



# A Discriminative Performance Metric for GMM-UBM Speaker Identification

Omid Dehzangi<sup>1</sup>, Bin Ma<sup>2</sup>, Eng Siong Chng<sup>1</sup>, Hai Zhou Li<sup>1,2,3</sup>

<sup>1</sup> School of Computer Engineering, Nanyang Technological University, Singapore

<sup>2</sup> Human Language Technology Department, Institute for Infocomm Research, A\*STAR, Singapore 138632

<sup>3</sup> Department of Computer Science and Statistics, University of Eastern Finland, FI-80101 Joensuu, Finland

{dehzangi, aseschnj}@pmail.ntu.edu.sg, {mabin, hli}@i2r.a-star.edu.sg

## Abstract

Gaussian mixture modeling with universal background model (GMM-UBM) is a widely used method for speaker identification, where the GMM model is used to characterize a specific speaker's voice. The estimation of model parameters is generally performed based on the maximum likelihood (ML) or maximum a posteriori (MAP) criteria. In this way, interspeaker information that discriminates between different speakers is not taken into account. To overcome this limitation, we design a discriminative performance metric to capture interspeaker variabilities leading to improve the classification capability of speaker models. A learning algorithm is presented to tune the Gaussian mixture weights by optimizing the frame classification accuracy of GMM classifiers. We design an objective function to directly relate the model parameters to the performance metric. The comparative study of the proposed method is done with the basic GMM-UBM system on the 2001 NIST SRE corpus. Experimental results demonstrate that the proposed learning algorithm considerably improves the GMM-UBM system on speaker identification.

**Index Terms:** speaker identification, GMM-UBM, discriminative performance metric

## 1. Introduction

The goal of speaker identification is to automatically determine the identity of an unknown speaker from his/her voice, by matching the unknown speech sample to a set of  $N$  known speakers in the database [1]. The best-matched speaker in the database is recognized as the identified person. Speaker identification task can be generally classified as text-dependent (where speakers are required to speak certain phrases) or text-independent (where speakers are allowed to utter arbitrary text). Speaker identification is widely applied to many fields from confidential data access to audio indexing in multimedia [2].

Many classifier approaches, such as vector quantization (VQ) [3], Gaussian mixture models (GMMs), hidden Markov models (HMMs) and neural networks (NN) [4] [5], have been studied for speaker recognition, with GMMs being one of the best performing approaches [6], [7]. GMM is a powerful approach to modeling a speaker's characteristics for its flexibility to approximate the underlying probability distribution in the speaker acoustic space [1], [8]. To obtain a high resolution characterization of the underlying acoustic space for good recognition performance, the number of Gaussian mixture components should be adequately large.

Generally, a UBM with a large number of Gaussian mixture components is created first, based on a large amount of speech data from many non-target speakers. Then the speaker specific models are created by adapting the UBM using MAP with speaker specific speech data [9]. This GMM-

UBM based approach has shown to be very successful in accurately identifying speakers from a large population and is currently the most predominant approach in text-independent speaker recognition [7]. The GMM-UBM is used as our baseline system.

In this paper, we design a discriminative performance metric (DPM) to adjust classifier parameters in a direct manner so that the frame classification error can be minimized. Each MAP adapted GMM target speaker model in a GMM-UBM system is considered as a detector of the current speaker versus the rest of the speakers. We propose an algorithm to learn the Gaussian mixture weights of the GMM models based on a criterion for minimizing the frame classification error rate of each GMM speaker model in a one versus the rest manner.

A receiver operating characteristic (ROC) curve is widely used to illustrate explicitly the overall performance and the possible error at each of the operating points [10]. An operating point on the ROC curve is determined by a decision threshold. The performance of such an operating point is measured by a performance cost function [11]. Naturally, the best operating point on a ROC curve yields the lowest cost. We are interested in improving the ROC curve by tuning the Gaussian mixture weights of each speaker GMM. The proposed algorithm reestimates Gaussian weights in the GMM model by finding the best operating point of the classifier assuming that the weights of all other Gaussians are given and fixed. The resulting weight is locally optimal in the sense that it maximizes the frame classification accuracy of the GMM classifier on the training data. Note that the actual objective in speaker identification is the classification accuracy for the whole speech segments, while the criterion to be optimized in the proposed learning algorithm is frame classification rate. However, the experimental results confirm that the optimization of such criterion leads to improve results on speaker identification.

The paper is organized as follows: In Section 2, we describe the GMM-UBM system speaker identification system. In Section 3, we introduce the proposed DPM design. In Section 4, we describe the experimental setup and results, and section 5 concludes the paper.

## 2. GMM-UBM Speaker Identification

In GMM-UBM based speaker identification system, a speaker-independent UBM of  $M$  Gaussian mixture components is trained via the expectation-maximization (EM) algorithm using a large amount of speech data from many non-target speakers [12], [7]. Given a feature vector  $\mathbf{x}$ , a probability density function (pdf), parameterized in a mixture of Gaussians, is

$$p(\mathbf{x} | \lambda) = \sum_{i=1}^M w_i g_i(\mathbf{x}) \quad (1)$$

where  $w_i$  is the weight of the  $i^{th}$  component and  $g_i(\cdot)$  is the Gaussian function with corresponding mean vector,  $\mu_i$ , and covariance matrix,  $\Sigma_i$ ,

$$g_i(\mathbf{x}) = \frac{1}{(2\pi)^{D/2} |\Sigma_i|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x} - \mu_i)' \Sigma_i^{-1} (\mathbf{x} - \mu_i)\right\} \quad (2)$$

For each speaker, a GMM can be created through an MAP adaptation of the UBM. For a group of  $S$  speakers,  $\Gamma = \{1, 2, \dots, S\}$ , represented by the corresponding speaker models,  $\lambda_1, \lambda_2, \dots, \lambda_S$ , the objective is to find the speaker model which has the maximum log likelihood, given the input feature vector sequence,  $X = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_T\}$ . Assuming independence between observations, the decision rule then becomes

$$\hat{s} = \arg \max_{1 \leq s \leq S} \sum_{t=1}^T \log p(\mathbf{x}_t | \lambda_s) \quad (3)$$

We do not consider channel and score normalization for the speaker identification task in this paper.

### 3. The Proposed Discriminative Performance Metric

Figure 1 illustrates the training mechanism of the speaker identification system. The objective function for the optimization is defined as a function of GMM parameters of interest. Here, we attempt to tune the Gaussian mixture weights of GMMs. In the following sections we introduce a learning mechanism to learn the mixture weights by optimizing a local performance metric.

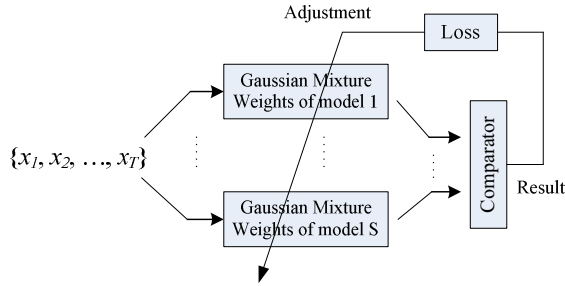


Figure 1: The architecture of the speaker identification system and training mechanism.

#### 3.1. Best operating point in 2-class problems

In this section, we describe how to find the best operating point of a classifier in a 2-class problem. We will use it as an ingredient in the learning algorithm introduced in the next section. Assuming a 2-class problem (with positive 'pos' and negative 'neg' class labels), given a test set of  $P$  positive and  $N$  negative labeled patterns, a classifier with crisp outputs generates a  $2 \times 2$  confusion matrix as shown in Figure 2 representing the performance of the classifier. The accuracy of the classifier is defined as:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (4)$$

while  $TP$ ,  $FP$ ,  $TN$ , and  $FN$  are numbers of outputs for true positives, false positives, true negatives and false negatives. On the other hand, scoring classifiers assign a scalar value to each input feature vector  $\mathbf{x}_t$ . For instance, Bayes classifier outputs posterior probability distribution over classes. A measure  $\Phi(\mathbf{x}_t)$  for 2-class problem can be defined as:

$$\Phi(\mathbf{x}_t) = \frac{p('neg' | \mathbf{x}_t)}{p('pos' | \mathbf{x}_t)} \quad (5)$$

where  $p('pos' | \mathbf{x}_t)$  and  $p('neg' | \mathbf{x}_t)$  denote the probabilities that the pattern  $\mathbf{x}_t$  belongs to positive and negative classes, respectively. With the above definition,  $\Phi(\mathbf{x}_t)$  is a numeric value expressing the degree that  $\mathbf{x}_t$  is believed to be of negative class. A classifier score can be converted to a crisp output by specifying a threshold on it. A feature vector is classified as negative if  $\Phi(\mathbf{x}_t)$  is greater than the specified threshold and positive otherwise. Since the posterior probabilities provided by the classifiers are not perfect due to deficient parameterization of the observations and insufficient data, a discriminative threshold based on the available data is to be found. In this way, the frame accuracy corresponding to each specified threshold can be calculated using Eq. (4).

		Actual Class	
		pos	neg
Predicted Class	pos	True Positives	False Positives
	neg	False Negatives	True Negatives
		P	N

Figure 2. Confusion matrix for crisp outputs

Based on a set of training samples, one is able to find the threshold to make the best decision for a classifier. An efficient algorithm for calculating the best threshold in such a way has been proposed in [13] where the patterns (feature vectors) are ranked in an ascending order of their  $\Phi(\cdot)$  measure (i.e.,  $\Phi(\mathbf{x}_1) < \Phi(\mathbf{x}_2) < \dots < \Phi(\mathbf{x}_T)$ ) while  $T$  is the number of feature vectors. Considering any threshold between  $\Phi(\mathbf{x}_k)$  and  $\Phi(\mathbf{x}_{k+1})$ , the first  $k$  patterns will be classified as positive and the remaining  $T-k$  patterns as negative. Maximum of  $T+1$  different threshold should be examined to find the *best\_th* (best threshold). The *best\_th* is simply the threshold that maximizes the accuracy of the classifier in Eq. (4). The important point is that the *best\_th* can be used as the weight for positive class.

#### 3.2. The learning mechanism

In this section, we propose an algorithm to learn the weights of Gaussian mixtures in each speaker GMM using the labelled training data. We develop the algorithm based on the idea that is described in the previous section. For a group of  $S$  speakers,  $\Gamma = \{1, 2, \dots, S\}$ , represented by the corresponding speaker models,  $\lambda_1, \lambda_2, \dots, \lambda_S$ . Given the input feature vector sequence,  $X = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_T\}$ , the speaker model  $\varphi$  maps the feature vector  $\mathbf{x}_t$  to a likelihood score, shown in Eq. (1), associated with speaker  $\varphi$ . To classify frame  $\mathbf{x}_t$  as the speaker  $\varphi$ , the associated likelihood score has to be the highest among all the speaker models,

$$\sum_{m=1}^M w_{m,\varphi} g_{m,\varphi}(\mathbf{x}_t) > \max \left\{ \sum_{m=1}^M w_{m,s} g_{m,s}(\mathbf{x}_t) \mid s=1, \dots, S \text{ and } s \neq \varphi \right\} \quad (6)$$

Each Gaussian  $g_{m,\varphi}(\cdot)$  in a target speaker GMM can be considered as a classifier and the corresponding weight  $w_{m,\varphi}$  is the degree to which this Gaussian contributes in the classification decision.

In the following, we present a procedure that optimizes the weights of a Gaussian mixtures one at a time. We consider the

problem of finding the weight  $w_{i,\varphi}$  of the  $i^{\text{th}}$  mixture from the speaker GMM  $\varphi$  as a 2-class problem where speaker  $\varphi$  is the positive class and  $\bar{\varphi}$ , the negative. The weight  $w_{i,\varphi}$  (given the weight of other mixtures) is found such that the classification rate on the training data is maximized. For this purpose, in the first step, the  $w_{i,\varphi}$  is set to zero (i.e.  $g_{i,\varphi}(\cdot)$  does not contribute in classification decision). Then, the training vectors of class  $\varphi$  that are classified correctly (TPs) with the current values of Gaussian mixture weights are removed from the training set. Note that these vectors will be classified correctly regardless of the value of  $w_{i,\varphi}$ . Similarly, the training vectors of  $\bar{\varphi}$  that are misclassified (FPs) with the current values of Gaussian mixture weights are removed from the training set. These vectors will be misclassified regardless of the value of  $w_{i,\varphi}$ . Then, a measure,  $\Phi_{i,\varphi}(\mathbf{x}_t)$  is calculated for each training vector  $\mathbf{x}_t$  left in the training set (i.e. TN and FN training patterns) as follows:

$$\Phi_{i,\varphi}(\mathbf{x}_t) = \frac{\max \left\{ \sum_{m=1}^M w_{m,s} g_{m,s}(\mathbf{x}_t) \Big|_{s=1, \dots, S \text{ and } s \neq \varphi} \right\} - \sum_{m=1, m \neq i}^M w_{m,\varphi} g_{m,\varphi}(\mathbf{x}_t)}{g_{i,\varphi}(\mathbf{x}_t)} \quad (7)$$

while  $\Phi_{i,\varphi}(\mathbf{x}_t)$  is the amount of weight needed for  $\mathbf{x}_t$  to be classified as speaker  $\varphi$ .

The pseudocode given below is then used to find the weight,  $w_{i,\varphi}$ . This algorithm receives a set of feature vectors  $F_x$  and their corresponding  $\Phi_{i,\varphi}(\cdot)$  and gives the best threshold (*best\_th*) as the output. The training vectors are first sorted in ascending order of their  $\Phi_{i,\varphi}(\cdot)$  measure. Assuming that a set of  $T$  vectors and their associated  $\Phi_{i,\varphi}(\cdot)$  is passed to this algorithm, a maximum of  $T+1$  thresholds are examined to find the best threshold. Note that for each specified threshold  $th$ , all training vectors,  $\mathbf{x}_t$  with  $\Phi_{i,\varphi}(\mathbf{x}_t) < th$  are classified as class  $\varphi$ . The best threshold is the one that maximizes the classification rate on the listed vectors. The *best\_th* is assigned as the weight  $w_{i,\varphi}$  of the  $i^{\text{th}}$  mixture from the speaker GMM  $\varphi$ .

#### Find\_Thresh

```

Input:  $F_x: \{ \mathbf{x}_1 \dots \mathbf{x}_T \}$  and their corresponding measure  $\Phi_{i,\varphi}(\cdot)$ 
 $F_x \leftarrow \text{Sort}(F_x, \text{'ascending'})$ 
 $th \leftarrow 0$  {classifying every vector as class  $\bar{\varphi}$ }
 $optimum \leftarrow \text{Calculate\_Accuracy}(F_x, th)$ 
 $best\_th \leftarrow 0$ 
{assume that  $\mathbf{x}_t$  and  $\mathbf{x}_{t+1}$  are two successive vectors in the list}
for every different  $th \leftarrow [ \Phi_{i,\varphi}(\mathbf{x}_t) + \Phi_{i,\varphi}(\mathbf{x}_{t+1}) ] / 2$ 
{ vectors  $\mathbf{x}_t$  having  $\Phi_{i,\varphi}(\mathbf{x}_t) < th$  are classified as class  $\varphi$ }
 $current \leftarrow \text{Calculate\_Accuracy}(F_x, th)$ 
if  $current > optimum$  then
 $optimum \leftarrow current$ 
 $best\_th \leftarrow th$ 
end if
end for
{assume  $\varepsilon$  is a positive number}
 $current \leftarrow \text{Calculate\_Accuracy}(F_x, \Phi_{i,\varphi}(\mathbf{x}_T) + \varepsilon)$  {classifying every vector as class  $\varphi$ }
if  $current > optimum$  then
 $optimum \leftarrow current$ 
 $best\_th \leftarrow th$ 
end if
return  $best\_th$ 

```

The search for the best combination of weights is conducted by optimizing each weight using the overall procedure given above in turn assuming that the order of the weights to be optimized is fixed. The proposed weight

learning mechanism assigns a weight to each Gaussian mixture weight attempting to better discriminate the vectors of the target speaker,  $\varphi$  and those of all other classes by finding the best operating point to include FN feature vectors and exclude TN feature vectors. By doing so, we locally optimize the tradeoff between the two decision error types namely, miss-detect and false alarm.

## 4. Experiments

In the following experiments, the 2001 NIST speaker recognition evaluation (SRE) corpus is used for a quick assessment. The data in the 2001 SRE is part of the Switchboard-Cellular corpora, which had been processed to remove any pauses and transmission channel echoes. This corpus includes files from 60 development speakers (2 minutes of speech for each of 38 males and 22 females) and files from 174 target speakers (74 males and 100 females). The training data consist of spontaneous speech recorded in different acoustic conditions: inside, outside, and vehicle. All of training data are transmitted over the mobile cellular networks of USA. There are totally 2038 test segments range between few seconds and a minute. A detailed description of the speech corpus may be found in the 2001 NIST SRE Plan [14].

### 4.1. Experimental setup

The feature extraction process was performed using the following steps: A feature vector including 12 Mel-Frequency Cepstral Coefficients (MFCCs) plus their first and second derivatives was extracted from each of the speech frames with a window of 30ms and a frame shift of 12.5ms. A voice activity detection algorithm was applied to select the speech frames and remove non-speech frames.

The UBM was trained on a total 2 hours of data from the 60-speaker development set of the 2001 NIST SRE, using the iterative EM algorithm by pooling all the development data together. Each Gaussian mixture in the UBM has a diagonal covariance matrix. All speaker GMM models were trained by MAP adaptation of the UBM. The baseline results were based on adapting the means and the weights of Gaussian components since it yielded a slightly lower identification error rate than adapting only the means. Then, the Gaussian mixture weights were reestimated using the DPM learning approach. The training procedure is repeated for several iterations over the whole training set while the order of training samples was fixed in all iterations.

The performance evaluation is conducted on 3 test cases: test segments in 10 seconds and 20 seconds which were extracted from the beginning of the whole test utterances, and the whole test utterances.

### 4.2. Results and discussion

Table 1 summarizes the results of the baseline GMM-UBM system and those with DPM learning mechanism attached on the three cases of 10-second segments, 20-second segments, and the whole test utterances. From the Table, we observe that ‘‘MAP+DPM’’ achieves 13.2% relative improvement over the baseline GMM-UBM system (‘‘MAP’’). It also shows that the performance of the system with different number of mixtures. As seen, the improvement over the baseline is consistent and the best results are achieved with 512 mixtures. Note that the same UBM is shared by all speaker models, it is only used in the DPM process for the optimization of mixture weights, but doesn’t involve in the identification test.

Table 1. Identification Error Rates (in %) of the system on 2001 NIST SRE task with different # of mixtures.

# of mixtures	system	IER (%)		
		10s	20s	whole
32	MAP	47.31	35.09	30.38
	MAP+DPM	44.66	33.35	28.51
64	MAP	43.28	30.58	27.09
	MAP+DPM	40.24	26.93	25.18
128	MAP	39.48	28.37	24.85
	MAP+DPM	35.29	25.38	21.5
256	MAP	38.11	27.74	24.05
	MAP+DPM	35.02	24.82	20.99
512	MAP	38.18	27.71	23.57
	MAP+DPM	34.53	24.41	20.46

We are now interested in Top-N identification results which consider the test utterance correctly identified if only the target speaker appears in the resulting top-N shortlist. It is to check if there is any gain obtained by DPM method in those cases. In Table 2, we present Top-1 to Top-5 identification error rates of the baseline and DPM method on 2001 NIST SRE test set using GMMs with 512 mixtures. The results demonstrate that the DPM method constantly improves the performance over the baseline.

Table 2. Identification Error Rates (in %) of the system with TOP-1 to TOP-5 shortlist

system		IER (%)		
		10s	20s	whole
Top-1	MAP	38.18	27.71	23.57
	MAP+DPM	34.53	24.41	20.46
Top-2	MAP	27.99	19.23	16.02
	MAP+DPM	23.83	16.49	13.81
Top-3	MAP	23.10	15.35	12.75
	MAP+DPM	19.71	13.15	10.73
Top-4	MAP	19.96	13.68	10.82
	MAP+DPM	17.19	12.11	9.22
Top-5	MAP	17.55	11.72	9.41
	MAP+DPM	15.28	10.21	8.07

Note that the DPM learning mechanism uses both miss-detect and false-alarm errors in the objective function. The results suggest that taking into account the two decision error types in parameter estimation is useful to improve separation of GMM models of different speakers.

## 5. Conclusions and Future work

There have been many attempts to provide discriminative solutions to the generative GMM-UBM speaker recognition framework [15] to improve its speaker discriminative ability. In this work, we introduced a discriminative performance metric to tune the Gaussian mixture weights of speaker GMMs in a GMM-UBM speaker identification system, that provides an alternative to the research problem. We presented a learning mechanism to learn the Gaussian mixture weights in an attempt to minimize the detection error rate of the each target speaker GMM by locally optimizing the tradeoff between the two decision error types namely, miss-detect errors and false alarm errors on the training data. To assess the usefulness of the proposed learning mechanism, we used a basic GMM-UBM system and the 2001 NIST SRE corpus for a quick assessment. The experiments were conducted on test segments with 10s, 20s durations as well as the whole test

utterances. The experimental results show that the proposed approach consistently and considerably improve the performance over GMM-UBM baseline.

As the future work, we intend to further the study towards speaker verification tasks on the state-of-the-art speaker recognition system (addressing the channel variability and score normalization) and over the latest NIST SRE corpora.

## 6. References

- [1] Reynolds D. A., "Speaker Identification and Verification using Gaussian Mixture Speaker Models", *Speech Comm.*, 17: 91-108, 1995.
- [2] Furui S., "Recent advances in speaker identification", *Patt. Rec. Lett.*, 18(9):859-872, 1997.
- [3] Soong, F.K., Rosenberg, A.E., Rabiner, L.R., Juang, B.H., "A vector quantization approach to speaker recognition", *Proc. IEEE Internat. Conf. on Acoust., Speech and Signal Process. (ICASSP)*, 387-390, 1985
- [4] Ramachandran, R.P., Farrell, K.R., Ramachandran, R., Mammon, R.J., "Speaker recognition – general classifier approaches and data fusion methods", *Patt. Rec.* 35:2801-2821, 2002.
- [5] Reynolds, D.A., "An overview of automatic speaker recognition technology", *Proc. IEEE Internat. Conf. on Acoust., Speech and Signal Process. (ICASSP)*, 472-475, 2002.
- [6] Reynolds, D.A., Comparison of background normalization methods for text-independent speaker verification. In: *Proc. Eurospeech*, pp. 963–966, 1997.
- [7] Reynolds, D.A., Quatieri, T., Dunn, R., "Speaker verification using adapted Gaussian mixture models", *Digit. Signal Process.* 10:19-41, 2000.
- [8] Reynolds, D.A., Rose, R.C., "Robust text-independent speaker identification using Gaussian mixture speaker models", *IEEE Trans. Speech Audio Process.* 3 (1):72-83, 1995.
- [9] Gauvain, J.L., Lee, C.H., "Maximum a posteriori estimation for multivariate Gaussian Mixture observations of Markov chains", *IEEE Trans. Speech Audio Process.*, 2(2):29-298, 1994.
- [10] Fawcett, T., "An introduction to ROC analysis", *Patt. Rec. Lett.*, 27: 861-874, 2006.
- [11] [http://www.itl.nist.gov/iad/mig/tests/sre/2010/NIST\\_SRE10\\_evalplan.r6.pdf](http://www.itl.nist.gov/iad/mig/tests/sre/2010/NIST_SRE10_evalplan.r6.pdf)
- [12] Dempster, A., Laird, N., Rubin, D., "Maximum likelihood from incomplete data via the EM algorithm", *J. Roy. Statist. Soc.* 39:1-38, 1977.
- [13] Fawcett, T., "ROC Graphs: Notes and Practical Considerations for Researchers", Technical Report HPL-2004.
- [14] <http://www.itl.nist.gov/iad/mig/tests/sre/2001/2001-spkrrec-evalplan-v05.9.pdf>
- [15] Longworth, C., Gales, M.J.F., "Discriminative adaptation for speaker verification", *Interspeech* 2006.