Text-independent speaker recognition using support vector machine

HOU Fenglei, WANG Bingxi
NO. 306, P.O. Box 1001, Zhengzhou 450002, Henan, P.R. China
houfl@263.net, bingxiwang@263.net

ABSTRACT

Support vector machine (SVM) is an important learning method of statistical learning theory, and is also a powerful tool for pattern recognition problems. This paper studies the speaker identification and verification problem using support vector machine, and presents a SVM training method on large-scale samples according to the speech signal. A text-independent speaker recognition system based on SVMs was implemented and the results show good performance.

Key words:speaker identification, speaker verification, support vector machine

1. INTRODUCTION

Speaker recognition consists of speaker identification and speaker verification. The goal of speaker identification is to determine which speaker is present based on the individual's utterance. It's different from speaker verification, the goal of which is to verify the speaker's claimed identity based on his or her utterance. It has been widely used in the banking, security, defense and information retrieving. Most current speaker identification systems are either based on Bayes decision (like gaussian mixture model) or on neural network classifier (like radial basis function neural network). The shortcoming of them is that they need cross validation to avoid over training on limited amount of training data [3]. The paper presents a novel approach on speaker identification using support vector machines (SVMs). SVM is a learning method based on the statistical learning theory (SLT). SLT is a special theory on machine learning in the case of limited learning data. SLT is based on a set of theory to form a unit frame for limited sample learning problems. It can contain many current learning methods, and solve many problems, such as construction selection of neural networks, local minimum, and overfitting. SVM shows many good properties over other methods in solving limited samples problems and non-linear high dimension pattern recognition problems. SLT and SVM are becoming the new study hot area after neural network, and will further develop the machine learning theory and technology. [1][2]

2.SUPPORT VECTOR MACHINE

2.1 OptimalSeparating Hyperplane

SVM is developed from the optimal separating hyperplane in the linear separable case. The simple binary classification problem is showed as follows: Given training set $S=\{(x_i, y_i)\}_i^l$, samples in the set are $x_i \in IR^d$ (d is the dimension of the input space), which belong to one of the two classes labeled as $y_i \in \{-1,+1\}$. The goal is to establish the equation of a hyperplane that divides

0-7803-7010-4/01/$10.00©2001 IEEE.

S completely (no misclassification), i.e. leaving all the points of the same class on the same side while maximizing the distance between the two classes and the hyperplane. This idea is illustrated at Fig.1. In the figure, the solid points and the hollow points represent two classes. The lower bound of the minimum distance between points of different classes is named *margin*. The OSH is the hyperplane that can completely separate the two classed while the margin is maximized. The maximization of the margin can be viewed as the control of generalization ability. The larger the margin is, the better the generalization ability is, which is the core idea of the SVM. In SLT it is point out that to make the separating margin maximum is to minimize the upper bound of VC dimension, which implements the selection of the complexity of functions in structure risk minimization (SRM) criteria.

In the linearly non-separable case, a trade-off between the minimum error classification and the maximum margin should be taken into consider, which is the generalized OSH. But firstly the misclassification penalty parameter C should be set. Where C > 0 is a constant that controls the degree of penalty to the misclassification samples.

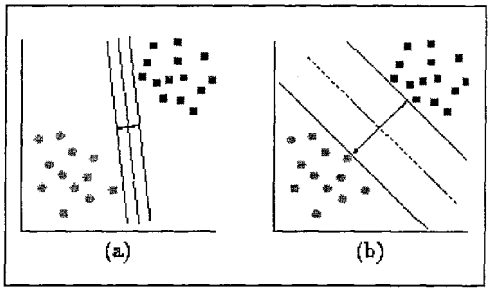

Fig.1 (a) OSH with smaller margin (b) OSH with larger margin
((b) has a better generalization ability)

It can be proved that in this case the linear classifier that solve the problem is:

$$f(\mathbf{x}) = \mathrm{sgn}\{\sum_{i=1}^{l} \alpha_i y_i \mathbf{x}^T \mathbf{x}_i + b\} \qquad (1)$$

where is the solve of next equation:

$$\text{Minimize} \qquad W(\mathbf{a}) = -\mathbf{a}^T \mathbf{1} + \frac{1}{2}\mathbf{a}^T \mathbf{Q}\mathbf{a} \qquad (2)$$

$$\text{subject to} \qquad \mathbf{a}^T y = 0$$

$$0 \le \mathbf{a} \le C\mathbf{1}$$

Where $(\mathbf{a})_i = \alpha_i$, $(\mathbf{1})_i = 1$, $(\mathbf{Q})_{ij} = y_i y_j \mathbf{x}_i^T \mathbf{x}_j$. It can be proved that there are only a few α_is are not equal to 0, and the corresponding samples are support vectors. So the sum in Eq. (1) operates only on support vectors. b is the threshold value of classification, and can be calculated by any one of the support vectors.

2.2 Support Vector Machine

In real occasions, linear classifiers cannot solve most classification problems. In the case of non-linear classification problems, nonlinear transform is often used to transform the low dimension space to high dimension space, and convert the non-linear problem into a linear problem. Such a transform may be very complex even almost infeasible. Note that in the dual

problem, calculation of either the optimal function Eq. (2) or the classification function Eq. (1) involves only the inner product $(x_i \cdot x_j)$ of training samples. So only the inner production is needed for transformation to the high dimension space, which is implemented by the functions in the low dimension space, without knowing the form of transformation. According to the relevant theory of fonctionelle, if a kernel function $K(x_i \cdot x_j)$ satisfies the Mercer condition, it has a corresponding inner production in some transform space. Such a condition is not difficult to satisfy. Therefore, adopting some proper inner production function $K(\mathbf{x}_i \cdot \mathbf{x}_j)$ can perform linear classification after some non-linear transformation, without any computation complexity incensement. In this case, the $\mathbf{Q}$ changes to $(\mathbf{Q})_{ij} = y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$, and the corresponding classification function changes to

$$f(\mathbf{x}) = \mathrm{sgn}\{\sum_{i=1}^{l} \alpha_i y_i K(\mathbf{x}, \mathbf{x}_i) + b\} \qquad (3)$$

This is the support vector machine.

To summarize up, the support vector machine first non-linearly transforms from low dimension space to a high dimension space by inner production, then finds the OSH in this space.

Different kernel functions form different SVM algorithms. Here are three kind of Kernel function that are often used:

Classifier Type	Kernel function
Polynomial	$K(\mathbf{x}, \mathbf{x}_i) = [(\mathbf{x} \cdot \mathbf{x}_i) + 1]^q$
Radial basis function (RBF)	$K(\mathbf{x}, \mathbf{x}_i) = \exp\{-\frlac{\lvert \mathbf{x} - \mathbf{x}_i \rvert^2}{\sigma^2}\}$
Sigmoid function	$K(\mathbf{x}, \mathbf{x}_i) = \tanh[v(\mathbf{x} \cdot \mathbf{x}_i) + c]$

Table 1. Kernel functions

Here, SVM can be viewed as new algorithms for training Polynomial, RBF and neural networks.

3. TRAINING ALGORITHM OF SVM

Training SVMs involves the optimization of a quadratic Lagrangian and thus techniques from quadratic programming (QP) are most applicable. Although there are certain QP packages that are readily applicable, they have the disadvantage that the kernel matrix $\mathbf{Q}$ is stored in memory, which is unsuitable for larger datasets. An alternative would be to recompute $\mathbf{Q}$ every time it is needed. But this becomes prohibitively expensive, if $\mathbf{Q}$ is used often. In speech processing, the training speech samples are usually large, so it is very important to find an approach to train the SVMs efficiently [5]. Osuna et al [6]. presents a decomposition idea to train SVMs, which splits QP problem into an inactive part - N and active part - B, called 'wo rking set'. It decompositions a

large QP problem into a set of small QP sub-problems, and solves the QP problem through solving QP sub-problems and changing components in "working set" continuously. Joachims [4] presents ideas of selecting working set, "shrinking" and caching to simplify the problem and train SVMs faster. Platt's [7] sequential minimal optimization (SMO) can be viewed as a special case that the size of the "working set" is 2, i.e. deal with only 2 components at each iteration. The advantage is that it can directly get the analytic solution of the QP sub-problem. Combining the upper methods such as the caching and SMO together make training SVMs more efficient, which is adopted in this paper.

4. SPEAKER RECOGNITION SYSTEM

4.1 Speaker identification

Every SVM is trained to classify two speakers because that SVM separates two classes better. So every SVM performs as a binary classifier. When there are N enrolled speakers in the set, the speaker identification becomes a N-way classification problem. There are N*(N-1)/2 SVMs needed altogether because each SVM classifies two. In training phase, speech features from two speakers are labeled as "+1" or " -1" separately. After all the SVMs are trained using the algorithm described before, the support vectors (SV) of each SVM are stored and then the speaker models are constructed. In test phase, the test speech features are input into one SVM, using Eq. (1) to find which class each feature belongs to. After all the features have input to the SVM, the class taking the largest number of test features is identified, which can be called as "winner". To identify the speaker from the N speakers, the N speakers should arrange into pairs. Each pair is corresponded to a SVM classifier. The "winners" of each pair then arrange into pairs again. After several iterations, there is only one "winner" at last, which is the identification result.

4.2 Speaker verification

The speaker verification is to verify the speaker based on the claimed speaker's voice. The training speech feature vectors can be this: Class 1 contains the speech features of the claimed speaker; class 2 contains the speech features from many other speakers. If the number of test features belongs to the claimed speaker is large enough than those of other speakers, the affirmation is accepted, otherwise, rejected. See Eq. (4):

$$\frac{\sum_{i}(f(x_i)=1)}{\sum_{j}(f(x_j)=-1)} > T \quad , \qquad i,j=1,\ldots,n \tag{4}$$

Where x_i is test feature, n is the number of test features, T is the threshold.

There is a problem that the other speakers' speech features usually much greater than the speaker's. In this case the SVM may be ill trained. One method to solve this problem is to use vector quantification technique to compress the number of the other speakers' speech feature s before training.

5. EXPERIMENTS AND RESULTS

5.1 Feature extraction

After A/D conversion, the analog speech signal converts to 8kHz, 16bit digital signal. A 32ms

Hamming window is applied to the speech every 10ms. A fifteenth order linear predictive analysis is performed for each speech frame from which fifteen cepstral coefficients (LPCC) are derived.

5.2 Training of SVM

In fact the training of the SVM is a non-linear inseparable problem. So it is necessary to preliminarily choose the penalty of misclassification constant C. Fig.2 shows that the average ratio of the number of support vectors (SV) over all training vectors using different Kernel function and C. The larger the C is, the small the ratio of SVs is, vice versa. Selecting C properly is very important to the system because that the ratio is the upper bound of misclassification error rate [1]. Considering the trade-off between the misclassification and learning time, C=1000, RBF kernel is chosen for each SVM.

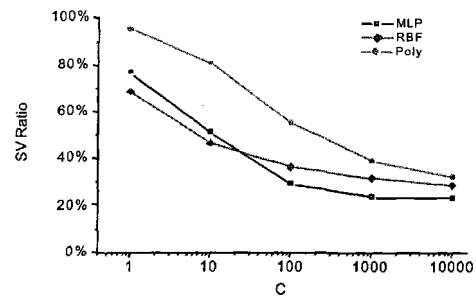

Fig.2 relationship between SV ratio and C

5.3 Speaker identification experiment

For each 30 speakers (23 male, 7 female), one approximately 10-15 second utterance is provided for training. The test utterances are about 2 seconds in length. These utterances are text free so it is a text-independent case. A close-set speaker identification experiments is implemented using the upper database. A MLP neural network based experiment is also done for comparison. The identification results are showed in Table 2.

	SVM	MLP
Correction rate	91.4%	90.8%

Table 2. Identification results

5.4 Speaker verification experiments

The speech features of each claimed speaker are extracted from 10-15 second utterance, and the features of the each speaker in the cohort set are compressed using VQ. Selecting the codebook size of each speaker and number of speakers in cohort set properly to make the number of feature vectors of the cohort set and the claimed speaker approximatively equal. The verification EER of the test is 2.31%, which is showed in Fig. 3.

6. CONCLUSION

Unlike traditional training methods based on empirical risk minimization (ERM), the training of SVM is based on SRM, which has better generalizaion ability for classification. A speaker recognition system based on SVM was implemented and experiments on speaker identification and speaker verification were made. The results showed that SVM has good performance, and outperforms neural networks.

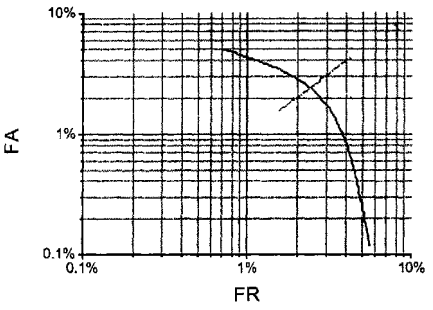

Fig.3 Verification error rates

7. REFERENCES

[1] Vapnik V N, The Nature of Statistical Learning Theory, Second Edition, NY: Springer-Verlag, 1999

[2] Burges C JC, A tutorial on supportvector machines for pattern recognition. Data Mining and Knowledge Discovery, 2 (2) 1998

[3] Schmidt, M, Identifying speakers with support vector networks, Proceedings of Interface '96, Sydney, 1996.

[4] T. Joachims, Making large-Scale SVM Learning Practical. Advances in Kernel Methods - Support Vector Learning, B. Schölkopf and C. Burges and A. Smola (ed.), MIT Press, 1999

[5] Campbell C, Algorithmic approaches to training support vector machines: A survey, Proceedings of ICML2000, page 8, 2000

[6] E. Osuna, R. Freund, and F. Girosi, Training support vector machines: An application to face detection. Proceedings of CVPR'97, Puerto Rico, 1997

[7] J. Platt, Fast training of support vector machines using sequential minimal optimization. In B. Schölkopf, C. J. C. Burges, and A. J. Smola, editors, Advances in Kernel Methods --- Support Vector Learning, pages 185-208, Cambridge, MA, 1999. MIT Press