Mixture of Support Vector Machines for Text-independent Speaker Recognition

Zhenchun Lei, Yingchun Yang and Zhaohui Wu

College of Computer Science and Technology
Zhejiang University, Hangzhou, P.R.China
{leizhch,yyc,wzh}@zju.edu.cn

Abstract

In this paper, the mixture of support vector machines is proposed and applied to text-independent speaker recognition. The mixture of experts is used and is implemented by the divide-and-conquer approach. The purpose of adopting this idea is to deal with the large scale speech data and improve the performance of speaker recognition. The principle is to train several parallel SVMs on the subsets of the whole dataset and then combine them in the distance or probabilistic fashion. The experiments have been run on the YOHO database, and the results show that the mixture model is superior to the basic Gaussian mixture model.

10.21437/Interspeech.2005-638

1. Introduction

Support vector machines (SVMs) [1, 2] have got more attention in machine learning recently for its superior performance. SVM is based on the principle of structural risk minimization. Experimental results indicate that SVMs can achieve a generalization performance that is greater than or equal to other classifiers, while requiring significantly less training data. Another key property of SVMs is that training SVMs is equivalent to solving a linearly constrained quadratic programming problem so that the solution is always unique and globally optimal.

SVMs have also got more attention in speaker recognition [3] and speech recognition recently. Most of these methods are to construct a superior kernel function, which map the utterances having different length into the fixed size vectors, such as the fisher kernel [4], etc. This class of method is utterance-based, and constructing a superior kernel function for utterances can be difficult and still a challenge. Like the generative models, the SVM can be used in a scoring fashion. Every frame is scored by the SVM and the decision was made based on the accumulated score over the entire utterance [5, 6]. This class of method is frame-based.

In this paper, we will propose the mixture of support vector machines for text-independent speaker recognition, and the basic idea is the mixture of experts (ME) architecture [7, 8]. The ME consists of a set of expert networks and a gating network that cooperate with each other to solve a complex problem. The expert networks are sued to solve different input regions which are decomposed form the whole input space, and the outputs of the expert networks are combined by the gating network to obtain the solution of the problem. The motivation of the ME is that individual expert networks can focus on specific regions and attack them well. The divide-and-conquer principle is used to attack the complex problem by dividing it into simpler problems.

This idea have been applied to SVMs early, Collobert proposed a parallel mixture of SVMs aimed at solving problem having very large scale examples [9]. The reason for adopting this idea is twofold. First, they can deal with the large scale speech data using SVMs, and second, they can improve the recognition performance. Like GMM model, the score of every frame is decided by all components. Two type of scoring method were developed according to the distance and the probability used in the VQ and the GMM respectively. Our experiments were tested on the YOHO database and in text-independent speaker identification case.

This paper is organized in the following way: In section 2 we review the SVMs theory briefly and the method in speaker recognition using SVMs. In section 3, we explain the detail of mixture model. Section 4 presents the experimental results on the YOHO database. Finally, section 5 is devoted to the main conclusions and the future work.

2. Support Vector Machine for Speaker Recognition

2.1. Support Vector Machine Theory

SVM theory [1, 2] is mainly from the problem of binary classification, and its main idea can be concluded as the following two points: it constructs a nonlinear kernel function to present an inner product of feature space. It implements the structural risk minimization principle in statistical learning theory by generalizing optimal hyper-plane with maximum margin between the two classes.

The hyperplane is defined by $x \cdot w + b = 0$ that leaves the maximum margin between the two classes. It can been shown that maximizing the margin is equivalent to minimizing an upper bound on the generalization error of the classifier, providing a very strong theoretical motivation for the technique. The vector w that maximizes the margin can be shown to have the form:

$$w = \sum_{i=1}^{l} \alpha_i y_i x_i \tag{1}$$

where the parameters α_i are found by solving the following quadratic programming (QP) problem.

$$\max_{\alpha}(\sum_{i} \alpha_i + \sum_{i,j} \alpha_i \alpha_j y_i y_j x_i \cdot x_j) \tag{2}$$

subject to:

$$\sum_i \alpha_i y_i = 0$$
$$0 \le \alpha_i \le C \tag{3}$$

The main feature of the SVM is that its target functions attempts to minimize the number of errors made on the training set while simultaneously maximizing the margin between the individual classes. This is an effective prior for avoiding over-fitting, which results in a sparse model dependent only on a subset of kernel functions.

The extension to non-linear boundaries is acquired through the use of kernels that satisfy Mercer's condition. The kernels map the original input vector x into a high dimension space of features and then compute a linear separating surface in this new feature space. In practice, the mapping is achieved by replacing the value of dot production between two data points in input space with the value that results when the same dot product is carried out in the feature space. The following is formations:

$$\max_{\alpha}(\sum_i \alpha_i + \sum_{i,j} \alpha_i \alpha_j y_i y_j K(x_i, x_j)) \tag{4}$$

The kernel function K defines the type of decision surface that the machines will build. In our experiments, the radial basis function (RBF) kernel is used and it takes the form:

$$k(x_i, x_j) = \exp\left[-\frac{1}{2}\left(\frac{\left\| x_i - x_j \right\|}{\sigma}\right)^2 \right] \tag{5}$$

where σ is the width of the radial basis function. The use of kernels means that an explicit transformation of the data into the feature space is not required.

2.2. Support Vector Machine for Speaker Recognition

The score of an utterance is computed simply as the arithmetic means of the activation of the SVM for each acoustic feature vector.

The score of an utterance of length N is

$$S = \frac{1}{N}\sum_{i=1}^{N}(w \cdot x_i + b) \tag{6}$$

use the kernel function, the equation is:

$$S = \frac{1}{N}\sum_{i=1}^{N}(\sum_j \alpha_j y_j K(x_j, x_i) + b) \tag{7}$$

After the utterance score has been computed, it is compare to a threshold T in speaker verification. A decision is made according to the rule: if S is greater than T, then accept the speaker, or reject. The equal error rate (EER) was used for the purpose of evaluation in speaker verification task.

In the speaker identification, the classifiers to separate each speaker form all of the others are constructed, and the identity of the speaker is determined from the classifier that yields the largest utterance score.

3. Mixture of Support Vector Machines

3.1. Training

We adopt the multiple experts' idea, and the mixture model is proposed. The idea is used in the previous work to replace the experts by SVMs. The parallel mixture of SVM model has been used for dealing with the large scale data in Collobert's paper [9]. Chakrabartty and Cauwenberghs use SVMs to decide the state transition probabilities in HMM [10]. The divide-and-conquer approach is used for decomposition of a complex prediction problem into simpler local sub-problems [11]. The reason for adopting this idea is twofold. First, they can deal with the large scale speech data using SVMs; second, we want to improve the recognition performance.

We also use the divide-and-conquer strategy, which decompose the global problem into a number of sub-problems according to a division process. We propose to divide the training set using an unsupervised algorithm to cluster the data (k-means), and then train an SVM expert on each subset. The training process is described in figure 1:

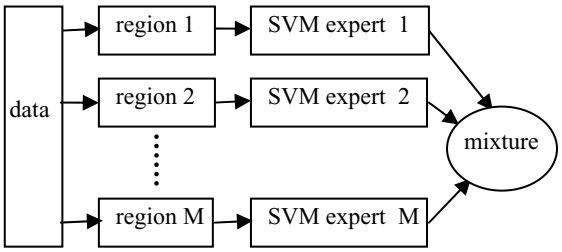

Figure 1: Schematic diagram of training process.

In the training phase, the data set is divided into M subset, and a SVM is trained using each subset as the positive samples while the negative samples are the others speakers' data. So the M SVMs are trained for each speaker, and M hyper-planes are got in some high dimension space. The dividing method used is the k-means clustering algorithm for its simplicity.

We can also explain our methods from another point of view. The VQ and the GMM are the popular methods for text-independent speaker recognition. In the VQ method, each speaker is characterized with several code vectors, and the set of code vectors of each speaker is referred to as that speaker's codebook. A speaker's codebook is trained to minimize the quantization error for the training data from that speaker. In the recognition stage, an input utterance is vector-quantized using the codebook of each reference speaker and the VQ distortion accumulated over the entire input utterance is used to make the recognition decision. In the GMM, the score are got by a probability density function on the distance between the vector and the mean vectors.

The VQ and the GMM models are vector point based. The VQ model sums the distances between the frame vectors and the codebook vectors; The GMM compute the probability density according to the distance between the frame vectors and the mean vectors. We used the hyper-plans instead of the reference vectors, and there three different sides to the previous model: first, we used the distances between the

vector and the hyper-plans, not the vectors; second, the distances were in some higher dimension space which may be not clear; third, the distances have positive and negative distance, and we pursue the maximal distance, not the minimal in the vector point based model.

Like the VQ and the GMM, we can introduce two types of models for scoring in recognition phase as following:

3.2. Distance model

Unlike the VQ model, the positive and negative distances to the hyper-planes are used. For a frame vector, the score is the maximum distance among all the distances to the hyper planes.

In the recognition stage, an input utterance is scored using the SVMs of each reference speaker and the distance accumulated over the entire input utterance is used to make the recognition decision. Denote the sequence of feature vector extracted from the unknown speaker as $X = \{x_1, ..., x_T\}$. The goal is to find the maximum distance from all SVMs. The average distance $\overline{D}$ that results from an utterance is following:

$$\overline{D} = \frac{1}{T}\sum_{i=1}^{T} \max_j (d(x_i, SVM_j)) \tag{8}$$

where d is the output of SVM:

$$d(x_i, SVM_j) = \sum_k (\alpha_{jk} y_{jk} k(x_{jk}, x_i) + b_{jk}) \tag{9}$$

3.3. Probabilistic mixture model

We use the probabilistic outputs of the SVM instead of the probability density function in the GMM. The score is very similar:

For a feature vector x, the mixture probabilistic is defined as:

$$p(x \mid \lambda) = \sum_{i=1}^{M} w_i p_i (d(x, SVM_i)) \tag{10}$$

The score is a weighted linear combination of M support vector machine probabilistic outputs [12, 13]. The class-conditional densities between the margins are exponential, and a parametric form of a sigmoid is used:

$$p(f) = \frac{1}{1 + \exp(Af + B)} \tag{11}$$

This sigmoid model is equivalent to assuming that the output of the SVM is proportional to the log odds of a positive example. The mixture weights are got according to the number of samples in each subset for simplicity.

$$w_i = \frac{\#\ of\ samples\ in\ the\ subset}{\#\ of\ samples\ in\ the\ whole\ set} \tag{12}$$

Usually, the feature vectors of X are assumed independent, so the log-likelihood of a model λ for a sequence of feature vector, $X = \{x_1, ..., x_T\}$, is computed as:

$$\log p(X \mid \lambda) = \sum_{t=1}^{T} \log(p(x_t \mid \lambda)) \tag{13}$$

Generally, the average log-likelihood value is used by dividing $\log p(X \mid \lambda)$ by T.

4. Experiments

4.1. Database

Our experiments were performed using the YOHO database. This database consists of 138 speaker prompted to read combination lock phrases, for example, “29_84_47”. Every speaker has four enrollment sessions with 24 phrases per session and 10 verify sessions with 4 phrases per session. The features are derived from the waveforms using 12th order LPC analysis on a 30 millisecond frame every 10 milliseconds and deltas computed making up a twenty four dimensional feature vector. Mean removal, preemphasis and a hamming window were applied. Energy-based end pointing eliminated non-speech frames.

4.2. Speaker identification

The SVM is constructed to solve the problem of binary classification. For N-class, the general method is to construct N SVMs. The ith SVM will be trained with all of the examples in the ith class with positive labels, and all other examples with negative labels. We refer to SVMs trained in this way as 1-vs-r (one-versus-rest) SVMs. Another method is 1-vs-1 (one-versus-one) SVMs, which construct *K=N(N-1)/2* classifiers and each classifier be trained on only two out of N classes. In our experiments, the 1-vs-r method was adopted.

The 30 speakers, labeled 101 to 132, were used in our experiments and 30 SVMs were trained one speaker from all other 29 speakers. In training phase, 30 SVM sets were trained discriminating one speaker from all other 29 speakers. And for every SVM in the speaker's SVM set, the subset was the positive samples and the other speakers' data were the negative samples. In order to construct a small data set for training, only 100 representative vectors were selected by k-means clustering in the subset, and 100 negative samples selected on every others speaker's data. So for every SVM, the number of positive sample was 100 and the negative sample was 100*29=2900.

Table 1: Performance of the mixture of support vector machines for text independent speaker identification experiments on the YOHO database

M	Distance model	probabilistic mixture model	Basic GMM
2	8.6	11.3	30.6
4	5.5	5.8	26.1
8	4.4	3.8	17.2
16	3.7	2.9	11.4
32	3.3	2.4	7.8

Training SVMs rely on quadratic programming optimizers, so it is not easily to large problems. There are some algorithms for this problem and the SMO [14] was used

in this paper. And the SVMs with radial basis function kernel were trained in our experiments.

We also tested the GMM with diagonal covariance matrices on the same data and showed the result here as a reference. Table 1 shows the result.

The table 1 shows that the performances are superior to the GMM at the same number of components. And when the M is greater, the probabilistic model has better performance than the distance model.

4.3. Speaker verification

The same models were used for speaker verification, and the table 2 shows performance using the EER as the results. The condition was the same with the speaker identification.

Table 2: Speaker verification results on the YOHO database using mixture of support vector machines

M	Distance model	probabilistic mixture model
2	8.3	5.8
4	4.8	3.4
8	3.0	2.2
16	1.7	2.1
32	1.5	1.9

Table 2 shows that the distance model is better in speaker verification when M is greater.

5. Conclusions

We proposed the mixture of support vector machines for speaker recognition in this paper. The mixture of experts architecture was adopted and was used to attack a complex problem by dividing it into simpler problems whos solutions are combined to yield a solution to the complex problem. The VQ and the GMM were the popular models in speaker recognition, and we developed the distance model and probabilistic mixture model according to their scoring ideas separately. The experiments on the YOHO database showed that our methods were superior to the basic GMM. In another side, our models were the hyperplane-based instead of the point-based in the VQ and the GMM, which was very attractive in theory: first, the distances to the hyper-plans were used, not to the vectors; second, the distances were in some higher dimension space; third, the distances had positive and negative distance, so we would pursue the maximal distance.

Although the model shows the advantages, there are some problems which still are not solved. First, the training time and the test time are still long for the SVM's characters in nature. Second, the probabilistic mixture mode is simple in our experiments, and we will develop the new ways to improve the performance. There are three main research sides in our future works: the clustering algorithm, the probabilistic output of SVM and the weight algorithm for the components.

6. Acknowledgements

This work is supported by National Natural Science Foundation of P.R.China (60273059), Zhejiang Provincial Natural Science Foundation (M603229) and National Doctoral Subject Foundation (20020335025)

7. References

[1] V.Vapnik, *Statistical Learning Theory*, Johm Wiley and Sons, New York, 1998.

[2] SC.J.C.Burges, “A tutorial on Support Vector Machines for Pattern Recognition,”*Data Mining and Knowledge Discovery,* vol. 2, 1–47, 1998.

[3] M.Schmidt and H.Gish, “Speaker Identification via Support Vector Machies,”*in ICASSP,* 105-108, 1996.

[4] T.S.Jakkola and D.Haussler, “Exploiting generative models in discriminative classifiers,”*In Advances in Neural Information Processing System 11* MTT Press, 1999.

[5] V.Wan, W.M.Campbell “Support Vector Machines for Speaker Verification and Identification,” *in Proc. Neural Nerworks for Signal Processing X,* 775-784, 1999.

[6] Xin Dong and Wu Zhaohui, "Speaker Recognition Using Continuous Density Support Vector Machines", *ELECTRONICS LETTERS* 16th August 2001

[7] R.A.Jacobs, M.A.Jordan, S.J.Nowlan, G.E.Hinton, "Adaptive mixtures of local experts", *Neural Comput.*, 79-87, 1991

[8] M.I.Jordan, R.A.Jacobs, "hierarchical mixture of experts and the EM algorithm", *Neural Computation*, 181-214, 1994

[9] Ronan Collobert, Samy Bengio and Yoshua Bengio, “A Parallel Mixture of SVMs for Very Large Scale Problems,” *Advances in Neural Information Processing Systems, Neural Computation*, 2002.

[10] Shantanu Chakrabartty, and Gert Cauwenberghs. " Forward-Decoding Kernel-Based Phone Recognition" *in NIPS 2002*, 1165-1172, 2002

[11] Rida, A.,Labbi,A. and Pellegrini, C., “Local experts combination throuth density decomposition.” *In International workshop on ai and statistics*, 1999.

[12] J.C.Platt, “Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Mthods.” *In Advances in Large Margin Classifiers*, MIT Press, 1999

[13] J.T.Kwork, "Moderating the Outputs of Support Vector Machine Classifiers", *IEEE TRANSCATIONS ON NEURAL NETWORKS* , Vol.10, No.5, 1018-1031, 1999

[14] J.Platt, “Fast training of SVMs using sequential minimal optimisation,” *Advances in Kernel Methods: Support Vector Learing,* MIT press, Cambridge, MA, 1999

[15] SD.A.Reynolds and R.C.Rose, “Robust text-independent speaker identification using Gaussian mixture speaker models,”*in EEE Trans. Speech Audio Processing,* vol.3, 72–83, 1995.