K-SVCR. A Multi-class Support Vector Machine

Cecilio Angulo[1] and Andreu Català[2]

[1] Dept. of Systems Engineering, Polytechnical University of Catalonia
E-08222 Terrassa, Spain
cangulo@esaii.upc.es
[2] LEA-SICA, European Associated Lab. Intelligent Systems and Advanced Control
E-08800 Vilanova i la Geltrú, Spain
andreu@esaii.upc.es

Abstract. Support Vector Machines for pattern recognition are addressed to binary classification problems. The problem of multi-class classification is typically solved by the combination of 2-class decision functions using voting scheme methods or decison trees. We present a new multi-class classification SVM for the separable case, called K-SVCR. Learning machines operating in a kernel-induced feature space are constructed assigning output +1 or -1 if training patterns belongs to the classes to be separated, and assigning output 0 if patterns have a different label to the formers. This formulation of multi-class classification problem ever assigns a meaningful answer to every input and its architecture is more fault-tolerant than standard methods one.

1 Introduction

The problem of multi-class classification from examples addresses the general problem of finding a decision function f, approximation of an unknown function $\widehat{f}$, defined from an input space Ω into an unordered set of classes $\{\theta_1, \ldots, \theta_K\}$, given a training set

$$\mathcal{T} = \{(\mathbf{x}_p, y_p = f(\mathbf{x}_p))\}_{p=1}^{\ell} \subset \Omega \times \{\theta_1, \ldots, \theta_K\}. \quad (1)$$

Support Vector Machines (SVMs) that learn classification problems - in short SVMC -, are specific to binary classification problems, also called dichotomies. The problem of multi-class classification (K ¿ 2) is typically solved by the combination of 2-class decision functions.

In this paper we present a new multi-class classification SVM for the separable case, called K-SVCR. When K ¿ 2, we will construct learning machines assigning output $+1$ or -1 if training patterns belongs the classes to be separated, and output 0 if patterns belongs a different class to the formers. So, we are forcing the computed separating hyperplane to cover all the '0-label' training patterns. Like in the construction of SVMs, the new method exploits the basic idea of map the data from the input space Ω into some other higher dimension dot product space $\mathcal{F}$, called feature space, via a non linear map and perform the above linear algorithm in $\mathcal{F}$. The associated restricted QP-problem could be

R. López de Mántaras, E. Plaza (Eds.): ECML 2000, LNAI 1810, pp. 31–38, 2000.

subject to

$$\alpha_i \geq 0, \quad i = 1, \ldots, \ell \tag{8}$$
$$\sum_{i=1}^{\ell} \alpha_i y_i = 0.$$

The hyperplane decision function can thus be written as

$$f(\mathbf{x}) = sign\left(\sum_{i=1}^{SV} \alpha_i y_i k(\mathbf{x}_i, \mathbf{x}) + b\right), \tag{9}$$

where b is computed using the Karush-Kuhn-Tucker complementary conditions

$$\alpha_i \cdot [y_i \cdot (\langle \mathbf{w}, \mathbf{x}_i \rangle_{\mathcal{F}} + b) - 1] = 0, \quad i = 1, \ldots, \ell. \tag{10}$$

Among all the training patterns, only a few of them have an associated weight α_i non-zero in the expansion (9). These elements lie on the margin - some strict constraint in (6) is accomplished - and them are called support vectors.

To generalize the SV algorithm to regression estimation, 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

$$|y - f(\mathbf{x})|_{\varepsilon} \stackrel{def}{=} \max\{0, |y - f(\mathbf{x})| - \varepsilon\}. \tag{11}$$

For a priori chosen $\varepsilon \geq 0$, the associated constrained optimization problem for the separable case is

$$\arg\min \tau(\mathbf{w}) = \frac{1}{2} \|\mathbf{w}\|_{\mathcal{F}}^2 \tag{12}$$

subject to

$$\begin{aligned} (\langle \mathbf{w}, \mathbf{x}_i \rangle_{\mathcal{F}} + b) - y_i \leq \varepsilon, \quad i = 1, \ldots, \ell \\ y_i - (\langle \mathbf{w}, \mathbf{x}_i \rangle_{\mathcal{F}} + b) \leq \varepsilon, \quad i = 1, \ldots, \ell. \end{aligned} \tag{13}$$

Introducing Lagrange multipliers, we arrive at the constrained optimization problem: find multipliers $\alpha_i, \alpha_i^* \geq 0$ which

$$\begin{aligned} \min W(\alpha, \alpha^*) = \frac{1}{2} \sum_{i,j=1}^{\ell} (\alpha_i^* - \alpha_i) k(\mathbf{x}_i, \mathbf{x}_j) (\alpha_j^* - \alpha_j) + \\ + \varepsilon \sum_{i=1}^{\ell} (\alpha_i^* + \alpha_i) - \sum_{i=1}^{\ell} (\alpha_i^* - \alpha_i) \end{aligned} \tag{14}$$

subject to

$$\alpha_i, \alpha_i^* \geq 0, \quad i = 1, \ldots, \ell \tag{15}$$

$$\sum_{i=1}^{\ell} (\alpha_i - \alpha_i^*) = 0.$$

The regression estimate takes the form:

$$f(\mathbf{x}) = \sum_{i=1}^{SV} (\alpha_i^* - \alpha_i)\, k(\mathbf{x}_i, \mathbf{x}) + b. \tag{16}$$

The solution expands again in terms of a subset of the training patterns, and b is calculated from (13) in strict equal form over the support vectors.

3 Multi-class Support Vector Machines

The standard method of decomposing a general classification problem into dichotomies is to place K binary classifiers in parallel. In the original method [3,10], the ith SVMC is trained with positive labels for all the examples in the ith class, and negative labels for all other examples. We refer to SVMs trained in this way as 1-v-r SVMCs - short for one-versus-rest -. The training time of the standard method scales linearly with K.

The output scale of a SVM is determined so that the separating hyperplane is in canonical form, i.e., support vector output is ± 1. In [6] is asserted that this scale is not robust, since it depends on just a few points, often including outliers, and different alternatives are proposed to circumvent this problem

Another general method to construct multi-class classifiers is to build all possible binary classifiers - $K{\cdot}(K\text{-}1)/2$ hyperplane decision functions - from a training set of K classes, each classifier being trained on only two out of K classes. We refer to the SVMCs trained with this method like 1-v-1 SVMCs - short for one-versus-one -. The combination of these binary classifiers to determine the label assigned to each new input can be made by different algorithms, for example the voting scheme [4]. The 1-v-1 approach is, in general, preferable to the 1-v-r one [5]. Unfortunately, the size of the 1-v-1 classifier may grow superlinearly with K.

In addition to these two general methodologies, it is possible to construct multi-class classifiers combining 1-v-1 SVMCs with decision trees, that are able to handle many classes. In [8] a learning architecture is presented, the DAGSVM algorithm, which operates in a kernel-induced feature space and uses 2-class maximal margin hyperplanes at each decision-node of the Decision Directed Acyclic Graph (DDAG). The class of functions implemented naturally generalizes the class of decision trees.

In [1] the relationship between SVMC and a family of mathematical programming methods (MPM) are examined and a new method for nonlinear discrimination, the Support Vector Decision Tree (SVDT), is generated. It construct decision trees in which each decision is a support vector machine. In this sense, the architecture method is similar to the DAGSVM algorithm.

Working in a different way, in [11] the original SVMC constrained optimization problem is redefined and generalized to construct a decision function by

considering all classes at once. The K-SVCR multi-class classification method is also defined in this sense, the constrained QP problem is redefined, but we are not considering the classification of all classes at once. In the other hand, it is possible to make an extension of our algorithm to capture the advantageous properties of the DAGSVM algorithm.

4 *K*-SVCR Learning Machine

Given the training set $\mathcal{T}$ defined in (1) we would find a decision function f in the form (3) with:

$$\begin{aligned} f(\mathbf{x}_p) &= +1, \quad p = 1, \ldots, \ell_1 \\ &= -1, \quad p = \ell_1 + 1, \ldots, \ell_1 + \ell_2 \\ &= 0, \quad p = \ell_1 + \ell_2 + 1, \ldots, \ell, \end{aligned} \tag{17}$$

where, without loss of generality, we suppose the first $\ell_{12} = \ell_1 + \ell_2$ patterns corresponding to the two classes to be separated, and the other patterns ($\ell_3 = \ell - \ell_{12}$) belonging to any different class - we will label them with 0 -.

Obviously, in general, do not exist any hyperplane accomplishing the constraints (17) in the input space Ω, and hence is useless looking for a linear solution to the problem in this space. But, if we insert this space via a nonlinear map into a feature space with a dimension high enough, the hyperplane capacity to accomplish the constrains increase, and it will be possible to find a solution.

For instance, when we solve the QP problem leading to the SVMC solution it is very usual to formulate the problem with $b = 0$, which is equivalent to require that all hyperplanes contain the origin. This is considered a mild restriction for high dimensional spaces, since it is equivalent to reduce the number of degrees of freedom by one [2].

The requirement of the K-SVCR learning machine is higher. It requires that optimal hyperplane contains all ℓ_3 training patterns with label 0.

We define below the constrained optimization problem associated to K-SVCR method, for the separable case: for $0 \leq \delta < 1$ chosen a priori,

$$\arg\min \tau(\mathbf{w}) = \frac{1}{2} \|\mathbf{w}\|_{\mathcal{F}}^2 \tag{18}$$

subject to

$$\begin{aligned} y_i \cdot (\langle \mathbf{w}, \mathbf{x}_i \rangle_{\mathcal{F}} + b) - 1 \geq 0, &\quad i = 1, \ldots, \ell_{12} \\ \langle \mathbf{w}, \mathbf{x}_i \rangle_{\mathcal{F}} + b \leq \delta, &\quad i = \ell_{12} + 1, \ldots, \ell \\ \langle \mathbf{w}, \mathbf{x}_i \rangle_{\mathcal{F}} + b \geq \delta, &\quad i = \ell_{12} + 1, \ldots, \ell, \end{aligned} \tag{19}$$

with a decision function solution similar to (3), defined by

$$\begin{aligned} f(\mathbf{x}) &= +1, \quad \text{if} \quad \langle \mathbf{w}, \mathbf{x} \rangle_{\mathcal{F}} + b > \delta \\ &= -1, \quad \text{if} \quad \langle \mathbf{w}, \mathbf{x} \rangle_{\mathcal{F}} + b < \delta \\ &= 0, \quad \text{otherwise.} \end{aligned} \tag{20}$$

If $\delta = 0$ then decision function (20) is the same as (3) and we are exactly requiring that the separating hyperplane will contains the last ℓ_3 training patterns. Nonetheless, this imposition implies no generalization for the '0-label class', no sparsity in the support vectors set over the training patterns with label 0 [9], and higher computational cost. So, even if our task is learning pattern recognition, it could seems that we make a certain use of the ε-insensitive loss function (11) employed in the SVMR method for the output $y_i = 0$.

A solution for the problem defined in (18) and (19) can be found by locating the saddle point of the Lagrangian

$$\begin{aligned} L\left(\mathbf{w}, b, \alpha, \beta, \beta^*\right) = \frac{1}{2}\left\|\mathbf{w}\right\|_{\mathcal{F}}^2 & - \sum_{i=1}^{\ell_{12}} \alpha_i \left[y_i \left(\left\langle \mathbf{w}, \mathbf{x}_i \right\rangle_{\mathcal{F}} + b\right) - 1\right] + \\ & + \sum_{i=\ell_{12}+1}^{\ell} \beta_i \left[y_i \left(\left\langle \mathbf{w}, \mathbf{x}_i \right\rangle_{\mathcal{F}} + b\right) - \delta\right] - \\ & - \sum_{i=\ell_{12}+1}^{\ell} \beta_i^* \left[y_i \left(\left\langle \mathbf{w}, \mathbf{x}_i \right\rangle_{\mathcal{F}} + b\right) - \delta\right] \end{aligned} \tag{21}$$

with constraints

$$\begin{aligned} \alpha_i \geq 0, \quad & i = 1, \ldots, \ell_{12} \\ \beta_i, \beta_i^* \geq 0, \quad & i = \ell_{12}+1, \ldots, \ell. \end{aligned} \tag{22}$$

which has to be maximized with respect to the dual variables α_i and β_i, β_i^* and minimized with respect to the primal variables $\mathbf{w}$ and b. In the saddle point the solution should satisfy the conditions, leading to

$$\begin{aligned} \mathbf{w} &= \sum_{i=1}^{\ell_{12}} \alpha_i y_i \mathbf{x}_i - \sum_{i=\ell_{12}+1}^{\ell} \left(\beta_i - \beta_i^*\right) \mathbf{x}_i \\ 0 &= \sum_{i=1}^{\ell_{12}} \alpha_i y_i - \sum_{i=\ell_{12}+1}^{\ell} \left(\beta_i - \beta_i^*\right) \mathbf{x}_i. \end{aligned} \tag{23}$$

Finally, if we define

$$\begin{aligned} \gamma_i &= \alpha_i y_i, \quad i = 1, \ldots, \ell_{12} \\ \gamma_i &= \beta_i, \quad i = \ell_{12}+1, \ldots, \ell \\ \gamma_i &= \beta_{i-\ell_3}^*, \quad i = \ell+1, \ldots, \ell+\ell_3 \end{aligned} \tag{24}$$

the primal variables are eliminated and we arrive at the Wolfe dual of the optimization problem: for $0 \leq \delta < 1$ chosen a priori

$$\arg\min L\left(\gamma\right) = \frac{1}{2}\gamma^T \cdot \mathbf{H} \cdot \gamma + \mathbf{c}^T \cdot \gamma \tag{25}$$

with

$$\gamma^T = (\gamma_1, \ldots, \gamma_\ell, \gamma_{\ell+1}, \ldots, \gamma_{\ell+\ell_3}) \in \mathbb{R}^{\ell_{12}+\ell_3+\ell_3} \tag{26}$$
$$\mathbf{c}^T = \left(\frac{-1}{y_1}, \cdots, \frac{-1}{y_{\ell_{12}}}, \delta, \ldots, \delta\right) \in \mathbb{R}^{\ell_{12}+\ell_3+\ell_3}$$
$$\mathbf{H} = \begin{pmatrix} (k(\mathbf{x}_i, \mathbf{x}_j)) & -(k(\mathbf{x}_i, \mathbf{x}_j)) & (k(\mathbf{x}_i, \mathbf{x}_j)) \\ -(k(\mathbf{x}_i, \mathbf{x}_j)) & (k(\mathbf{x}_i, \mathbf{x}_j)) & -(k(\mathbf{x}_i, \mathbf{x}_j)) \\ (k(\mathbf{x}_i, \mathbf{x}_j)) & -(k(\mathbf{x}_i, \mathbf{x}_j)) & (k(\mathbf{x}_i, \mathbf{x}_j)) \end{pmatrix} = \mathbf{H}^T \in \mathcal{S}\left(\mathbb{R}^{\ell+\ell_3}\right),$$

subject to

$$\begin{aligned} \gamma_i \cdot y_i \geq 0, \quad & i = 1, \ldots, \ell_{12} \\ \gamma_i \geq 0, \quad & i = \ell_{12}, \ldots, \ell + \ell_3 \end{aligned} \tag{27}$$

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

The hyperplane decision function can be written as

$$\begin{aligned} f(\mathbf{x}) = +1, \quad & \text{if} \quad \sum_{i=1}^{SV} \nu_i k(\mathbf{x}_i, \mathbf{x}) + b > \delta \\ = -1, \quad & \text{if} \quad \sum_{i=1}^{SV} \nu_i k(\mathbf{x}_i, \mathbf{x}) + b < \delta \\ = 0, \quad & \text{otherwise} \end{aligned} \tag{28}$$

where

$$\begin{aligned} \nu_i = \gamma_i, \quad & i = 1, \ldots, \ell_{12} \\ \nu_i = \gamma_{i+\ell_3} - \gamma_i, \quad & i = \ell_{12}+1, \ldots, \ell, \end{aligned} \tag{29}$$

and b is calculated from (19) in strict equal form over the support vectors in terms of parameters γ_i. We observe that the third constraint in (27) can be written as

$$\sum_{i=1}^{SV} \nu_i = 0. \tag{30}$$

This formulation of multi-class classification problem is more fault-tolerant than the 1-v-r general method, because there exist more redundancy in the answers [7]. On the other hand, all the K-SVCRs answers have sense: each machine classifies any input into a class, the two class implicated in the binary classification or into the 'rest' class (0-label class). The 1-v-1 general classification method is more fault-tolerant that the 1-v-r one, but the classifiers give no sense answers if the evaluated input does not belong to the classes implicated in the binary classification.

5 Conclusions and Further Research

The K-SVCR algorithm, a novel learning machine based in SVMs for multi-class pattern recognition for the separable case is presented. This algorithm construct a decision function to separate two classes containing the patterns of all the others classes. These 1-v-1 SVMCs can be combined in an "AND" scheme, in a voting scheme or in a decision tree formulation. Two initial schemes are easily implemented, meanwhile the last formulation is part of our actual study, employing a DDAG architecture to reduce the evaluation time and control the generalization performance.

Further research involves the test of the method on large data sets and a more detailed comparison with other methods over real data benchmarks.

A generalization of the K-SVCR procedure for the non-separable case is being developed in the present, and future work will establish a comparison between the generalized algorithm and a modification over the sensitivity parameter for the present formulation.

References

1. Bennett, K.P.: Combining Support Vector and Mathematical Programming Methods for Classification. In: Schölkopf, B., Burges, C.J.C., Smola, A.J. (eds.): Advances in Kernel Methods - Support Vector Learning. MIT Press, Cambridge, MA (1999) 34
2. Burges, C.J.C.: A Tutorial on Support Vector Machines for Pattern Recognition. Data Mining And Knowledge Discovery **2** (1998) 1-47 32, 35
3. Cortes, C., Vapnik, V.: Support Vector Networks. Machine Learning **20** (1995) 273-297 32, 34
4. Friedman, J.H.: Another Approach to Polychotomous Classification. Tech. Report, Dept. Statistics, Stanford Univ. (1996) 34
5. Kreβel, U.: Pairwise classification and support vector machines. In: Schölkopf, B., Burges, C.J.C., Smola, A.J. (eds.): Advances in Kernel Methods - Support Vector Learning. MIT Press, Cambridge, MA (1999) 34
6. Mayoraz, E., Alpaydin, E.: Support Vector Machines for Multi-class Classification. Proc. 5th IWANN, Alicante, Spain (1999) 833-842 34
7. Moreira, M., Mayoraz, E.: Improved Pairwise Coupling Classification with Correcting Classifiers. In: Nédellec, C., Rouveirol, C. (eds.): Lecture Notes in Artificial Intelligence, Vol. 1398. Springer-Verlag, Berlin Heidelberg New York (1998) 160–171 37
8. Platt, J., Cristianini, N., Shawe-Taylor, J.: Large Margin DAGs for Multiclass Classification. Advances in NIPS **12,** MIT Press, Cambridge, MA (2000) To appear 34
9. Smola, A.J.: Learning with Kernels. Doctoral thesis. TU Berlin (1998) 32, 36
10. Vapnik, V.: The Nature of Statistical Learning Theory. Springer Verlag, Berlin Heidelberg New York (1995) 32, 34
11. Weston, J., Watkins, C.: Multi-class Support Vector Machines. Tech. Report CSD-TR-98-04, Royal Holloway, Univ. of London, Egham, UK (1998) 34