Ensemble of Support Vector Machine for Text-Independent Speaker Recognition

Zhenchun Lei, Yingchun Yang, Zhaohui Wu

College of Computer Science, Zhejiang University, Hangzhou, China

Summary
In this paper, the ensemble of support vector machines is applied to text-independent speaker recognition, and the bagging-like model and boosting-like model are proposed by adopted the ensemble idea. The purposes of adopting this idea are to deal with the large scale speech data and improve the performance of speaker recognition. The distance-based and probability-based scoring methods are used to score the new utterance. Compared with the conventional vector-based speaker models (Vector Quantization and Gaussian Mixture Model), our method is hyperplan-based. The experiments have been run on the YOHO database, and the results show that our models can get attractive performances.

Key words:
Speaker recognition, support vector machine, ensemble leaning, mixture of experts

Introduction

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

SVM has also got more attention in speaker recognition and speech recognition recently [2]. 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 [3], 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. Each frame is scored by the SVM and the decision was made based on the accumulated score over the entire utterance [4, 5]. This class of method is frame-based.

Manuscript received May 5, 2006.
Manuscript revised May 25, 2006.

In this paper, we will propose the ensemble of support vector machines for text-independent speaker recognition, and the basic ideas are the bagging and boosting. The support vector machine ensemble is also constructed by Hyunchul Kim [6], but it is not feasible for the large-scale speech data. The reasons for adopting these ideas are twofold. First, they can deal with the large scale speech data using SVMs, and second, they can improve the recognition performance. In the scoring phase, two type of scoring method will be developed according to the distance and the probability like the VQ and the GMM respectively.

This paper is organized in the following way: In section 2 we review the SVM theory briefly and the method in speaker recognition using SVM. In section 3, we explain the ensembles of support vector machine. Our ensemble models for speaker recognition are proposed in section 4. Section 5 presents the experimental results on the YOHO database. Finally, section 6 is devoted to the main conclusions.

2. Support Vector Machine

2.1 Support Vector Machine Theory

SVM theory [1] 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:

$$w = \sum_{i=1}^{l} \alpha_i y_i x_i \quad (1)$$

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

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

subject to:

$$\sum_i \alpha_i y_i = 0 \qquad 0 \le \alpha_i \le C \quad (3)$$

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:

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

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:

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

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:

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

use the kernel function, the equation is:

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

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 from all of the others are constructed, and the identity of the speaker is determined from the classifier that yields the largest utterance score.

3. Ensemble Methods

Ensemble techniques have received considerable attention within the recent machine learning literature [7]. The idea to obtain a diverse set of classifiers for a single learning problem and to vote or average their predictions is both simple and powerful, and the obtained accuracy gains often have a sound theoretical foundation. Averaging the predictions of these classifiers helps to reduce the variance and often increase the reliability of the predictions. There are several techniques for obtaining a diverse set of classifiers. The most common technique is to use subsampling to diversify the training sets as in bagging and boosting。

3.1 Bagging

Bagging [8], a sobriquet for bootstrap aggregating, is an ensemble method for improving unstable estimation or classification schemes. It has attracted much attention, probably due to its implementational simplicity and the popularity of the bootstrap methodology. It has been shown that bagging is a smoothing operation which turns out to be advantageous when aiming to improve the predictive performance of regression or classification trees.

3.2 Boosting

The combination of the hypotheses is chosen in a more sophisticated manner for boosting [9]. Unlike bagging which is a parallel ensemble method, boosting methods are sequential ensemble algorithms where the weights are depending on the fitted functions. The intuitive idea is that examples that are misclassified get higher weights in the next iteration, for instance the examples near the decision boundary are usually harder to classify and therefore get high weights after a few iterations. This idea of iterative reweighting of the training is essential to boosting, and a famous algorithm is the AdaBoost algorithm.

4. SVM Ensemble for speaker recognition

4.1 Bagging-like model

We adopt the bagging idea, and use a parallel ensemble method for mixture of SVMs like the mixtures of experts [10]. The parallel mixture of SVM model has been used for dealing with the large scale data in Collobert's paper [11]. The divide-and-conquer approach is used for decomposition of a complex prediction problem into simpler local sub-problems [12]. 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 propose to divide the training set using an unsupervised algorithm to cluster the data (k-means), and then train an SVM on each subset. The training process is described in figure 1:

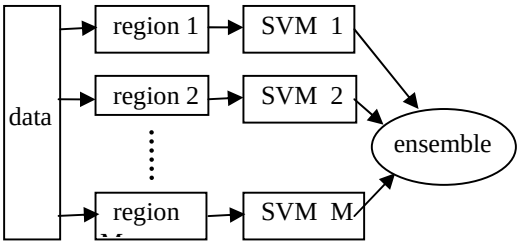

Figure 1: Training Bagging-like model.

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 is the k-means clustering algorithm for its simplicity.

We can also explain our method 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. We used the hyper-plans instead of the reference vectors, and there are 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.

4.2 Boosting-like model

The boosting idea can also be adopted and figure 2 shows the schematic diagram of training processing:

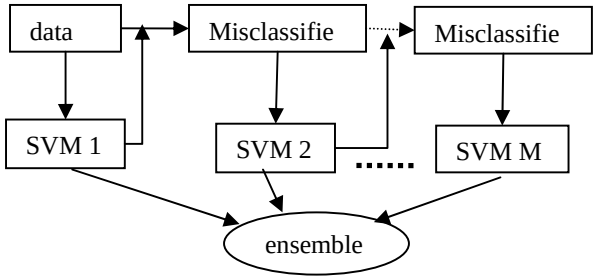

Figure 2: Training Boosting-like model.

The boosting-like model uses a sequential ensemble algorithm instead of the parallel ensemble method in the bagging-like model. The SVM is trained at each iteration using the misclassified data which is chosen based on the performance of the earlier SVMs in the series.

4.3 Scoring

4.3.1 Distance scoring

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:

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

where d is the output of SVM:

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

4.3.2 Probabilistic mixture scoring

We use the probabilistic outputs [13] of the SVM instead of the distance like the probability density function in the GMM. The score is very similar: For a feature vector x, the mixture probabilistic is defined as:

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

The score is a weighted linear combination of M support vector machine probabilistic outputs [14]. The class-conditional densities between the margins are exponential, and a parametric form of a sigmoid is used:

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

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.

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

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:

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

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

5. Experiments

5.1 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 16 order MFCC on a 20 millisecond frame every 10 milliseconds and deltas computed making up a 32 dimensional feature vector. Mean removal, preemphasis and a hamming window were applied. Energy-based end pointing eliminated non-speech frames.

5.2 Speaker verification

The SVM is constructed to solve the problem of binary classification, and the one-vs-others is used for the N-class problem. Training SVM relies on quadratic programming optimizers and the SMO [15] algorithm is used in our experiments. The kernel function is the radial basis function.

The whole databases were divided into two parts. The first parts of speakers, labeled 101 to 174, were trained respectively on the same imposters who labeled 175 to 277. 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.

There are some score normalization methods for speaker verification, and we used the cohort approach [16], which uses a set of cohort speaker who are close to the target speaker. The size of the cohort in our experiments is 1. Table 1 shows the Equal Error Ratio (EER) and figure 3 shows the DET curves with different number of components in bagging-like model.

Table 1: EER for text independent speaker verification

M	Bagging-like model		Boosting-like model	
	No cohort	cohort	No cohort	cohort
1	5.43	2.46	5.43	2.46
2	4.74	1.56	6.32	1.77
4	3.74	1.12	6.67	1.01
8	3.51	0.79	7.39	0.85

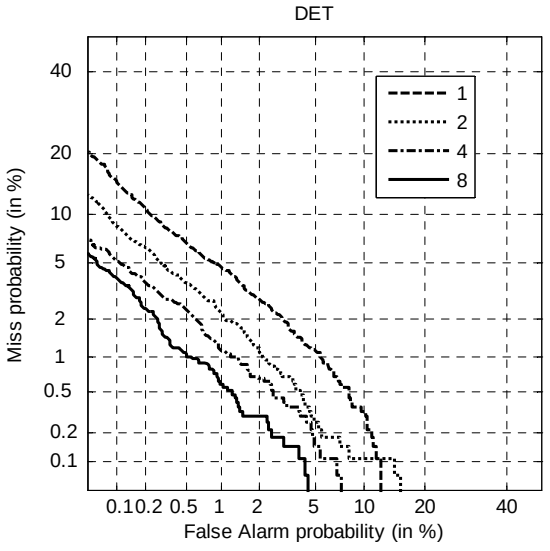

Fig 3: DET curves for speaker verification using bagging-based model

We can see that the performance of bagging-based model is better than boosting-based model's. Table 2 shows that the performance of two type of scoring, and we can see that the probabilistic scoring is better than the distance scoring slightly.

Table 2: Comparisons of scoring type using bagging-like model for speaker verification

M	Distance scoring (%)	Probabilistic scoring (%)
1	2.46	2.28
2	1.56	1.43
4	1.12	1.06
8	0.79	0.74

5.3 Speaker identification

The same models were used for speaker identification, and the table 3 shows performance using the error ratio (ER) as the results. The condition was the same with the speaker verification.

Table 3: Error ratio for speaker identification

M	Bagging-like model (%)	Boosting-like model (%)
1	11.5	11.5
2	7.90	8.2
4	5.29	6.67
8	4.67	6.27

Like speaker verification, the performance of bagging-based model is also better than boosting-based model's for speaker identification.

6. Conclusions

We proposed two support vector machine ensemble models for text-independent speaker recognition in this paper. The bagging and boosting idea are adopted and were used to attack a complex problem by dividing it into simpler problems whose 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-based and probabilistic-based scoring methods by using their ideas separately. 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.

Acknowledgments

This work is supported by National Science Fund for Distinguished Young Scholars60525202, Program for New Century Excellent Talents in University NCET-04-0545 and Key Program of Natural Science Foundation of China 60533040.

References

[1] V.Vapnik, Statistical Learning Theory, Johm Wiley and Sons, New York, 1998.
[2] M.Schmidt and H.Gish, “Speaker Identification via Support Vector Machies,”in ICASSP, 105-108, 1996.
[3] T.S.Jakkola and D.Haussler, “Exploiting generative models in discriminative classifiers,”In Advances in Neural Information Processing System 11 MTT Press, 1999.
[4] V.Wan, W.M.Campbell “Support Vector Machines for Speaker Verification and Identification,” in Proc. Neural Nerworks for Signal Processing X, 775-784, 1999.
[5] Xin Dong and Wu Zhaohui, "Speaker Recognition Using Continuous Density Support Vector Machines", ELECTRONICS LETTERS 16th August 2001
[6] Hyunchul Kim, etc. Constructing support vector machine ensemble, Pattern Recognition, vol 36, pp.2757-2767, 2003
[7] Giorgio Valentini, Francesco Masulli, Ensembles of Learning Machines, WIRN VIETRI, pp.3-20, 2002
[8] Breiman, L., Bagging predictors, Machine Learning, 24, pp.123-140, 1996
[9] R. Meir and G. Rätsch, An introduction to boosting and leveraging. In S. Mendelson and A. Smola, editors, Advanced Lectures on Machine Learning, LNCS, pages 119-184. Springer, 2003
[10] R.A.Jacobs, M.A.Jordan, S.J.Nowlan, G.E.Hinton, "Adaptive mixtures of local experts", Neural Comput., 79-87, 1991
[11] 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.
[12] Rida, A.,Labbi,A. and Pellegrini, C., “Local experts combination throuth density decomposition.” In International workshop on ai and statistics, 1999.
[13] J.C.Platt, “Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Mthods.” In Advances in Large Margin Classifiers, MIT Press, 1999
[14] J.T.Kwork, "Moderating the Outputs of Support Vector Machine Classifiers", IEEE TRANSCATIONS ON NEURAL NETWORKS , Vol.10, No.5, 1018-1031, 1999
[15] J.Platt, “Fast training of SVMs using sequential minimal optimisation,” Advances in Kernel Methods: Support Vector Learing, MIT press, Cambridge, MA, 1999
[16] R.Auckenthaler, M..Carey and H.L.Thomas, "Score Normalization for Text-Independent Speaker Verification Systems" Digital Signal Processing 10, pp42-54, 2000