SETIT 2007
4th International Conference: **S**ciences of **E**lectronic, **T**echnologies of **I**nformation and **T**elecommunications
March 25-29, 2007 – TUNISIA

An Improvement of One-against-all Method for Multi-class Support Vector Machine

Yang Liu*, Rui Wang** Yingsheng Zeng* and Hangen He*

* *Institute of Mechatronics and Automation, National University of Defense Technology*
lyhn12003@yahoo.com.cn
lyhn12003@hotmail.com

** *School of Computer Science, National University of Defense Technology*
ruiwang@nudt.edu.cn

Abstract: The support vector machine (SVM) has an excellent ability to solve binary classification problems, and how to process multi-class problems with it is one of the present focuses. Among the existing multi-class SVM methods, there are one-against-one method, one-against-all method, and some other methods. Generally speaking, each of them has their advantages and disadvantages. This paper presents an improved technique of the one-against-all method for multi-class Support Vector Machine. It has the faster learning speed than the traditional methods. The experimental results show that the number of the hyper-planes has been reduced and the accuracy of identification are improved significantly compared to that of the traditional methods.
Key words: Support vector machine, multi-class problems, one-against-all method.

1. Introduction

The support vector machine (SVM) [1], [2] is originally introduced for binary classification problems, and it has an excellent ability to solve these problems. And how to process multi-class problems with it is one of the present focuses. Many methods have been proposed to solve multi-class problems. There are three important ones: one-against-one [3], [4], one-against-all [3], [4], and directed acyclic graph SVM (DAGSVM) [3]. The most important index for evaluating the classifier is the accuracy of the method. In this aspect, the one-against-one method is better than the others. In applications, only one index is not enough, the complexity of computation should be considered. For an n-class problem, both one-against-one and DAGSVM [3] methods construct n*(n-1)/2 hyper-planes. The one-against-all method constructs n hyper-planes where each one is constructed by using all data from the training set. Although the one-against-one and the DAGSVM methods construct n*(n-1)/2 hyper-planes, the training time is less than the one-against-all method because they use a small number of training data for learning. So, it seems that the one-against-one and the DAGSVM methods are more suitable than one-against-all method. But in [5], it disagrees with a large body of some published work on multi-class classification and believes that the one-against-all method is as accurate as any other approach.

In this paper, we consider an improved technique of one-against-all method. It uses a new classification method. In some cases, the traditional one-against-all method will not classify all the classes accurately. The proposed method firstly classifies one class from the other n-1 classes, and then it processes the classification in the rest n-1 classes, and so on. This method will not increase the number of hyper-planes and can improve the accuracy of one-against-all method. Furthermore, the proposed method can resolve the unclassifiable regions in a much better way than the traditional methods.

In Section2, we briefly describe the binary SVM and the one-against-all method of multi-class problems. In Section3, we introduce our algorithms. In Section4, experimental results are reported to illustrate the superiority of the proposed method. Finally, concluding remarks are discussed in Section5.

2. Review of SVM for classification

2.1. Support vector machine

Consider a set of training examples $D = \{(x_1, y_1), (x_2, y_2), ..., (x_N, y_N)\}$,

where the ith sample $x_i \in R^n$ (n is the dimension of the input space) belongs to two separate classes labeled by $y_i \in \{-1,1\}$ .The classification problem is to find a hyper-plane in a high dimensional feature space Z , which divides the set of examples in the feature space such that all the points with the same label are on the same side of the hyper-plane [6], [7]. SVM is to construct a map $z = \phi(x)$ from the input space R^n to a high-dimensional feature space Z and to find an "optimal" hyper-plane $w^T z + b = 0$ in Z such that the separation margin between the positive and negative examples is maximized. A decision function of the classifier is then given by $f_{w,b} = \text{sgn}[w^T z + b]$,

where w is a weight vector and b is a threshold. Without loss of generality, we consider the case when the training set is not linearly separable. The SVM classification amounts to finding w and b satisfying

$$\min \frac{1}{2} w^T w + c\sum_{i=1}^{N} \varepsilon_i$$
$$s.t. \begin{cases} y_i[w^T \phi(x_i) + b] \geq 1 - \varepsilon_i, i = 1,..., N \\ \varepsilon_i \geq 0, i = 1,..., N \end{cases} \quad (1)$$

Where c > 0 is a regularization parameter for the tradeoff between model complexity and training error, and ε_i measures the (absolute) difference between $w^T z + b$ and y_i . Solving (1) directly is more complex because of a number of variables and unknown $\phi(x)$. Thus, solving (1) is converted into solving a dual problem

$$\max -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} y_i y_j (\phi(x_i)^T \phi(x_j))\alpha_i \alpha_j + \sum_{i=1}^{N} \alpha_i$$
$$s.t. \begin{cases} \sum_{i=1}^{N} \alpha_i y_i = 0 \\ 0 \leq \alpha_i \leq c, i = 1,..., N \end{cases} \quad (2)$$

Let a kernel function $K(x, y)$ satisfying $K(x_i, x_j) = \phi(x_i)^T \phi(x_j)$. The above dual problem becomes

$$\min \frac{1}{2}\sum_{i=1}^{N}\sum_{i=1}^{N} y_i y_j K(x_i, x_j)\alpha_i \alpha_j - \sum_{i=1}^{N} \alpha_i$$
$$s.t. \begin{cases} \sum_{i=1}^{N} \alpha_i y_i = 0 \\ 0 \leq \alpha_i \leq c, i = 1,..., N \end{cases} \quad (3)$$

Moreover, the decision function of the classifier can be represented as

$$f(x) = \text{sgn}[\sum_{i=1}^{N} \alpha_i y_i K(x_i, x) + b] \quad (4)$$

For convenient computation here, let $a_i = \alpha_i y_i$. Then (3) can be equivalently written as

$$\min \frac{1}{2}\sum_{i=1}^{N}\sum_{i=1}^{N} a_i a_j K(x_i, x_j) - \sum_{i=1}^{N} a_i y_i$$
$$s.t. \begin{cases} \sum_{i=1}^{N} a_i = 0 \\ -c_i^1 \leq a_i \leq c_i^2, i = 1,..., N \end{cases} \quad (5)$$

Where for $i = 1,..., l$, $c_i^1 = -c(\text{sgn}(1 - y_i))$ and $c_i^2 = c(\text{sgn}(1 + y_i))$. Therefore, the learning problem in SVM is equivalent to the quadratic programming problem in (5) with N bounded variables.

2.2. One-against-all method

For a k-class problem, the one-against-all method constructs k SVM models [2], [3], [4]. The ith SVM is trained with all of the training examples in the ith class with positive labels and the others with negative labels. The final output of the one-against-all method is the class that corresponds to the SVM with the highest output value. Given a set of training examples $D = \{(x_1, y_1), (x_2, y_2), ..., (x_N, y_N)\}$, where the ith sample $x_i \in R^n$ and $y_j \in \{1,..., k\}$ is the class of x_j , the ith SVM solves the following optimization problem:

$$\min \frac{1}{2}(w^i)^T w^i + c\sum_{i=1}^{N} \varepsilon_j^i$$

subject to

$$(w^i)^T \phi(x_j) + b^i \geq 1 - \varepsilon_j^i, y_j = i$$
$$(w^i)^T \phi(x_j) + b^i \geq -1 + \varepsilon_j^i, y_j \neq i \quad (6)$$
$$\varepsilon_j^i \geq 0, j = 1,..., N$$

The decision function of the ith SVM is

$$f_i(x) = (w^i)^T \phi(x) + b^i \quad (7)$$

A point x is in the class that corresponds to the largest value of the decision functions:
the class of

$$x = \arg\max_{i=1,...,k}((w^i)^T \phi(x) + b^i) \quad (8)$$

3. The proposed method

The one-against-all method is dividing each class from the others. Although it only constructs k hyper-planes, because of its error rate, it is not the best method of multi-class problems. We can consider some improvement of the original method. First, because we want to improve this method, the number of the hyper-planes should not be increased. But on the other hand, resolving the k-classes problem need at least k hyper-planes. So the number of the hyper-planes will not change. After fixed the number of the hyper-planes, we should use these hyper-planes to gain higher classification accurate. Now, all steps will

be stated below:

First we find one class that can be divided from the other k-1 classes correctly, and if it can not be divided correctly, find one class which minimizes the classification error probability. This class is denoted by class1, and then we find the corresponsive hyper-plane.

In the residual k-1 classes, finding one class which can be divided from the other k-2 classes correctly, if it can not be divided correctly, find one class which minimizes the classification error probability. This class is denoted by class2, and then finding the corresponsive hyper-plane. Using the 2 hyper-planes found in step1 and step2, we can divided class2 from the k-1 other classes.

Ignoring the first 2 classes, repeat the above process to divide the rest of the classes, and continue the steps until the classification is complete.

At the end of the process, we surprisingly find that the improved method only needs k-1 hyper-planes. Because the last class does not need another hyper-plane, it can be separated from the others by using the existing k-1 hyper-planes.

In this method, we only use k-1 hyper-planes to classify k classes, but we do not divide each class from the others by using only one corresponsive hyper-plane. From the proposed method, class2 is divided from the others by 2 hyper-planes, class3 is divided from the others by 3 hyper-planes, and at last, class k-I and class k is divided from the others by k-1 hyper-planes.

Now, we use a 4-classes problem to illustrate the superiority of this improvement. Look at figure 1 and figure 2, the 4 classes are called class1 (squares), class2 (rotundities), class3 (rhombuses), class4 (triangles), respectively. In figure 1, we use the traditional one-against-all method, and it is clear that both hyper-plane2 and hyper-plane4 make some mistakes. Using only one hyper-plane to divided one class from the others is not enough, because some classes are unclassifiable by one hyper-plane.

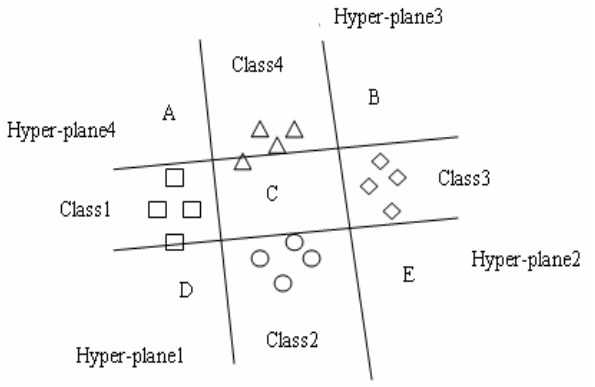

Fig. 1: The traditional one-against-all method

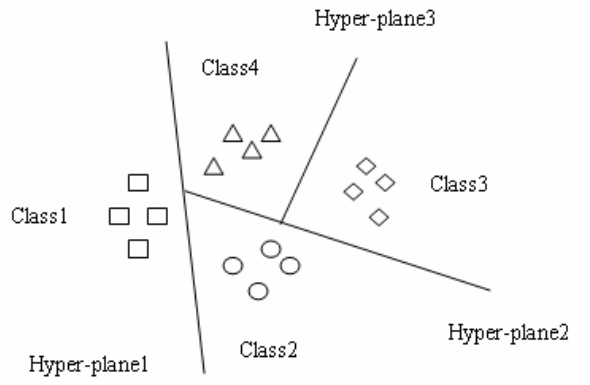

Fig. 2: The proposed method

Figure 2 illustrates the hyper-planes of the proposed method. From the figure, it is explicit that only 3 hyper-planes can classify 4 classes accurately. Class2 is divided from class1 by hyper-plane1, and divided from class3, class4 by hyper-plane2. Class3 is divided from class1, 2, and 4, by hyper-plane1, 2, and 3, respectively. Class4 is divided from class1, 2, and 3, by hyper-plane 1, 2, and 3, respectively. Compared with the traditional method, class k (k=2, 3) uses k-1 more hyper-planes to finish the classification in this method, so the accuracy will be increased. On the other hand, class 3 and class 4 use the same hyper-planes to finish the classification, so we only need 3 hyper-planes. The number of the hyper-planes is reduced, so the time of the classification can be improved.

Further more, in figure 1, there are 5 regions called region A, B, C, D, and E, respectively. If some testing examples in these regions, they are unclassifiable, this will increase the error rate. This is an inevitable problem in the traditional method. But in figure 2, there is no region that likes the regions A, B, C, D, and E, in figure 1. It means that the inevitable problem in the traditional method has been solved.

4. Experimental results

In this section, the performance evaluation of the proposed algorithm is given. Figure 3 shows the test image for the simulations.

Fig. 3: The test image

The purpose is to recognize the plane out of figure 3. We sort the pixels in the image into 4 categories through the RGB value (The colorized images have

been changed into monochrome images). The traditional method uses 4 linear discrimination equations to implement the classification. When a pixel locates at the uncertain zone, it will be put into some category randomly.

(a)The traditional method (b) The proposed method

Fig. 4: The simulation results

The traditional method (the left one) uses 4 linear discrimination equations to classify the pixels, and there is uncertain zone. The proposed algorithm (the right one) uses 3 linear discrimination equations, and there is no uncertain zone. As illustrated in the right image of figure 4, the proposed algorithm recognizes the outline of the plane more clearly and achieves better classification results than the traditional ones. Further more, by using the original method, the background noise is obvious, but the proposed method can eliminate the background noise effectively.

5. Conclusions

In this paper, we proposed an improved method based on the traditional one-against-all method. It has a faster speed than the traditional method because it only uses k-1 hyper-planes to classify k classes. Secondly, all classes use more than one hyper-plane in the classification except the class 1, so it can get higher classification accuracy than the traditional method. Further more, the proposed method can resolve the unclassifiable regions. In sum, the proposed method really improves the performance of the existing one-against-all method and has been of great value.

Although the proposed method has its own advantages, it is not perfect. We also have some work to do in the future. The generalization of this training method is the first work which is worth to consider. To improve the generalization of the method, we can use some methods of optimization to adjust several unreasonable hyper-planes. Another valuable work is to improve the accuracy of the method. It is more accurate than the traditional one-against-all method, but it is not always better than the one-against-one method, we should use the advantages of the one-against-one method to make some progress. The future work will mainly focus on these two important aspects.

REFERENCES

[1] A.B. Smith, C.D. Jones, and E.F. Roberts, “Article Title,” Journal, Publisher, Location, pp. 1-10, Date. Cristianini N, Shawe-Taylor J (2000) An introduction to support vector machines and other kernel-based learning methods. Cambridge University Press, Cambridge, UK.

[2] Platt JC, Cristianini N, Shawe-Taylor J (2000) Large margin DAGs for multi-class classification. In: Advances in neural information processing systems, vol 12, MIT Press, Cambridge, MA, pp 547-553.

[3] Hsu C-W, Lin C-J (2002) A comparison of methods for multi-class support vector machines. IEEE Trans Neural Networks, 13:415-425.

[4] Ryan Rifkin, Aldebaro Klautau, In defense of one-vs-all classification. Journal of Machine Learning Research 5 (2004) 101-141.

[5] V. Vapnik, The nature of statistical learning theory. New York: Springer-Verlag, 1995.

[6] C. Cortes and V. Vapnik, Support-vector networks. Mach. Learn., vol. 20, pp. 273–297, 1995.

[7] Krebel UH-G (1998) Pairwise classification and support vector machines. In: Scholkopf B, Burges C, Somla A (eds) Advances in kernel methods: support vector machine. MIT Press, Cambrige, MA, pp 255-268.