Odyssey 2010
The Speaker and Language Recognition Workshop
28 June – 1 July 2010, Brno, Czech Republic

Computationally Efficient Speaker Identification for Large Population Tasks using MLLR and Sufficient Statistics

A. K. Sarkar, S. Umesh and S. P. Rath

Department of Electrical Engineering
Indian Institute of Technology Madras, India
sarkar.achintya@gmail.com, umeshs@iitm.ac.in, shaktirath@gmail.com

Abstract

In conventional Speaker-Identification using GMM-UBM framework, the likelihood of the given test utterance is computed with respect to all speaker-models before identifying the speaker, based on the maximum likelihood criterion. The calculation of likelihood score of the test utterance is computationally intensive, especially when there are tens of thousands of speakers in database. In this paper, we propose a computationally efficient (Fast) method to calculate the likelihood of the test utterance using speaker-specific Maximum Likelihood Linear Regression (MLLR) matrices (which are precomputed) and sufficient statistics estimated from the test utterance only once. We show that while this method is an order of magnitude faster, there is some degradation in performance. Therefore, we propose a cascaded system with the Fast MLLR system identifying the top-N most probable speakers, followed by a conventional GMM-UBM to identify the most probable speaker from the top-N speakers. Experiments performed on the NIST 2004 database indicate that the cascaded system provides a speed up of 3.16 and 6.08 times for 1-side test (core condition) and 10 sec. test condition respectively, with a marginal degradation in accuracy over the conventional GMM-UBM system.

1. Introduction

In speaker identification, the unknown speaker is determined by the best matching model from among a list of registered speaker models [1]. Mathematically, it can be written as,

$$\hat{S} = \arg \max_{1 \leq S \leq N_S} \sum_{t=1}^{T} log\ P(x_t|\lambda_S) \quad (1)$$

where x and λ_S are the test feature vectors and speaker models in the database respectively. $\hat{S}$ is the recognized speaker based on Maximum Likelihood (ML) criteria of log likelihood score among N_S speakers in the database.

It can be observed from Eqn. (1) that it is a computationally intensive task to find the best speaker especially when the database has many speakers (e.g. $20,000$ or even more). This is because most of the time will be spent in calculating the likelihood of all the speaker models before finding the one with the maximum score.

Several techniques have been proposed in literature to reduce the computational load, and these include, pruning [2, 3], pre-quantization [4], Hash model [5], speaker cluster selection based [6] and Universal Background Model (UBM) based adaptive approach [7]. Many of these techniques have been proposed for speaker-verification task, but nevertheless are equally applicable in speaker-identification frame-work.

The pruning method as described in [2, 3] iteratively reduces the search space by dropping the most unlikely speakers on the fly. The pruning interval and number of speakers that will be pruned out during each iteration are the control parameters in the method. The pruning interval is defined as the number of new features vectors which will be taken and this successively acts on a reduced set of models in each iteration.

In the pre-quantization technique in [4], test feature vectors were down-sampled to reduce the computational task of the system . McLaughlin et al. [4] showed that upto $20:1$ down-sampling can be done on test feature vectors without loss of speaker verification performance. However, these studies have been done only on speaker verification task.

The Hash-model was introduced by Auckenthaler et al. [5] using an idea similar to that of fast scoring GMM-UBM system [7]. In Hash Modeling case, two GMM models are trained with the same amount of training data. One contains smaller number of Gaussian components called the *Hash model* (for e.g. 32 components) where as other is larger (e.g. 2048 components). There exists a correspondence between each component of the Hash model to a list of best scoring components of the large GMM. In test phase, feature vectors are first scored against all components of the Hash-model and then the best scoring Gaussian components are used to retrieve the dominant scoring mixtures of the large GMM. The speaker verification task was speeded up by about $10:1$, with minor degradation in performance.

A speaker clustering based identification task was proposed by Vijendra et al. [6], where speaker models were clustered into different groups or clusters based on a similarity measure. Each cluster was represented by a single speaker model *called representative speaker*. During identification, the test utterance was first scored against the representative speakers to select the best matching speaker group. It was shown that the identification can be speeded up by 4.4 to 8.7 times by varying the number of clusters and its respective speakers with minor degradation in performance.

It is important to note that the results reported in [3, 4, 5, 6] compare the efficiency of their proposed method to the conventional system where the conventional system likelihood calculation are done by *considering all Gaussian Components in the model*.

The conventional large Gaussian Mixture Model Universal Background Model (GMM-UBM) based speaker identification system was proposed by Reynolds et al. [7]. In this method, GMM-UBM is trained using data from various speakers. Individual speaker models are then derived from the GMM-UBM using Maximum a Posteriori (MAP) adaptation technique. Since speaker models are adapted from a single GMM-

UBM, a fast scoring technique was developed to calculate the likelihood of test utterance as follows: The top-C best Gaussian components are found by alignment of the test data with respect to GMM-UBM. Then, the top-C best Gaussian components per feature vector are traversed through all the speaker models to calculate the likelihood from respective models. The fast scoring technique significantly reduces the computational load of speaker identification system. If there are M Gaussian components in GMM-UBM and there are N_s speaker models, then the likelihood calculation requires only computing over $M + N_s \times C$ components.

In this paper, we propose an efficient speaker identification system using Maximum Likelihood Linear Regression (MLLR) matrices that are pre-computed for each speaker. This method is motivated by our earlier work in [8], where an efficient method was proposed for the estimation of Vocal Tract Length Normalization (VTLN) warp factor in speech recognition. The best warp factor was estimated by computing likelihoods for a range of possible warp factors. This was efficiently implemented by doing only one alignment of speech segment to collect *sufficient statistics*. The likelihood computation was done by a simple matrix multiplication of pre-computed VTLN matrices on the sufficient statistics. Hence, the proposed method provided huge computational gain compared to conventional VTLN system without degradation in performance.

Similar to [8], we propose an efficient speaker identification system, where each speaker is represented by a unique MLLR matrix which is estimated by using the speaker training data as adaptation data. During identification, two sufficient statistics are first estimated by alignment of test speech segment with respect to GMM-UBM. The sufficient statistics are multiplied appropriately by speaker's MLLR matrix to get the corresponding speaker-model likelihood. The speaker with maximum likelihood score is recognized as the identified speaker. Since the sufficient statistics are computed *only once* irrespective of the number of speaker-models and the computation of likelihood involves only matrix multiplication, the proposed method is computationally efficient. We call it Fast MLLR method.

Although the Fast-MLLR method is very computationally efficient, there is a degradation in speaker-identification performance compared to the conventional GMM-UBM system. We therefore, propose to use the Fast-MLLR method to obtain a set of most probable speakers (say, $N \ll N_s$) from the database. The conventional GMM-UBM system, then, finds the best speaker using top-C best Gaussian based scoring technique on these N speakers. The top-C best components are selected from the alignment information during estimation of sufficient statistics of the fast-MLLR system. Hence, the proposed cascade system does not need to align the test data twice. A schematic of the cascade speaker-identification system is shown in Fig. 1.

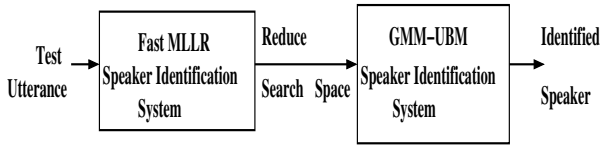

Figure 1: *Cascade speaker identification system.*

The paper is organized as follows: In Section 2, we describe the procedure to obtain the MLLR matrices for each speaker in the database. These matrices are then used for efficient computation of likelihood during test as described in Section 3. Section 4 describes the baseline GMM-UBM system based on top C-best mixture scoring technique. Section 5 describes experimental setup. In Section 6, we compare the performance of the baseline system with Fast-MLLR system. In Section 7, we describes the cascade speaker identification system. Finally, in Section 8, we conclude the paper.

2. Estimation of Speaker-Specific MLLR Matrix

Maximum Likelihood Linear Regression (MLLR) [9] is a speaker adaptation technique commonly used in Automatic Speech Recognition (ASR). It estimates a linear transformation matrix, W with respect to speaker independent (SI) model using speaker training data in ML framework. The matrix is then applied to mean vectors of the Gaussian components to get the adapted model. Mathematically,

$$\hat{\mu} = W\mu + b, \quad \hat{\Sigma} = \Sigma \tag{2}$$

where μ and Σ represent the mean and co-variance matrix of the original SI model, and (W, b) are the MLLR transformation parameters. $\hat{\mu}$ and $\hat{\Sigma}$ are the parameters of the transformed model.

The training of the Fast MLLR system is similar to the conventional MLLR system, with the difference being that it only estimates a speaker-wise MLLR matrix using the training data instead of actually deriving the speaker model from the GMM-UBM. The speaker wise MLLR matrix acts as the representation of the speaker. In our experiment, GMM-UBM is considered as the SI model. The speaker specific MLLR transformation is estimated using his/her training data. Single iteration is followed for MLLR transformation. Bias, b is not considered in our experiment.

The following steps are used to estimate the MLLR matrix for speaker, S.

1. Determine the probabilistic alignment of speaker training vectors, $x = \{x_1, \ldots x_t\}$ with respect to GMM-UBM mixtures, i.e. $\gamma_j(t)$ for mixture, j.

$$\gamma_j(t) = Pr(j|x_t) = \frac{w_j p_j(x_t)}{\sum_{k=1}^{M} w_k p_k(x_t)} \tag{3}$$

2. Using $\gamma_j(t)$ and x_t compute the following two statistics [10],

$$K_S^{(i)} = \sum_{r=1}^{R_s} \sum_{j=1}^{M} \frac{\mu_j^{(i)}}{\sigma_j^{(i)^2}} \sum_{t=1}^{T} \gamma_j^r(t)\, x_r^{'}(t) \tag{4}$$

$$G_S^{(i)} = \sum_{r=1}^{R_s} \sum_{j=1}^{M} \frac{1}{\sigma_j^{(i)^2}} \mu_j \mu_j^{'} \sum_{t=1}^{T} \gamma_j^r(t)\,. \tag{5}$$

$\mu_j^{(i)}$, $\sigma_j^{(i)^2}$ and M are the the i^{th} component of mean, the i^{th} component of co-variance and total number of Gaussian components in the model, respectively. R_s denotes number of training utterances for speaker S.

3. Finally, i^{th} row of the MLLR matrix for speaker S is obtained using,

$$W_{S,i}^{'} = G_S^{(i)^{-1}} K_S^{(i)^{'}} \tag{6}$$

The symbol $(.)'$ indicates matrix transpose operation.

These MLLR matrices $\{W_S\}_{S=1}^{N_s}$ are pre-computed using speaker-training data and stored. In the next section, we show how these pre-computed matrices can be used efficiently to compute the likelihood of test-utterance with respect to speaker-models.

3. Efficient calculation of Likelihood using pre-computed MLLR matrices

In this section, we describe how the optimal speaker can be identified by using the pre-computed MLLR matrix and the Sufficient Statistics collected only once *from the test data*. The following steps are involved in finding the best matching speaker:

Initial Step: Store the MLLR matrices of all speakers as described in Sec. 2.

1. Compute the $\gamma_j(t)$ of the *test feature vectors*, $x = \{x_1, \ldots x_t\}$ with respect to the GMM-UBM as Eqn. (3).
2. Estimate the following two statistics,

$$K^{(i)} = \sum_{j=1}^{M} \frac{\mu_j^{(i)}}{\sigma_j^{(i)^2}} \sum_{t=1}^{T} \gamma_j(t)\, x^{'}(t) \tag{7}$$

$$G^{(i)} = \sum_{j=1}^{M} \frac{1}{\sigma_j^{(i)^2}} \mu_j \mu_j^{'} \sum_{t=1}^{T} \gamma_j(t). \tag{8}$$

   The symbol $(.)'$ indicates matrix transpose operation.
   The statistics, K and G, are exactly *same sufficient statistics* as before, except that they are computed from *test data* and are *not specific to any speaker*.
3. The best probable speaker is selected in ML sense using the MLLR matrices of all speakers in the database, i.e.,

$$S^* = \arg\max_S \left\{ -\frac{1}{2} \left\{ \sum_{i=1}^{D} (w_{s,i} G^{(i)} w_{s,i}^{'} - 2K^{(i)} w_{s,i}^{'}) \right\} \right\} \tag{9}$$

   where, $w_{s,i}$ and D are the i^{th} row of MLLR matrix (W_S) of speaker S, and the dimension of the feature vectors, respectively. S^* is the identified speaker in the test utterance.

As mentioned earlier, the method is efficient, since the test utterance is aligned *only once* against the GMM-UBM model to compute the $\gamma_j(t)$ and the two necessary sufficient statistics, K and G. The statistics are then modified by the MLLR matrices of the speakers to find the identified speaker, S^*. It is a simple search over the MLLR matrices of the speakers in the database.

4. GMM-UBM system

The state-of-the-art GMM-UBM system [7] is used as the baseline system. GMM-UBM with 2048 gaussian components, is trained using data from the NIST 2002 SRE and the Switchboard-1 Release-2 with EM algorithm. Diagonal covariance matrices are considered in each Gaussian mixture. Speaker models are derived from GMM-UBM using his/her training data with single iteration of MAP. Only mean vectors of the GMM-UBM are adapted to get speaker adapted models. The value of relevance factor, 16 is used during MAP adaptation. During identification, top 15 best mixtures per feature vector [7] are considered to calculate the log likelihood from speaker models, which yields the best performance in our experiment.

5. Experimental setup

Experiments are performed on NIST 2004 SRE in two condition. One is single-side (1-side) training single-side (1-side) testing (core condition) and other is single-side training 10 sec. testing i.e. speaker models are trained using data from single-side condition and tested by utterances in single-side and 10 sec. condition. Details of the database can be found in [11]. The utterances are 5 minutes long in 1 side condition having around 2.5 minutes of speech. There are 310 unique speakers in the train set and 306 speakers in the test example set. For close loop speaker identification task, we considered 306 speakers who have both training and test utterances. The setup resulted in 1163 test utterances in each condition.

A 39 dimensional MFCC feature vector (C_1 to C_{13} with Δ and $\Delta\Delta$ excluding C_0) is extracted from the 8 kHz sampled speech signal at 10ms frame rate with 20ms Hamming windowing, over frequency band 300-3400Hz. Different frame removal techniques are used to remove the silence and very low energized feature vector as in [12]. Bi Gaussian modeling of energy components of the frames for NIST 2002 SRE and Switchboard-1 Release-2 and tri Gaussian modeling of normalized energy components of the frames for NIST 2004 SRE are used respectively. Finally, silence removed frames are normalized to zero-mean and unit-variance by removing the global mean and by dividing the standard deviation at utterance label. All experiments are conducted on a desktop with Intel Quad Core Processor (Q9550) with 2.83GHz CPU and 8 GB RAM. We measure the relative time taken between the approaches to assess their computational complexity.

6. Performance Comparison of GMM-UBM and Fast-MLLR systems

In this section, we compare the performance of GMM-UBM and Fast-MLLR speaker identification systems in terms of identification accuracy and time taken to process the data on identical computer set-up.

Table 1: *Comparison GMM-UBM system with Fast MLLR system for the NIST 2004 SRE with 306 speaker models.*

Test seg.	System	Acc. (%)	Avg. time/ uttn. (sec.)	Speed up
10 sec.	GMM-UBM	40.76	7.388	1×
	Fast MLLR	31.29	1.078	6.85×
1 side (5 min.)	GMM-UBM	60.71	44.155	1×
	Fast MLLR	47.54	4.984	8.86×

Table. 1 shows the individual system performance for different lengths of test utterances. Both systems in Table. 1 determine the best speaker for the unknown test utterance by estimating the likelihood from all speakers in the database i.e. 306 in our experiment. It is observed that Fast MLLR system always performs poorer than GMM-UBM system. On the other hand, the Fast-MLLR system provides a speed up of $1:6.85$ and $1:8.86$ for 10 sec. and 5minutes test utterances respectively. Note that in comparison, the GMM-UBM is using only top-15 mixtures in likelihood calculation and is already computationally efficient. The gains provided by Fast-MLLR are over this computationally efficient GMM-UBM system.

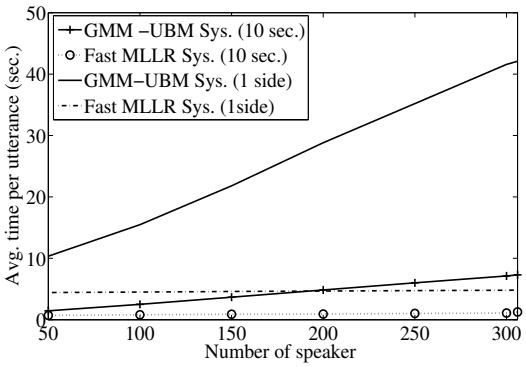

Figure 2: *Comparison of time taken to find the optimal speaker by GMM-UBM and Fast MLLR system.*

From Table. 1, we can conclude that Fast MLLR system is faster but does not provide the same accuracy as the baseline GMM-UBM system.

Fig. 2 shows the time taken by each system to find the optimal speaker as the number of speakers in the database and test duration increases. It can be observed that the time taken by Fast MLLR system to find the optimal speaker does not increase significantly as the number of speakers in database and test data duration increase. This is because, Fast MLLR System involves only matrix multiplication to calculate the likelihood after estimating *sufficient statistics*. Therefore, while Fast-MLLR has degradation in performance, it has significant advantage in terms of computation time. This computational advantage becomes even more significant when there are thousands of speakers in the database unlike the 306 considered in this experiment. In the next section, we investigate the use of a cascade system to combine the computational advantage of Fast-MLLR with the performance advantage of GMM-UBM.

7. Cascaded speaker identification system

In our proposed cascade-system, Fast-MLLR is used to *efficiently* identify the top-N most probable speakers from the thousands of speakers in the database. The conventional GMM-UBM is then used to test among these top-N speakers to find the optimal speaker. During test, the cascade system finds the best speaker from the database using the following steps,

- **Step1**: Fast MLLR System
  - (i) Find the best N ($\ll N_S$) probable speakers from the database of the test utterance as described in Sec. 3.
- **Step2**: GMM-UBM System
  - (ii) Get the top-C ($C = 15$) best Gaussian components per feature vector from the alignment information used in **Step1**.
  - (iii) Calculate the likelihood among N best speaker models using top-C Gaussian components knowledge [7].
  - (iv) Identify the speaker whose model provide max likelihood as Eqn. (1).

It is to be noted that speakers are represented by MLLR matrices (rather than models) in the Fast MLLR and adapted speaker models in the GMM-UBM system.

7.1. Effect of choice of N-best on the performance of Cascade system

We now investigate the choice of N in N-best for the Fast MLLR based system, so that the cascade system performance comes closer to the GMM-UBM system while saving computational time. First, we study the performance of the Fast MLLR system for N-best speaker identification task. The N-best speaker identification accuracy involves testing to see if the correct speaker of the test utterance lies within most probable N speakers. Fig.3 shows the N best speaker identification performance of the Fast MLLR system with GMM-UBM system for different duration of test segments.

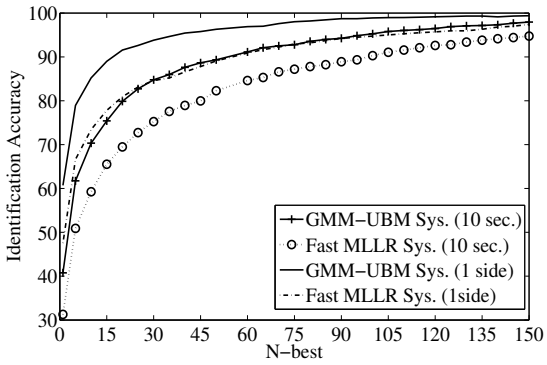

Figure 3: *Comparison of N-best speaker identification performance of Fast MLLR system with baseline system for different test segment.*

From Fig.3, it is observed that Fast MLLR system provides 10-best accuracy of 59.24 and 73.43, which is much higher than the GMM-UBM baseline 1-best performance of 40.76 and 60.71 for 10-sec and 1-side respectively. Now if we consider our cascade system configuration, it may not yield comparable results to the standalone GMM-UBM system, since, the GMM-UBM system itself has some identification error. However, as N is increased the performance of the cascade system should approach close to that of the standalone GMM-UBM system.

Table. 2 shows how accuracy and computation time of the cascade speaker identification varies over different value of N at Fast-MLLR stage. From Table. 2, following observation can be drawn:

- As value of N increases, the performance of cascade system performance comes closer to baseline GMM-UBM system but there is a corresponding drop in computational gain.
- For test segments with longer duration (e.g. 5 min.), cascade system achieves small computational gain when compared to shorter test segments (e.g. 10 sec.). This is due to the significant computation time taken by GMM-UBM system even for the reduced set of speakers (in Table. 1). However, in all cases, cascade system provides significant gain in computational time with marginal loss in performance
- By tuning the value of N a compromise between accuracy loss versus system speed can be achieved.

If we consider $N = 20$ for the cascade system, the computational speed is 6.08× and 3.16× faster than conventional baseline system with loss of accuracy being 0.86% and 1.04% for 10 sec. and 5 minutes (1-side) test segment respectively.

We would like to remind the reader that the above observations are for the case of 306 speakers in the database. If we

Table 2: *Comparison of speaker identification accuracy and computational time with baseline system for different value of N in cascade system on NIST 2004 SRE.*

Test seg.	System	N	Acc. (%)	(%) Acc. degrade over baseline	Avg. time/ uttn. (sec.)	Speed up over baseline
10 sec.	GMM-UBM	-	40.76	-	7.388	1×
	Cascade	10	38.95	1.81	1.028	7.19×
		20	39.90	0.86	1.215	6.08×
		30	40.33	0.43	1.408	5.25×
		40	40.41	0.35	1.594	4.63×
1 side (5 min.)	GMM-UBM	-	60.71	-	44.155	1×
	Cascade	10	58.21	2.50	13.335	3.31×
		20	59.67	1.04	13.970	3.16×
		30	59.93	0.78	14.797	2.98×
		40	60.02	0.69	15.662	2.82×

have thousands of speakers in the database, then the computational advantage offered by our cascade system will be very significant.

8. Conclusion

In this paper, we have proposed a cascade system for speaker identification tasks involving large population database. The first-stage system is based on MLLR with Sufficient Statistics to reduce the search space of speakers. This stage is computationally very efficient since it involves only matrix multiplication to compute likelihood for different speakers but involves some degradation in performance. The second stage uses GMM-UBM system which provides good identification accuracy but is computationally expensive especially when there are many speakers. The use of cascade system reduces the search-space for the GMM-UBM to identify the optimal speaker in a computationally efficient manner. The speed up in system performance is 6.08 and 3.16 times faster than conventional GMM-UBM for test utterance of duration 10 sec. and 1-side respectively and involves very little degradation in performance. In this paper, we have used only 306 speakers in the database, and the computational gain is expected to be very significant for databases having thousands of speakers.

9. Acknowledgment

A part of this work is supported by SERC project funding `SR/S3/EECE/058/2008` from the Department of Science & Technology, Ministry of Science & Technology, India.

10. References

[1] D. A. Reynolds, "Speaker Identification and Verification using Gaussian Mixture Speaker Models," *Speech Communication*, vol. 17, pp. 91–108, 1995.

[2] B. L. Pellom and J. H. L. Hansen, "An Efficient Scoring Algorithm for Gaussian Mixture Model Based Speaker Identification," *IEEE Signal Proc. Lett.*, vol. 5, pp. 281–284, Nov. 1998.

[3] T. Kinnunen, E. Karpov, and P. Franti, "A Speaker Pruning Algorithm for Real-Time Speaker Identification," *in Proc. Audio- and Video-Based Biometric Authentication, Guildford,U.K.*, pp. 639–646, 2003.

[4] J. McLaughlin, D. A. Reynolds, and T. Gleason, "A Study of Computation Speed-ups of the GMM-UBM Speaker Recognition System," *Eurospeech,Hungary*, pp. 1215–1218, 1999.

[5] R. Auckenthaler and J. S. Masion, "Gaussian Selection applied to Text-Independent Speaker Verification," *in Proc. Speaker Odyssey: The Speaker Recognition Workshop,Greece*, pp. 83–88, 2001.

[6] V. R. Apsingekar and P. L. De Leon, "Speaker Model Custering for Efficient Speaker Identification in Large Population Application," *IEEE Tran. on Audio, Speech and Language Processing*, vol. 17, No.4, May 2009.

[7] D.A. Reynolds, T.F. Quateri, and R.B. Dunn, "Verification using Adapted Gaussian Mixture Models," *Digital Signal Processing*, vol. 10, pp. 19–41, Jan2000.

[8] P. T. Akhil, S. P. Rath, S. Umesh, and D. R. sanand, "A Computationally Efficient Approach to Warp Factor Estimation in VTLN Using EM Algorithm and Sufficient Statistics," in *Interspeech2008*, September 2008.

[9] C. Leggetter and P. Woodland, "Maximum likelihood linear regression for speaker adaptation of Hmms," *Computer Speech and Language*, vol. 9, pp. 171–186, 1995.

[10] M.J.F. Gales, "Maximum Likelihood Linear Transformations for HMM-based Speech Recognition," *Computer Speech & Language*, vol. 12, pp. 75–98, 1998.

[11] The Evaluation Plan of NIST 2004 Speaker Recognition Campaign. http://www.itl.nist.gov/iad/mig//tests/sre/2004/SRE04_evalplan-v1a.pdf, ," .

[12] Jean Francois Bonastre, Nicolas Scheffer, Corinne Fredouille, and Driss Matrouf, "Nist'04 Speaker Recognition Evaluation Campaign: New LIA Speaker Detection Plateform based on ALIZE Toolkit," *in NIST SRE'04 Workshop, Toledo, Spain*, Jun. 2004.