MULTI-CLASS SUPPORT VECTOR MACHINES: A NEW APPROACH

Jerónimo Arenas-García and Fernando Pérez-Cruz

Department of Signal Theory and Communications, Universidad Carlos III de Madrid,
Avda. de la Universidad 30, 28911 Leganés-Madrid, Spain.
e-mail: {jarenas,fernando}@tsc.uc3m.es

ABSTRACT

In this paper, we propose a new approach for solving multi-class problems with Support Vector Machines. We modify the existing technique to properly reduce the empirical error, therefore we will be ideally able to outperform the previously proposed scheme for multi-class SVMs. The proposed approach also provides solutions with a significant reduction in the number of support vectors, which is an important feature for fast systems.

1. INTRODUCTION

Support Vector Machines (SVMs) have become, in a very short period of time, the standard state-of-the-art tool to solve linear and non-linear knowledge discovery problems [1], specifically for binary classification and regression estimation. The SVM has also been extended to multi-class problems [2, 3]. The SVM extension for multi-class is far from being unique and none of the approaches seems to be superior to the others [4]. Basically, we can distinguish between 2 different trends. The first divides the multiple class problem into a number of binary classification. The generalization step is based on a voting among the binary classifiers to derive the winning class. There are different transformations into binary problems [3, 4], being the most widely used: one-vs.-all, in which each class is compared with all the other classes considered as one [5]; and one-vs.-one, in which each class is individually compared with all the others [3]. The limitation of these approaches is that they do not consider the full problem directly. Particularly, the one-vs.-all approach unbalances the training sets (if the classes are balanced, the negative class in each binary classifier will have far more samples than the positive class), and the one-vs.-one will be using only information from two classes, losing each classifier the information from all the remaining classes.

The second trend considers the multi-class problem directly as a generalization of the binary classification scheme [2] and [1] (Chapter 10). This formulation is very promising because it deals with all the samples and classes at the same time, without losing any relevant information for arriving to the best solution for each problem. Besides, the resulting machines need a lower number of support vectors [4] and achieve higher performances in the case where the training set is separable. If not, the incorrectly classified samples can be multiple times penalized, as we will show in the next section, leading to solutions that are biased towards these samples. We will present in this paper a novel approach to solve the multi-class problem with a unique formulation, in which we eliminate the bias introduced by the previous multiple class SVM formulation.

The rest of the paper is outlined as follows. Section 2 presents the new multi-class setting for SVMs. The optimization procedure is detailed in Section 3. Section 4 shows by means of experiments the behavior of the multi-class SVM. Section 5 ends the paper with some conclusions.

2. MULTI-CLASS SVM

We briefly introduce the formulation for binary classification SVM. Given a training data set $((\mathbf{x}_i, y_i)$, for $i = 1, \ldots, n$ and $\mathbf{x}_i \in \mathbb{R}^d$ and $y_i \in \{\pm 1\})$ and a non-linear transformation to a higher dimensional space, the feature space ($\boldsymbol{\phi}(\cdot)$, $\mathbb{R}^d \xrightarrow{\boldsymbol{\phi}(\cdot)} \mathbb{R}^H$), the SVM solves:

$$\min_{\mathbf{w},\xi_i,b} \left\{ \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_i \xi_i \right\} \tag{1}$$

subject to

$$y_i(\boldsymbol{\phi}^T(\mathbf{x}_i)\mathbf{w} + b) \geq 1 - \xi_i \qquad \forall i = 1, \ldots, n \tag{2}$$

$$\xi_i \geq 0 \qquad \forall i = 1, \ldots, n \tag{3}$$

where $\mathbf{w}$ and b define the linear classifier in the feature space (might be non-linear in the input space). The SVM tries to enforce that the positive samples present an output greater than +1, and the negative samples present an output less than -1. Those samples not fulfilling this condition need a nonzero ξ_i in (2) and therefore they will introduce a penalty in the objective functional (1). The inclusion of the norm of $\mathbf{w}$ in (1) ensures that the solution is maximum margin [1].

The main difference between binary classification and multiple class problem is that $y_i \in \{1, 2, \ldots, k\}$, instead.

Therefore, we can generalize the binary SVM to multi-class by using a different weight vector and bias for each class ($\mathbf{w}^j$ and b^j for $j \in \{1, 2, \ldots, k\}$). So, this classifier computes k outputs to classify any pattern. The classification function is then:

$$f(\mathbf{x}) = \underset{j \in \{1,\ldots,k\}}{\arg\max} \left(\boldsymbol{\phi}^T(\mathbf{x})\mathbf{w}^j + b^j \right) \quad (4)$$

Accordingly, we can impose (2) for each train pattern and each class it does not belong to, as proposed in [1, 2], leading to:

$$\min_{\mathbf{w}^j, b^j, \xi_i^{j,m}} \frac{1}{2} \sum_{j=1}^{k} \|\mathbf{w}^j\|^2 + C \sum_{j=1}^{k} \sum_{\substack{m=1 \\ m \neq j}}^{k} \sum_{i=1}^{n^j} \xi_i^{j,m} \quad (5)$$

subject to

$$(\boldsymbol{\phi}^T(\mathbf{x}_i^j)\mathbf{w}^j + b^j) - (\boldsymbol{\phi}^T(\mathbf{x}_i^j)\mathbf{w}^m + b^m) \geq 2 - \xi_i^{j,m} \quad (6)$$

$$\xi_i^{j,m} \geq 0 \quad (7)$$

$$\forall j = 1, \ldots, k, \; \forall m = 1, \ldots, k \; (m \neq j), \; \forall i = 1, \ldots, n^j.$$

where $\mathbf{x}_i^j$ is the i^{th} sample in class j and n^j is the number of training samples in that class. We will refer to this multi-class SVM formulation as M-SVM in the rest of the paper. This optimization problem seeks that the j^{th} output for $\mathbf{x}_i^j$ is larger than any other. Otherwise, $\xi_i^{j,m}$ will be nonzero for certain ms, and the sample will be penalized in the objective functional as many times as it does not follow this condition. Consequently, the penalization of any incorrectly classified sample will not depend just on the incorrectly assigned class, but also in the number of outputs larger than the true class output.

To show that this penalty strategy is not optimal for solving multi-class problems, let us look back at the Statistical Learning Theory (SLT) [1]. In the SLT, we are given a risk functional that has to be minimized with respect to the classifier $f(\mathbf{x}, \omega)$ for $\omega \in \Omega$, knowing the joint probability density function $p(y, \mathbf{x})$ and a discrepancy measure $L(\cdot, \cdot)$ between y and $f(\mathbf{x}, \omega)$:

$$R(\omega) = \int L(y, f(\mathbf{x}, \omega)) p(y, \mathbf{x}) dy d\mathbf{x} \quad (8)$$

For classification problems $L(\cdot, \cdot)$ is defined to be zero if the class given by the classifier equals the true one, and one otherwise [1]. Therefore, the proposed multi-class machine in [2] might give a higher penalty than desired over some samples, if the outputs corresponding to more than one incorrect classes are larger than the output for the correct one. And, as the weights of the SVM solution depend on how many times a sample is penalized, the "most" incorrectly classified samples (outliers) will be the ones with a higher influence on the weights of the multi-class SVM solution, which should not be the case, if we apply the Empirical Risk Minimization (ERM) inductive principle [1] over (8).

If we employed the ERM principle over (8) to get the SVM solution as in the binary case, we would be leaded to replace (6) with the following constraints:

$$(\boldsymbol{\phi}^T(\mathbf{x}_i^j)\mathbf{w}^j + b^j) - \max_{m, m \neq j} (\boldsymbol{\phi}^T(\mathbf{x}_i^j)\mathbf{w}^m + b^m) \geq 2 - \xi_i^j \quad (9)$$

$$\forall j = 1, \ldots, k, \quad \forall i = 1, \ldots, n^j$$

in which the correct class is only compared to the largest output for all the rest[1].

However, this modification can not be easily introduced in most learning problems, because it would make the problem non-linear, preventing us from using quadratic programming approaches as in the standard SVM. But we can again replace the constraints in (9) with:

$$(\boldsymbol{\phi}^T(\mathbf{x}_i^j)\mathbf{w}^j + b^j) - (\boldsymbol{\phi}^T(\mathbf{x}_i^j)\mathbf{w}^m + b^m) \geq 2 - \xi_i^j \quad (10)$$

$$\forall j = 1, \ldots, k, \; \forall m = 1, \ldots, k \; (m \neq j), \; \forall i = 1, \ldots, n^j$$

in which we have only deleted the $\max$. This can be done because the ξ_i^j for each pattern will be the one that enforces the condition for all ms, so it has to be the maximum of the $\xi_i^{j,m}$ in the M-SVM formulation. We can check that also in this case, each pattern is only allowed to penalize once in the objective function. Functional (5) with constraints (10) and $\xi_i^j \geq 0$ is our Empirical Risk SVM proposal for multi-class problems (ER-MSVM).

3. ER-MSVM RESOLUTION

As ER-MSVM constraints are linear, it is possible to solve it by using quadratic programming. The derivation would be similar to that for M-SVM [1, 2], being the major difference a new constraint for the dual problem:

$$\sum_{\substack{m=1 \\ m \neq j}}^{k} \alpha_i^{j,m} \leq C, \quad \forall j = 1, \ldots, k, \quad \forall i = 1, \ldots, n^j \quad (11)$$

where $\alpha_i^{j,m}$ are the Lagrange Multipliers associated to the constraints in (10). It is easy to interpret the meaning of these new constraints: as $\xi_i^{j,m} > 0$ (in M-SVM formulation) if and only if $\alpha_i^{j,m} = C$, (11) does not allow more than one penalization from each pattern in the objective function.

For our implementation of ER-MSVM, however, we have used an algorithm of the Iterative Recursive Weighted Least Squares (IRWLS) type [6]. In fact, the method we propose consists of a series of IRWLS problems that converge to the ER-MSVM solution. To do so, we replace constraints (10) with

$$(\boldsymbol{\phi}^T(\mathbf{x}_i^j)\mathbf{w}^j + b^j) - (\boldsymbol{\phi}^T(\mathbf{x}_i^j)\mathbf{w}^m - b^m) \geq v_i^j - \xi_i^{j,m} \quad (12)$$

[1]The $\sum_{\substack{m=1 \\ m \neq j}}^{k}$ in (5) must also be removed

1.	$v_i^j = 2, \quad \forall j = 1, \ldots, k, \quad \forall i = 1, \ldots, n^j$
2.	Do (until convergence)
-	Solve problem defined by functional (5) and constraints (7) and (12), using IRWLS algorithm
-	$e_i^{j,m} = f_m(\mathbf{x}_i^j) - f_j(\mathbf{x}_i^j) + v_i^j, \forall j = 1, \ldots, k,$ $\forall m = 1, \ldots, k \ (m \neq j), \quad \forall i = 1, \ldots, n^j$
-	For all $j = 1, \ldots, k$ and for all $i = 1, \ldots, n^j$ do if $e_i^{j,m} > 0$ for two values of m $(m \neq j)$ then $v_i^j = 0.5(\max_{m \neq j} e_i^{j,m} + \max 2_{m \neq j} e_i^{j,m})$

Table 1. Proposed algorithm to solve ER-MSMV problem

$$\forall j = 1, \ldots, k, \ \forall m = 1, \ldots, k \ (m \neq j), \ \forall i = 1, \ldots, n^j$$

where the v_i^j values are initially fixed to 2 and then iteratively modified. Basically, we would like that, for each pattern, at most one of the corresponding constraints in the above equation, is satisfied with $\xi_i^{j,m} \neq 0$, so it introduces no more than one penalty term in the functional (5). In order to accomplish with this, we will relax the margin amplitudes v_i^j until this condition is met. To be more concrete, at each stage of the algorithm, the v_i^j associated with a pattern is modified if and only if more than one of the $\xi_i^{j,m}$ from the previous iteration are different from 0, and, if so, its new value is the semisum of the two largest $\xi_i^{j,m}$ (other variation schemes are also possible, but this is the one that has resulted in a faster convergence in our experiments). Table 1 summarizes the proposed algorithm.

Finally, each basic problem can be solved with the IRWLS method for the M-SVM in a manner similar to that used for the binary classification problem described in [6].

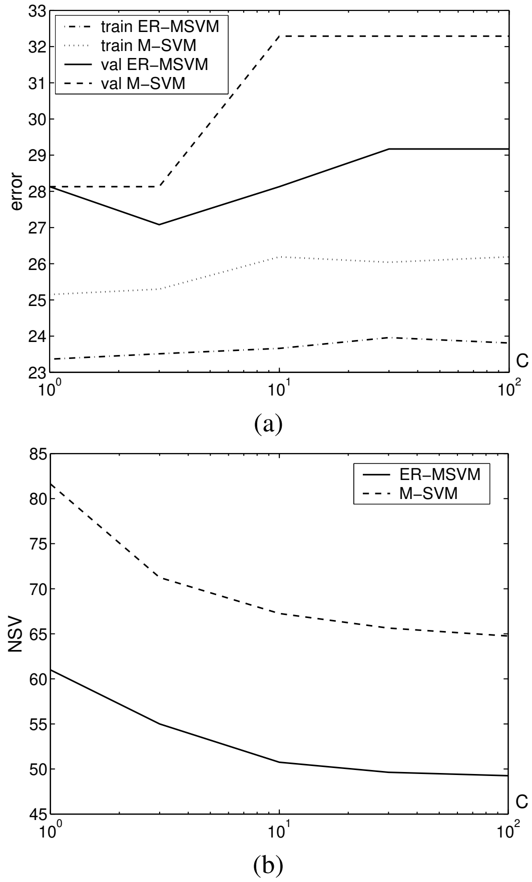

Fig. 1. Results for synthetic problem *4gauss* with linear kernel for different values of parameter C. (a) Train and 5-fold cross validation errors. (b) Number of support vectors.

4. EXPERIMENTAL RESULTS

In this section we compare the performance of the proposed ER-MSVM against that of M-SVM both in terms of performance and number of support vectors, by means of a synthetic problem and some real datasets. These experiments will serve to give a better comprehension about the differences between both methods, and the circumstances under which they obtain identical solutions.

The synthetic two-dimensional problem, *4gauss*, consists of 4 Gaussian distributions centered in (-1,0), (1,0), (0,-1) and (0,1), all of them with variance 0.36, corresponding to 4 different classes. We have generated a train partition of 100 patterns with the same a priori probabilities for the 4 classes. An independent test dataset with 10000 patterns has also been generated. Previous to applying any method, we have normalized the training dataset to have zero mean and unit variance, with the same scaling being applied to the test partition.

We have used a linear kernel for this problem, exploring different values for the only parameter: $C = [1, 3, 10, 30, 100]$. In Figure 1(a) we show the average train and validation errors for both methods when using a 5-fold cross validation scheme. We observe the superior performance of ER-MSVM for all the values of C. Results in the independent test set offer a similar behavior achieving minimum values of 22.51 and 22.95 (for ER-MSVM and M-SVM, respectively) at C=3. Figure 1(b) clearly suggests that this improvement is due to ER-MSVM avoids multiple penalizations from a single pattern, thus diminishing the number of support vectors (number of $\alpha_i^{j,m} \neq 0$). By instance, from this figure we could conclude that from the 80 support vectors of M-SVM with C=1, approximately 40 are associated to just 20 patterns with double penalization (triple penalty can be ignored because of the geometry of the problem). Thus, ER-MSVM reduces the number of support vectors to 60. In fact, the reduction is quite significant for all values of C, ranging from 20 to 25 %.

Finally, we have also tested both methods for multi-class

Problem	Method	val. err.	NSV	(C, σ)
ecoli	ER	12.21	166.8	(3,$\sqrt{3.5}$)
	M	13.4	182	
teaching	ER	40.41	122.4	(100,$\sqrt{5}$)
	M	39.74	143.2	
thyroid	ER	2.79	26	(30,$\sqrt{5}$)
	M	2.79	26	
zoo	ER	3.95	96.6	(10,$\sqrt{32}$)
	M	3.95	99.4	

Table 2. Summary of results for the real datasets. 5-fold cross validation error and number of support vectors are displayed for ER-MSVM (ER) and M-SVM (M) methods, as well as the C and sigma parameters that have been used.

SVM solving in four datasets from the UCI Machine Learning Repository [7]: *ecoli*, *teaching*, *thyroid* and *zoo*. *ecoli* has 8 classes, input dimension 7 and 336 samples. *teaching* and *thyroid* are both 3 class-problems with 5 input variables and 151 and 215 patterns, respectively. *zoo* is a problem with 101 samples belonging to 7 classes, and input dimension 16.

All problems have been preprocessed for having zero mean and unit variance. We have then carried out simulations on both methods, using Gaussian kernels, for all possible combinations of C and σ from $C = [1, 3, 10, 30, 100]$ and $\sigma = [\sqrt{4d}, \sqrt{2d}, \sqrt{d}, \sqrt{0.5d}, \sqrt{0.25d}]$, with d being the dimension of the input data. As no standard train/test partitions are defined, we have used 5-fold cross validation in all datasets.

In Table 2 we show the best validation results achieved by both methods together with the number of support vectors used and the corresponding parameters. In *zoo* and *thyroid*, both methods have obtained the same solution, but for different reasons. If we examined *zoo* results for all the combination of parameters, we could see that for every σ the solution never changes when increasing parameter C above a certain value. Thus, the problem is being solved with zero train error, and so all $\xi_i^{j,m} = 0$, making ER-MSVM and M-SVM solutions identical. *thyroid* solutions are also usually coincident but this is for a different reason. In this case it is the low overlapping among classes (validation errors are inferior to 3%), and also the fact that this problem has just three classes, which make it difficult that double penalizations occur.

On the other hand, *ecoli* and *teaching* have systematically given different solutions for ER-MSVM and M-SVM. ER-MSVM best result in *teaching* is slightly worse than that of M-SVM. However, the low number of classes of the problem, together with the high overlap, suggests that maybe all the effects from double penalizations are compensated. The reduction in the number of support vectors is consistent with the theory. Finally, *ecoli* is a problem with a high number of classes, what certainly makes it more difficult to compensate the multiple penalizations of different patterns in M-SVM. In this case, ER-SVM has outperformed M-SVM for more than 1%.

5. CONCLUSIONS

In this paper we have shown that following the Statistical Learning Theory we have been able to develop a new learning algorithm for solving multi-class problems using SVMs. This procedure has the advantage of providing solutions that are as good as the previously proposed schemes or better, and, at the same time, reducing the number of support vectors, without an increase in the training time. The classification error is not a great advantage in most real systems because the samples that are incorrectly classified in the training set are not many times penalized. Anyhow the proposed scheme is theoretically more sound and in the presence of many outliers will perform better, while having the advantage of providing solutions with far less support vectors.

6. REFERENCES

[1] V. Vapnik, *Statistical Learning Theory*, Wiley, 1998.

[2] J. Weston and C. Watkins, “Multi-class support vector machines,” CSD-TR-98-04, Royal Holloway, 1998.

[3] E. L. Allwein, R. E. Schapire, and Y. Singer, “Reducing multiclass to binary: A unifying approach for margin classifiers,” *JMLR*, pp. 113–141, 2000.

[4] C.-W. Hsu and C.-J. Lin, “A comparison of methods for multiclass support vector machines,” *IEEE TNN*, vol. 13, no. 2, pp. 415–425, Mar. 2002.

[5] B. Schölkopf, K.-K. Sung, C. J.C. Burges, F. Girosi, P. Niyogi, T. Poggio, and V. N. Vapnik, “Comparing support vector machines with Gaussian kernels to radial basis function classifiers,” *IEEE TSP*, vol. 45, no. 11, pp. 2758–2765, Nov. 1997.

[6] F. Pérez-Cruz, P. L. Alarcón-Diana, A. Navia-Vázquez, and A. Artés-Rodríguez, “Fast training of support vector classifiers,” in *NIPS 13*, Nov. 2000, M.I.T. Press.

[7] C.L. Blake and C.J. Merz, http://www.ics.uci.edu/~mlearn/MLRepository.html, University of California, Irvine, Dept. of Information and Computer Sciences, 1998.