Learning to Boost GMM Based Speaker Verification

Stan Z. Li, Dong Zhang, Chengyuan Ma, Heung-Yeung Shum, and Eric Chang

Microsoft Research China, Beijing Sigma Center, Beijing 100080, China
szli@microsoft.com, http://research.microsoft.com/∼szli

Abstract

The *Gaussian mixture models* (GMM) has proved to be an effective probabilistic model for speaker verification, and has been widely used in most of state-of-the-art systems. In this paper, we introduce a new method for the task: that using AdaBoost learning based on the GMM. The motivation is the following: While a GMM linearly combines a number of Gaussian models according to a set of mixing weights, we believe that there exists a better means of combining individual Gaussian mixture models. The proposed AdaBoost-GMM method is non-parametric in which a selected set of weak classifiers, each constructed based on a single Gaussian model, is optimally combined to form a strong classifier, the optimality being in the sense of maximum margin. Experiments show that the boosted GMM classifier yields 10.81% relative reduction in equal error rate for the same handsets and 11.24% for different handsets, a significant improvement over the baseline adapted GMM system.

1. Introduction

The speaker verification task is essentially a hypothesis testing problem or that of a binary classification between the target-speaker model and impostor model. The *Gaussian mixture models* (GMM) method [8] has proved to be an effective probabilistic model for speaker verification, and has been widely used in most of state-of-the-art systems [8]. Each speaker is characterized by a GMM, and the classification is performed based on the log likelihood ratio (LLR) of the two classes [8].

A GMM as aims to approximate a complex nonlinear distribution using a mixture of simple Gaussian models, each parameterized by its mean vector, covariance matrix and the mixing parameter. These parameters are learned by using, *e.g.* an EM algorithm. In adapted GMM based speaker verification systems, a GMM is learned for the imposter class, and the target-speaker model is approximated by adaptation from the imposter GMM, *i.e.* modification of the imposter GMM model. An complete adaptation should be done on the mean vector, covariance matrix and the mixing parameter; but it is usually performed for the mean vector only since it is empirically observed that adaptation also on covariance matrix and the mixing parameter would yield less favorable results.

As such, the GMM modeling, *i.e.* the estimation of the parameters, and especially the adaptation for the target-speaker model are not optimal. This motivated us to investigate into a method which would rely less on the estimated parameters and could rectify inaccuracies therein.

AdaBoost methods, introduced by Freund and Schapire [2], provides a simple yet effective stagewise learning approach: It learns a sequence of more easily learnable weak classifiers, each of them needing only slightly better than random guessing, and boosts them into a single strong classifier by a linear combination of them. The weak classifiers, each derived based on some simple, coarse estimates, need not to be optimal. Yet, the AdaBoost learning procedure provides an optimal approach for combining them into the strong classifier.

Originating from the PAC (probably approximately correct) learning theory [11, 5], AdaBoost provably achieves arbitrarily good bounds on its training and generalization errors [2, 10] provided that weak classifiers can perform slightly better than random guessing on every distribution over the training set. It is also shown that such simple weak classifiers, when boosted, can capture complex decision boundaries [1].

Relationships of AdaBoost to functional optimization and statistical estimation are established recently. It is shown that the AdaBoost learning procedure minimizes an upper error bound which is an exponential function of the margin on the training set [9]. Several gradient boosting algorithms are proposed [3, 6, 12], which provides new insights into AdaBoost learning. A significant advance is made by Friedman *et al.* [4]. It is shown that the AdaBoost algorithms can be interpreted as stagewise estimation procedures that fit an additive logistical regression model. Both the discrete AdaBoost [2] and the real version [10] optimize an exponential loss function, albeit in different ways. The work [4] links AdaBoost, which was advocated from the machine learning viewpoint, to the statistical theory.

In this paper, we propose a new method for the speaker verification: that using AdaBoost learning based on the GMM. We start with an imposter GMM and a target-speaker GMM, more specifically, their mean vectors and covariance matrices but not the mixing weights. Although the GMM models are inaccurate, we are able to construct a sequence of weak classifiers based on these GMM's, weak classifiers meaning slightly better than random guessing. The AdaBoost procedure (1) sequentially and adaptively adjusts weights associated with the training examples which helps to construct and select the next good weak classifiers, and (2) combine them sensibly to constitute a boosted strong classifier. The combination is optimality in the sense of maximum margin. Experiments show that the boosted GMM classifier yields 10.81% relative reduction in equal error rate for the same handsets and 11.24% for different handsets, a significant improvement over the baseline adapted GMM system.

2. GMM Representation of Speaker Voices

In speaker verification, the task is to verify the target speaker and to reject imposter speakers. Two GMM are built from training data, the *universal background model* (UBM) for the imposter class and the target speaker model. This section describes the GMM modeling, on which most of the state-of-the-art systems are based, as the starting point of the proposed method.

Mel-frequency cepstral coefficients (MFCCs) are used as acoustic features for signals of speaker voices. All utterances

are pre-emphasized with a factor of 0.97. A Hamming window with 32ms window length and 16ms window shift is used for each frame. Each feature frame consists of 10 MFCC coefficients and 10 delta MFCC coefficients. Finally, the *relative spectral* (RASTA) filter and *cepstral mean subtraction* (CMS) are used to remove linear channel convolutional effects on the cepstral features. Therefore, each window of signal frames is represented by a 20-dimensional feature vector, which we call a feature frame.

A sequence of N feature frames, denoted $\mathcal{X} = \{x_1, \cdots, x_N\}$, are observed from the utterance of the same speaker. Assuming the feature vectors are independent, the probability (likelihood) of the observation $\mathcal{X}$ can be obtained as follow:

$$p_k(x_i) = \mathcal{N}(x_i; \mu_k, \Sigma_k) \quad (1)$$

$$p(x_i|\Lambda) = \sum_{k=1}^{K} u_k \cdot p_k(x_i) \quad (2)$$

$$p(\mathcal{X}|\Lambda) = \prod_{t=1}^{T} p(x_i|\Lambda) \quad (3)$$

Here, u_k is the weight of Gaussian mixture $\mathcal{N}(x_i; \mu_k, \Sigma_k)$ with mean μ_k and covariance matrix Σ_k, and $\Lambda = \{u_k, \mu_k, \Sigma_k\}$ is also used to represent the model parameters.

Using the EM algorithm, we can get a local optimum $\hat{\Lambda}$ for Λ under the MLE criterion by maximizing the probabilistic density for all observation frames.

$$\hat{\Lambda} = \arg\max p(\mathcal{X}|\Lambda) \quad (4)$$

The *universal background model* (UBM) is obtained by the MLE training. The target speaker model may also be obtained in a similar way.

However, to reduce computation and to improve performance when only a limited number of training utterances are available, some adaptation techniques were proposed, *e.g.* [8], in which MAP adaptation outperforms the other two, maximum likelihood linear regression (MLLR) adaptation and eigenvoices method.

MAP adaptation derived target speaker model by adapting the parameters of the UBM using target speaker's training data under the MAP criterion. Experiments show that when only the mean vectors of the UBM model are adapted, we can get the best performance. Given a UBM model Λ_{UBM} and training data $\mathcal{X}$ from the hypothesized speaker, the adaptation can be done as follow,

$$P(k|x_i) = \frac{u_k \cdot p_k(x_i)}{p(x_i|\Lambda_{\mathrm{UBM}})} \quad (5)$$

$$n_k = \sum_{t=1}^{T} P(k|x_i) \quad (6)$$

$$E_k(\mathcal{X}) = \frac{1}{n_k} \sum_{t=1}^{T} P(k|x_i) x_i \quad (7)$$

$$\hat{\mu}_k = \alpha_k E_k(\mathcal{X}) + (1 - \alpha_k)\mu_k \quad (8)$$

$$\alpha_k = \frac{n_k}{n_k + r} \quad (9)$$

where r is a constant relevance factor fixed at $r = 16$. The parameters Λ_{SP} of the target speaker GMM is thus obtained.

The classification for speaker verification may be performed based on the the log likelihood ratio LLR. Given the observation $\mathcal{X}$, the LLR is defined as

$$LLR(\mathcal{X}) = \log \frac{p(\mathcal{X}|\Lambda_{\mathrm{SP}})}{p(\mathcal{X}|\Lambda_{\mathrm{UBM}})} = \sum_{i=1}^{N} [\log p(x_i|\Lambda_{\mathrm{SP}}) - \log p(x_i|\Lambda_{\mathrm{UBM}})] \quad (10)$$

Because the corresponding means and covariances have been estimated, the LLR can be computed analytically. The decision function is

$$H(\mathcal{X}) = +1 \quad if\ LLR(\mathcal{X}) > \tau \quad (11)$$

$$= -1 \quad otherwise \quad (12)$$

The threshold τ can be adjusted to balance between the accuracy and false alarm rates (*i.e.* to choose a point on the ROC curve).

3. AdaBoost Learning

The basic form of AdaBoost [2] is for two class problems. A set of N labelled training examples is given as $(x_1, y_1), \ldots, (x_N, y_N)$, where $y_i \in \{+1, -1\}$ is the class label for the example $x_i \in \mathbb{R}^n$. AdaBoost assumes that a procedure is available for learning a weak classifiers $h_m(x)$ $(m = 1, 2, \ldots, M)$ from the training examples, with respect to a distribution of the examples determined by the associated weight w_i. It aims to learn a stronger classifier as a linear combination of the M weak classifiers

$$H_M(x) = \sum_{m=1}^{M} \alpha_m h_m(x) \quad (13)$$

The classification of x is obtained as the sign of $H_M(x)$. It may be advantageous to use a normalized version of the above:

$$H(x) = \frac{\sum_{m=1}^{M} \alpha_m h_m(x)}{\sum_{m=1}^{M} \alpha_m} \quad (14)$$

Now, the classifier function is Sign[H(x)] and the normalized confidence score is $|H(x)|$. The learning procedure, described in Fig.1, is aimed to derive α_m and $h_m(x)$.

In the simplest case, the error is to be minimized. An error occurs when $H(x) \neq y$, or $yH_M(x) < 0$. The "margin" of an example (x, y) achieved by $h(x) \in \mathbb{R}$ on the training set examples is defined as $yh(x)$. This can be considered as a measure of the confidence of the h's prediction. The upper bound on classification error achieved by H_M can be derived as the following exponential loss function [9]

$$J(H_M) = \sum_i \exp[-y_i H_M(x_i)] = \sum_i \exp\left[-y_i \sum_{m=1}^{M} \alpha_m h_m(x)\right] \quad (15)$$

AdaBoost construct $h_m(x)$ by stagewise minimization of Eq.(16).

A characteristics of AdaBoost is that during the learning, the AdaBoost re-weights each example, *i.e.* adaptively updates $w_i^{(m)}$ according to the classification performance of the learned weak classifiers (Step 2(3)). More difficult examples are associated with large weights so that more emphasis in the next round

0. (Input)
   (1) Training examples $\{(x_1, y_1), \ldots, (x_N, y_N)\}$, where $N = a + b$; of which a examples have $y_i = +1$ and b examples have $y_i = -1$;
   (2) Termination condition C (*e.g.* based on Recall/False Alarm, or $M_{\max}$);
1. (Initialization)
   $w_i^{(0)} = \frac{1}{2a}$ for those examples with $y_i = +1$ or
   $w_i^{(0)} = \frac{1}{2b}$ for those examples with $y_i = -1$.
   $M = 0$;
2. (Stagewise Learning)
   while C not satisfied
   (1) $M \leftarrow M + 1$;
   (2) Learn h_M;
   (3) Update $w_i^{(M)} \leftarrow \exp[-y_i H_M(x_i)]$, and normalize to $\sum_i w_i^{(M)} = 1$;
3. (Output)
   $H(x) = \frac{\sum_{m=1}^{M} \alpha_m h_m(x)}{\sum_{m=1}^{M} \alpha_m}$.

Figure 1: Discrete AdaBoost Algorithm.

of learning weak classifier will be put on those examples. The weighted error is

$$\epsilon_m = \sum_i w_i^{(m)} 1[y_i \neq H(x_i)] \tag{16}$$

Minimizing the above is the objective of h_{m+1}.

Given the current $H_{M-1}(x) = \sum_{m=1}^{M-1} \alpha_m h_m(x)$, and the newly learned weak classifier h_M (learned according to the weights $w^{(M-1)}(x, y)$), the best combining coefficient α_M for the new strong classifier $H_M(x) = H_{M-1}(x) + \alpha_M h_M(x)$ is the one which leads to the minimum cost:

$$\alpha_M = \arg\min_{\alpha} J(H_{M-1}(x) + \alpha h_M(x)) \tag{17}$$

The minimizer is

$$\alpha_M = \frac{1}{2} \log \frac{P(y = +1 \mid x, w^{(M-1)})}{P(y = -1 \mid x, w^{(M-1)})} \tag{18}$$

The weight for each (x, y) are then updated

$$w^{(M)}(x, y) = w^{(M-1)}(x, y) \exp\left(-\alpha_M y h_M(x)\right) \tag{19}$$

which is equivalent to Step 2.(3).

The basic AdaBoost algorithm is for two-class problem. In multi-class problems, where the class number $C > 2$, the final decision could be obtained by combining multiple binary Adaboost classifiers. There are two popular schemes for extending binary classifiers to the multi-class case. One is the one-against-all strategy, in which classification is performed between each class and the remaining. This needs to train a total of C boosted classifiers. The other is the one-against-one strategy, in which classification is performed between each pair. The later strategy will require $C(C-1)/2$ classifiers. In our system, we adopt the former scheme to save the computational cost. The output of the C strong classifiers are normalized as Eq.(14) to provide the score for comparison. The final decision is made by maximizing the score.

Table 1: DCF, EER and relative reduction (SH: Same Handset. DH: Different Handset)

Method	DCF (SH)	DCF (DH)
Adapted GMM	0.0460	0.1816
Boosted GMM	0.0395	0.1729
Rel. Reduction	14.13%	4.79%
Method	EER (SH)	EER (DH)
Adapted GMM	4.90%	21.96%
Boosted GMM	4.37%	19.49%
Rel. Reduction	10.81%	11.24%

4. Learning Weak Classifiers

An algorithm is needed to learn weak classifiers (Step 2.(2) in Fig.1). In this work, we define a set of candidate weak classifiers based on the model parameters (μ_k, Σ_k) provided by the two GMM's (but without using the mixing parameter u_k in this stage), and find the best one from this set in each stage.

At iteration M, we define the set of base weak classifier using the component LLR's

$$h_k^{(M)}(x) = \log \frac{p(x \mid \Lambda_{\rm SP}, w^{(M-1)})}{p(x \mid \Lambda_{\rm UBM}, w^{(M-1)})} - \tau_k^{(M)} \tag{20}$$

$$\tag{21}$$

where the $\tau_k^{(M)}$ is set to ensure a required accuracy. The best weak classifier is the one for which the false alarm is minimized:

$$k^* = \arg\min_k FA(h_k^{(M)}(x)) \tag{22}$$

where FA is the false alarm caused by $h_k^{(M)}(x)$ (also w.r.t. $w^{(M-1)}$). This gives us the best weak classifier as

$$h_M(x) = h_{k^*}^{(M)}(x) \tag{23}$$

5. Experiments

5.1. Data Set

All experiments are conducted on the 1996 NIST speaker recognition evaluation data set. Only male speakers are investigated. There are 21 male target speakers and 204 male impostors in the evaluation data set. The training utterances for each target speaker are extracted from two sessions originating from two different handsets, one minute per session. As for the testing, there are 321 target trials and 1060 impostor trials. The duration of each test utterance is about 30 seconds.

5.2. Results

The system performance is evaluated using the *detection error tradeoff* (DET) curve, *detection cost function* (DCF) and *equal error rate* (EER). DCF can be defined as follow [7],

$$DCF = C_{\rm fr} \cdot p_{\rm fr} \cdot P_{\rm tar} + C_{\rm fa} \cdot p_{\rm fa} \cdot P_{\rm imp} \tag{24}$$

Here, $p_{\rm fr}$ and $p_{\rm fa}$ are false rejection rate and false acceptance rate respectively at a operating point. $C_{\rm fr}$ and $C_{\rm fa}$ are costs for false rejection and false acceptance. $P_{\rm tar}$ and $P_{\rm imp}$ are the prior probability of target trials and impostor trials. $P_{\rm tar} = 0.01$ and $P_{\rm imp} = 0.99$.

EER and DCF for Boosted GMM and Adapted GMM under the same and the different handsets are shown in Table 1. DET curves for Adapted GMM and Boosted GMM for the same handset data are shown in Figure 2, and those for the different handsets are shown in Figure 3. From the results, we could conclude that boosted GMM yields a significant improvement on relatively reduction of over 10% than the baseline adapted GMM approach. The improvement is due to the optimal combination of individual GMM learned by non-parametric AdaBoost method.

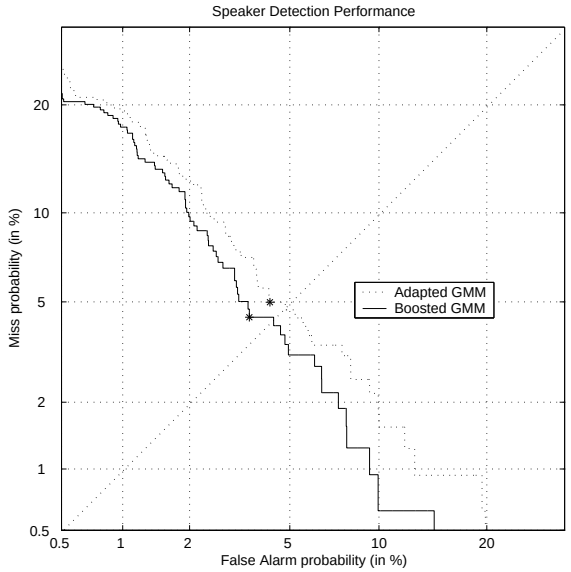

Figure 2: DET curves for boosted GMM and Adapted GMM on the same handset.

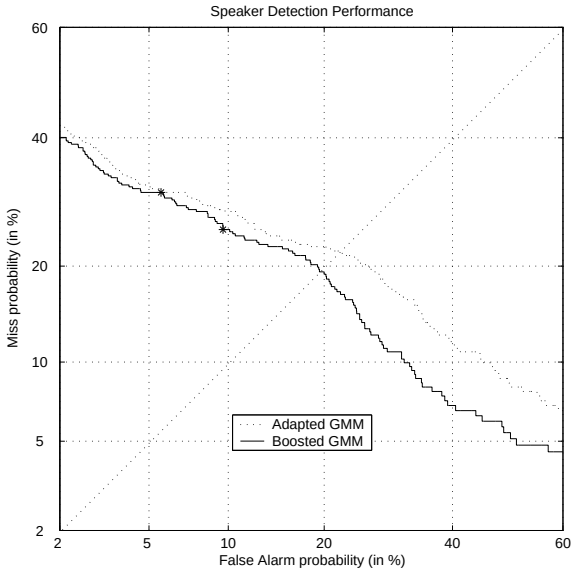

Figure 3: DET curves for boosted GMM and Adapted GMM on the different handsets

5.3. Conclusion

We proposed a novel boosting learning based algorithm that could significantly improve the performance of the baseline adapted GMM system. The boosted GMM optimally combines weak classifiers, *i.e.* component log likelihood ratio in GMM, to a strong classifier. AdaBoost provides an effective framework for fusing different models. In future, it would be interesting to investigate how to use AdaBoost at low level feature (such as cepstral) level to learn better speaker models.

6. References

[1] L. Breiman. "Arcing classifiers". *The Annals of Statistics*, 26(3):801–849, 1998.

[2] Y. Freund and R. Schapire. "A decision-theoretic generalization of on-line learning and an application to boosting". *Journal of Computer and System Sciences*, 55(1):119–139, Aug 1997.

[3] J. Friedman. "Greedy function approximation: A gradient boosting machine". *The Annals of Statistics*, 29(5), October 2001.

[4] J. Friedman, T. Hastie, and R. Tibshirani. "Additive logistic regression: a statistical view of boosting". *The Annals of Statistics*, 28(2):337–374, April 2000.

[5] M. J. Kearns and U. Vazirani. *An Introduction to Computational Learning Theory*. MIT Press, Cambridge, MA, 1994.

[6] L. Mason, J. Baxter, P. Bartlett, and M. Frean. Functional gradient techniques for combining hypotheses. In A. Smola, P. Bartlett, B. Schölkopf, and D. Schuurmans, editors, *Advances in Large Margin Classifiers*, pages 221–247. MIT Press, Cambridge, MA, 1999.

[7] NIST. the 1996 speaker recognition evaluation plan. http://www.nist.gov/speech/tests/spk/.

[8] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn. Speaker verification using adapted gaussian mixture models. *Digital Signal Processing*, 10:19–41, 2000.

[9] R. Schapire, Y. Freund, P. Bartlett, and W. S. Lee. "Boosting the margin: A new explanation for the effectiveness of voting methods". *The Annals of Statistics*, 26(5):1651–1686, October 1998.

[10] R. E. Schapire and Y. Singer. "Improved boosting algorithms using confidence-rated predictions". In *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*, pages 80–91, 1998.

[11] L. Valiant. "A theory of the learnable". *Communications of ACM*, 27(11):1134–1142, 1984.

[12] R. Zemel and T. Pitassi. "A gradient-based boosting algorithm for regression problems". In *Advances in Neural Information Processing Systems*, volume 13, Cambridge, MA, 2001. MIT Press.