\ \theta_0 \text{ if } P[Y=0] \geq \max\{P[Y=-1], P[Y=1]\} \\ \theta_2 \text{ if } P[Y=-1] > \max\{P[Y=0], P[Y=1]\}. \end{cases} \quad (14)$$

New mapping is more restrictive than (8) in order to assign a label θ_1 or θ_2. Definition (14) improves (8): if equalities in the number of votes there exist then label can be assigned by using a mean of probabilities for each class. A direct comparison between numeric outputs for different parallel SV machines is avoided.
Outputs in consideration for each implemented ℓ-SVCR are: (i) assigned label from ℓ-SVCR, and (ii) associated probability to the labeling. So, user have a more complete information about outputs from the overall multi-class architecture.
For illustration, a 4-classes problem is solved by applying a decomposition and reconstruction 4-SV parallel formulation. Outputs for a certain input $\mathbf{x}$ are

f_{kh}	1-2	1-3	1-4	2-3	2-4	3-4
Label	θ_1	θ_0	θ_4	θ_0	θ_4	θ_0
Probability	65%	80%	70%	80%	80%	63%

In this case, not equality is met and overall architecture output for the input $\mathbf{x}$ is labeled θ_4 with probability 75%, the mean of f_{14} (70%) and

f_{24} (80%). User observes than classifier f_{12} is wrong, assigning label θ_1. Mapping f_{34} introduce a worst error because final label output is implied, so a 'a posteriori' study should be considered.
If the pulling would be

f_{kh}	1-2	1-3	1-4	2-3	2-4	3-4
Label	θ_1	θ_1	θ_4	θ_0	θ_4	θ_0
Probability	65%	80%	70%	80%	80%	63%

then overall output would assign label θ_4 with probability 75%. In this case, a equality between two class, θ_1 and θ_4, is met and winner is selected by using probabilities. Moreover, the machine considering both classes assigns label θ_4 as winner, and maybe this information should have a higher weight in the final solution.

4.3 ℓ-SVCR Parameters

In our approach, three parameters, C_1, C_2 and δ, must be selected before the ℓ-SVCR learning machine is trained. The 'interpretation mapping' defined in (14) allows to make evident its relation. By using probabilities definition and symmetric relation between regions in (14), frontier between classes can be evaluated by calculating the value $\delta^* = \theta^*(x)$ such that the equation $P[Y = 1/\theta^*(x)] = P[Y = 0/\theta^*(x)]$ is verified, having solution

$$\theta^*(x) = \delta^* = \frac{C_1 + C_2\,\delta}{C_1 + C_2}$$

This solution can be interpreted like a weighted arithmetic mean of the frontiers for the ℓ-SVCR and the SVM standard machines, δ and 1, respectively, weights being C_2 and C_1.
If substitution is made, mapping can be regarded as

$$\Theta(f_{12}(x)) = \begin{cases} \theta_1 \text{ if } \theta(x) > \delta^* \\ \theta_0 \text{ if } |\theta(x)| \leq \delta^* \\ \theta_2 \text{ if } \theta(x) < -\delta^* \end{cases} \tag{15}$$

similar to that defined in (8), with δ^* depending on C_1, C_2 and δ.
Variations on the frontiers can be studied in this new expression with respect to the parameters. If C_2 and δ are fixed, increasing C_1 signifies to give more weight to migrations from/to labels θ_1 to/from θ_2. Frontier level is approximated towards value 1, hence 'tube' region is wider and resulting learning machine takes little risk. A similar reasoning can be done if C_1 decreases, with a more risked learning machine being generated.
If C_1 and δ are fixed, increasing C_2 is equivalent to increase the weight on errors with patterns labeled θ_0 and the number of inputs with label θ_1 or θ_2 are increased.
If C_1 and C_2 are fixed, interpretation on changes in $0 \leq \delta \leq 1$ is the same as in the original configuration problem.

Studying variations on the frontier with respect to joint variations on C_1 and C_2, it is 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 frontier tends towards 1; if ratio C_1/C_2 decrease, then frontier tends towards δ. As a particular case, if $C_1 = C_2$, then frontier is the middle point between δ and 1.

5 A Example on Enterprise Data

A benchmark problem on data called *empresa* is composed by 474 vectors, took from [Pér01]. 2-dimension patterns are grouped into 3 classes according a certain professional category. Labeling in the examples are ordered, so problem could be treated as 'ordinal regression problem', but it have not been considered in this study. A ℓ-SVCR ordinal regression approach ([AC01]) with probabilities is a future improvement to be done. Choice of this data has been motivated by its complexity, because there exits a label dominating the other two labels. Labeling distribution is

Label	1	2	3
Number	363	27	84
Percentage	75.68%	5.71%	17.72%

If a random labeling is assigned, probability to assign correct labels is

$$A = \{\text{Correct output}\}\,;$$

$$P(A) = \left(\frac{363}{474}\right)^2 + \left(\frac{25}{474}\right)^2 + \left(\frac{84}{474}\right)^2 = \frac{139450}{224676} = 0.6207$$

i.e., 62.07%. However, if information about label distribution is used, it could be considered to assign label "1" for any entry $\mathbf{x}$ and probability of correct output is 75.68%. Our overall multi-class machine must improve this percentage.
A training set is formed by extracting the first 200 vectors, being its label distribution:

Label	1	2	3
Number	150	11	69
Percentage	75%	6.5%	19.5%

similar to the overall set.
Classification has been developed over normalized data giving 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 form, influence from label "1" is reduced. Insensitivity parameter is adjusted to $\delta = 0.1$,

avoiding no-classification regions, and kernel is a gaussian function with parameter $\sigma = 1$.
Mean number of support vectors for the three learned machines is 48 (over 200), and labeling on the training set gives as a result

```
200     1     2     3     0
  1   139     1    10     0
  2     7     4     0     0
  3     4     0    35     0
```

Outputs are 89% correct, 11% error and all the training patterns were classified. A so low insensitivity parameter $\delta = 0.1$ has enforced to label all the data, however as a result several errors can be appreciated. In results matrix can be observed 4 "3"-labeled patterns introduced into class "1", the widest zone.
Results of the machine evaluated on the remaining vectors in *empresa* are

```
274     1     2     3     0
  1   196     3    10     4
  2    12     2     0     2
  3     7     0    38     0
```

In global, model makes a correct prediction on 236 patterns (86.13%), makes mistake on 32 (9.85%) and no label is assigned on 6 (2.19%). If each class is separately analyzed

Label	Correct	Error	No labeled
1	92.02%	06.10%	01.88%
2	12.50%	75.00%	12.50%
3	84.44%	15.56%	00.00%

it can be concluded than SVMs are sensible to the relative size of the classes, an inherent characteristic on any discriminant analysis.

6 Conclusions

In this paper, we introduced a new multi-class Support Vector Machine with probabilistic outputs. Multi-classification problems are analyzed by this machine and an ensemble of rules are provided to the user in the labeling process. New procedure is more complete and reliable than standard approaches.
The novelty δ-insensitivity zone generated for 'no-labeling' allows to cover all the difficult labeling patterns. In this form, rate of patterns without assigned label can be controlled by the δ parameter and the user force the machine to take more or less risk in the labeling process, like it has been empirically proved in [Gon02].

References

[AC01] C. Angulo and A. Català. Ordinal regression with k-svcr machines. *Proc. of the IWAN*, 2001.

[Ang01] C. Angulo. *Aprendizaje con máquinas núcleos en entornos de multiclasificación.* Tesis doctoral, Universidad Politécnica de Cataluña, Abril 2001.

[Gon02] L. González. *Análisis discriminante utilizando máquinas núcleos de vectores soporte. Función núcleo similitud.* Tesis doctoral, Universidad de Sevilla, Marzo 2002.

[Kre99] U. Kressel. Pairwise classification and support vector machine. *In B. Schölkopf, C. Burgues and A. Smola, editors*, Advances in Kernel Methods: support Vector Learning. MIT Press. Cambridge, MA, 1999.

[Kwo99] J.T.-Y. Kwok. Moderating the outputs of support vector machine classifiers. *IEEE Trans. on Neural Networks*, 10(5):1018–1031, 1999.

[MA99] E. Mayoraz and E. Alpaydin. Support vector machines for multiclass clasification. *Proc. of the IWANN*, 1999.

[Pla99] 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.

[Pér01] C. Pérez. *Técnicas Estadísticas con SPSS.* Prentice Hall, 2001.

[Sol00] P. Sollich. Bayesian methods for support vector machines: Evidence and predictive class probabilities. Kluwer Academic Publidhers, 2000.

[Vap98] V.N. Vapnik. *Statistical Learning Theory.* John Wiley & Sons, Inc, 1998.