Probability output of multi-class support vector machines*

XIN Dong(忻 栋)[†], WU Zhao-hui(吴朝晖), PAN Yun-he(潘云鹤)
(*Department of Computer Science & Engineering, Zhejiang University, Hangzhou 310027, China*)
[†] E-mail: xindong@cs.zju.edu.cn

Received July 8, 2001; revision accepted Oct.15, 2001

Abstract: A novel approach to interpret the outputs of multi-class support vector machines is proposed in this paper. Using the geometrical interpretation of the classifying heperplane and the distance of the pattern from the hyperplane, one can calculate the posterior probability in binary classification case. This paper focuses on the probability output in multi-class phase where both the one-against-one and one-against-rest strategies are considered. Experiment on the speaker verification showed that this method has high performance.

Key words: Support vector machines(SVM), Posterior probability, Multi-class, Speaker verification
Document code: A **CLC number:** TP183

INTRODUCTION

The support vector machine (Burges, 1998) has recently been introduced as a new technique for solving various function estimation problems. The outputs of support vector machine in classification are not only qualitative, but also quantitative. Using the geometrical interpretation of the classifying heperplane and the distance of the pattern from the hyperplane, one can calculate the posterior probability in binary classification case. In many pattern recognition applications such as speaker verification, such probability output will be used to construct a rejection threshold for the vectors belonging to other classes.

Many researchers proposed to solve this problem. Vapnik (1999) suggested a method for mapping the output of SVMs to probabilities by decomposing the feature space. Hastie and Tibshirani (1996) fitted probabilities to the output of an SVM by using Gaussians to the class-conditional densities $p(f \mid y = 1)$ and $p(f \mid y = -1)$. Another method proposed by Platt (1999) trains the parameters of an additional sigmoid function to map the SVM outputs into probabilities. The results were promising, but they did not extended their method to multi-class phase.

In this paper, we propose the approach to construct probability output of multi-class case. The sigmoid function is used to estimate the probability output in binary classification. This estimation can be substituted by any other forms of posterior probability in the combination architecture.

OVERVIEW OF SUPPORT VECTOR MACHINES

The main idea of the binary classification support vector approach (Burges, 1998) is to construct a hyperplane to separate the two classes (labeled $y \in \{-1, 1\}$), and let the decision function be:

$$f(\boldsymbol{x}) = \operatorname{sign}(\boldsymbol{w} \cdot \boldsymbol{x} + b) \qquad (1)$$

Maximization of the margin (the distance between the hyperplane and the nearest point) leads to the following optimization problem, minimize:

$$\varphi(\boldsymbol{w}, \xi) = \frac{1}{2}(\boldsymbol{w} \cdot \boldsymbol{w}) + C\sum_{i=1}^{l} \xi_i \qquad (2)$$

* Project supported by the National High Technology Research & Development Programme (863) of China (No.2001AA4180) and Zhejiang Provincial Natural Science Foundation for Young Scientist (No.RC01058).

With constraints

$$y_i((\mathbf{w}\cdot\mathbf{x}_i)+b)\geqslant 1-\xi_i,\ i=1,2,\cdots,l$$
$$\xi_i\geqslant 0,\ i=1,2,\cdots,l$$

The dual solution to this problem is: maximize the quadratic from Eq.(3) under constraints Eq.(4).

$$W(\alpha)=\sum_{i=1}^{l}\alpha_i-\frac{1}{2}\sum_{i,j=1}^{l}y_iy_j\alpha_i\alpha_jK(\mathbf{x}_i,\mathbf{x}_j) \tag{3}$$

With constraints

$$0\leqslant\alpha_i\leqslant C,\ i=1,2,\cdots,l$$
$$\sum_{i=1}^{l}\alpha_iy_i=1 \tag{4}$$

Where $K(\mathbf{x}_i,\mathbf{x}_j)$ is kernel function. Giving the decision function:

$$f(\mathbf{x})=\operatorname{sign}\Big[\Big(\sum_{i=1}^{l}\alpha_iy_iK(\mathbf{x},\mathbf{x}_i)\Big)+b\Big] \tag{5}$$

CONVERT SVM OUTPUT TO POSTERIOR PROBABILITY

The outputs of SVM classifiers can be divided into two parts (Kwok, 1999; Madevska-Bogdanova et al, 2000): the sign and the numerical value. The sign indicates the qualitative description of classification and the numerical value can be used to obtain the posterior probability.

Analytic geometry can be used to provide an explanation of the meaning of the outputs of the SVM classifiers. The classification of the SVM is given by Eq.(1), while the activation is:

$$g(\mathbf{x})=\mathbf{w}\cdot\mathbf{x}+b \tag{6}$$

Where $\mathbf{x}$ is an input vector.

The relation is invariant under a positive rescaling of the argument inside Eq. (1). Thus a canonical hyperplane is defined so that $|g(\mathbf{x})|=1$ for the closest points (support vectors).

It is clear that for a given hyperplane $g(\mathbf{x})=0$, and for a vector $\mathbf{x}$ that does not belong to the hyperplane, we have:

$$g(\mathbf{x})=\pm d\,\|\mathbf{w}\| \tag{7}$$

Where, d is the distance from the point $\mathbf{x}$ to the given hyperplane. The different signs determine the side of the hyperplane for the vector $\mathbf{x}$. So we can see that the output $g(\mathbf{x})$ of the SVM is actually the multiplication of the norm of the vector $\mathbf{w}$ and the distance from the chosen hyperplane, and analogously

$$d_x=\frac{g(\mathbf{x})}{\|\mathbf{w}\|} \tag{8}$$

And the margin between the canonical hyperplane and the closest points (support vectors) is:

$$d_{\text{sv}}=\frac{1}{\|\mathbf{w}\|} \tag{9}$$

Clearly, the ratio of d_x to d_{sv} is $g(\mathbf{x})$. Using the rate of the distance, we propose a modification of the SVM outputs. We convert the output of the SVM to the posterior probability. This estimate is applied to a sigmoid function to yield:

$$P(C_{+1}|\mathbf{x})=\frac{1}{1+\mathrm{e}^{-g(\mathbf{x})}} \tag{10}$$

And

$$P(C_{-1}|\mathbf{x})=\frac{1}{1+\mathrm{e}^{g(\mathbf{x})}} \tag{11}$$

The sigmoid transforms correct classifications to a quantity roughly equal to one and misclassification to a quantity roughly equal to zero.

POSTERIOR PROBABILITY IN MULTICLASS SVM

The following strategies were applied to build N classes classifiers utilizing binary SVM classifiers.

1. One-against-rest classifiers

In this method, N different classifiers are constructed, one classifier for each class. Here the i^{th} classifier is trained on the whole training data set in order to classify the members of class i against the rest. For this, the training samples have to be relabeled: Members of the i^{th} class are labeled to 1; members of the other classes are labeled to -1. In the classification phase, the classifier defines the estimated posterior probability of the current input vector as:

$$P(C_i \mid \boldsymbol{x}) = \frac{P_{\text{iar}}(C_i \mid \boldsymbol{x})}{\sum_{j=1}^{N} P_{\text{jar}}(C_j \mid \boldsymbol{x})} \quad (12)$$

Where, $P_{\text{iar}}(C_i \mid \boldsymbol{x})$ is the probability output of the binary SVM separating i^{th} class and the rest.

2. One-against-one classifiers

In this method, each possible pair of classes of a binary classifier is calculated. Each classifier is trained on a subset of the training examples of the two involved classes. As for the one-against-rest strategy, the training sets have to be re-labeled; all $N(N-1)/2$ binary classifiers are combined to estimate the final output.

$$P(C_i \mid x) = \frac{\sum_{j=1, j\neq i}^{N} P_{\text{iaj}}(C_i \mid \boldsymbol{x})}{\sum_{k=1}^{N}\left(\sum_{j=1, j\neq k}^{N} P_{\text{kaj}}(C_k \mid \boldsymbol{x})\right)} \quad (13)$$

Where, $P_{\text{iaj}}(C_i \mid \boldsymbol{x})$ is the probability output of the binary SVM for i^{th} class and j^{th} class.

3. Refining the one-against-one probability

As we will see later in the experiment, Eq.(13) does not achieve satisfying results. One can see that each hyperplane in one-against-one classifier is trained by two classes, while the hyperplanes within one-against-rest classifier are trained by all classes. Thus the probability output of a binary SVM in one-against-one strategy is based on the two classes under training. It should be refined as follows:

$$P_{\text{iaj}}'(C_i \mid \boldsymbol{x}) = P_{\text{iaj}}(C_i \mid \boldsymbol{x}) \cdot P_r(C_i \cup C_j \mid \boldsymbol{x}) \quad (14)$$

Where, $P_r(C_i \cup C_j \mid \boldsymbol{x})$ means the probability of vector $\boldsymbol{x}$ belonging to class $C_i \cup C_j$. Assuming

$$P_r(C_i \cup C_j \mid \boldsymbol{x}) = P_r(C_i \mid \boldsymbol{x}) + P_r(C_j \mid \boldsymbol{x}) \quad (15)$$

Thus

$$P_{\text{iaj}}'(C_i \mid \boldsymbol{x}) = P_{\text{iaj}}(C_i \mid \boldsymbol{x}) \cdot (P_r(C_i \mid \boldsymbol{x}) + P_r(C_j \mid \boldsymbol{x})) \quad (16)$$

Now we show how to estimate the value of $P_r(C_i \mid \boldsymbol{x})$. Every class can be trained individually using the one-class SVM proposed by Scholkopf et al.(1999). The initial output of this method is similar to that in Eq.(5), the probability estimation can be achieved using Eq.(10). Detailed introduction is omitted here.

EXPERIMENT ON SPEAKER VERIFICATION

Text-independent speaker verification (Compbell et al, 1999; Wan et al, 2000) is implemented by modeling each speaker with an individual class. Using the multi-classification strategy, the speakers' model is calculated in a speaker verification system. An individual (the claimant) claims a certain identity. The model stored in the system is then used to determine whether the utterance was indeed made by the user. A probability for the utterance is computed from the model and compared to a threshold to determine the claimant's validity. In the test phase, a sequence of test vectors is available for each user. The score for an utterance X is computed as the geometrical mean of the values for each acoustic feature vector $\boldsymbol{x}_k$, $(k = 1, \cdots, K)$. Using the method presented in this work, the posterior probability of utterance X as speaker i is defined as:

$$P(C_i \mid \boldsymbol{X}) = \prod_{k=1}^{K} \frac{P(C_i \mid \boldsymbol{x_k})}{\sum_{j=1}^{N} P(C_j \mid \boldsymbol{x_k})} \quad (17)$$

Our method is comparable to the traditional multi-class SVM output where the value of $P'(C_i \mid x_k)$ is binary (1 or 0), representing whether the vector $\boldsymbol{x}_k$ belongs to class C_i or not. And in this case, Eq.(17) was modified as:

$$P'(C_i \mid \boldsymbol{X}) = \frac{\sum_{k=1}^{K} P'(C_i \mid \boldsymbol{x_k})}{K} \quad (18)$$

The experiments were performed on YOHO (Campbell, 1995) database. The features were derived using 12th order LPC analysis and deltas (making up a twenty four dimensional feature space） on a 30 milliseconds frame every 10 milliseconds. In order to construct a small data set for training， the training data for each speaker was converged to two hundred centroids using k-means clustering algorithm. The RBF kernel was used with $\sigma = 0.5$. The first 50 speakers （labeled 101 to 154） were selected in our test.

We labeled binary output of one-against-rest SVM， binary output of one-against-one SVM， probability output of one-against-rest SVM， probability output of one-against-one SVM and refined probability output of one-against-one SVM as BOAR， BOAO， POAR， POAO and RPOAO， respectively. The experiment results are shown below.

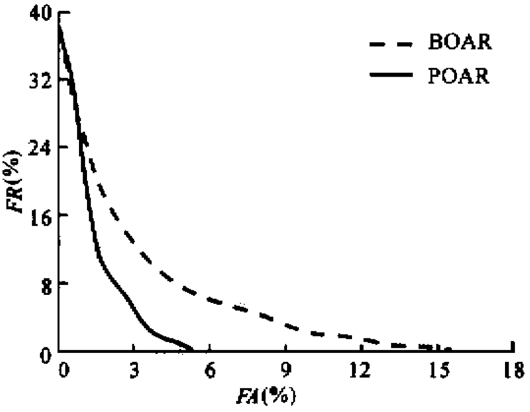

Fig.1　*FA-FR* curves of BOAR and POAR

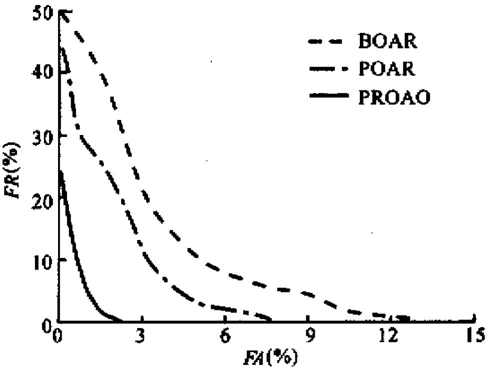

Fig.2　*FA-FR* curves of BOAO， POAO and RPOAO

Fig. 1 and Fig. 2 show the false reject （FR） and false accept（FA） curves for one-against-rest classifier and one-against-one classifier， respectively. One can see the probability output has better qerformance. In one-against-rest tests， the equal error rates （false reject = false accept） were 6.06% for BOAR and 3.35% for POAR. In one-against-one test， the equal error rates are 6.75%， 4.55% and 1.58% for BOAO， POAO and RPOAO.

CONCLUSIONS

Our proposed new approach for extracting probabilities from multi-class SVM outputs is useful for classification post processing. Experiments on speaker verification are shown in this paper. Future work will concentrate on comparing the different probability outputs of binary SVM in the multi-class probability architecture， and the modification of combining rules.

References

Burges，C.， 1998. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*， **2**:955-974.

Campbell， J.， 1995. Testing with the YOHO CD-ROM Voice Verification Corpus， Proc. ICASSP: p.341 – 344.

Compbell， W. M.， Assaleh， K. T.， 1999. Polynomial Classifier Techniques for Speaker Verification， Proc. ICASSP: p.321 – 324.

Hastie， T.， Tibshirani， R.， 1996. Classification by pairwise coupling， Technical report， Stanford University and University of Toronto. http://www-stat.satnford.edu/trevor/papers/2class.ps.

Kwok， J.T.Y.， 1999， Moderating the outputs of support vector machine classifiers. *Neural Networks*， *IEEE Transactions on*， **5**:1018 – 1031.

Madevska-Bogdanova， A.， Nikolic， D.， 2000， A new approach of modifying SVM outputs， Proceedings of the IEEE-INNS-ENNS International Joint Conference. **6**:395 – 398.

Platt， J.， 1999， Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods， *In*: Advances in Large Margin Classifiers， MIT Press.

Scholkopf， B.， Platt， J.， Shawe-Taylor， J.， Smola， A. J.， Williamson， R.C.， 1999. Estimating the support of a high-dimensional distribution， Microsoft Research Corporation Technical Report: MSR-TR-99-87.

Vapnik， V. N.， 1999. An overview of statistical learning theory. *Neural Networks*， *IEEE Transactions*， **5**:988 – 999.

Wan. V.， Campbell， W. M.， 2000. Support Vector Machines for Speaker Verification and Identification， Neural Networks for Signal Processing X. p.775 – 784.