

# EFFICIENT SPEAKER IDENTIFICATION USING SPEAKER MODEL CLUSTERING

Vijendra Raj Apsingekar and Phillip L. De Leon

New Mexico State University  
Klipsch School of Electrical and Computer Engineering  
Las Cruces, New Mexico USA 88003  
Phone: +1 (575) 646-3771, {vijendra, pdeleon}@nmsu.edu

## ABSTRACT

In large population speaker identification (SI) systems, likelihood computations between an unknown speaker's feature set and the registered speaker models can be very time-consuming and impose a bottleneck. For applications requiring fast SI, this is a problem. In prior work, we proposed the use of clusters of speaker models so that during the test stage, only a small proportion of speaker models in selected clusters are used in the likelihood computations resulting in a speed-up of  $2\times$  without loss in accuracy. In this paper, we improve the method by incorporating log-likelihoods into the initial clustering as well as cluster selection. The new method allows for fewer clusters to be searched and thus higher speed-up factors while still maintaining acceptable accuracy levels.

## 1. INTRODUCTION

The objective of speaker *identification* (SI) is to determine which voice sample from a set of known voice samples best matches the characteristics of an unknown input voice sample [1]. SI is a two-stage procedure consisting of training and testing. In the training stage speaker-dependent feature vectors are extracted from a training speech signal and a speaker model,  $\lambda_s$  is built for each speaker's feature set. Normally, SI systems use the mel-frequency cepstral coefficients (MFCCs) as the  $L \times 1$  feature vector and a Gaussian Mixture Model (GMM) of the feature set for the speaker model. The GMM is parameterized by the set  $\{w_i, \mu_i, \Sigma_i\}$  where  $w_i$  are the weights,  $\mu_i$  are the mean vectors, and  $\Sigma_i$  are the covariance matrices of the  $W$  component densities of the GMM. In the SI testing stage  $M'$  feature vectors,  $\mathbf{x}_m^{\text{test}}$  are extracted from a test signal (speaker unknown), scored against all  $S$  speaker models using a log-likelihood calculation, and the most likely speaker identity,  $\hat{s}$  decided according to

$$\hat{s} = \arg \max_{1 \leq s \leq S} \sum_{m=1}^{M'} \log p(\mathbf{x}_m^{\text{test}} | \lambda_s). \quad (1)$$

In assessing an SI system we measure identification accuracy as the number of correct identification tests divided by the total number of tests. For many years now, GMM-based systems have been shown to be very successful in accurately identifying speakers from a large population [1], [2].

In speaker *verification* (SV), the objective is to verify an identity claim. Although the SV training stage is identical to that for SI, the test stage differs. In the SV test stage, for the given test feature set a likelihood ratio is formed from the claimant model and that of a background model [3]. If the likelihood ratio is greater than a threshold value, the claim is accepted otherwise it is rejected. In SV, MAP-adapted

speaker models from a universal background model (UBM) with likelihood normalization are normally used [4].

In this paper, we consider the problem of slow speaker *identification* for large population systems. In such SI systems (and SV systems as well), the log-likelihood computations required in (1) have been recognized as the bottleneck in terms of time complexity [2], [5]. Although accuracy is always the first consideration, fast identification is also an important factor in many applications such as speaker indexing and forensic intelligence [6], [7].

Among the earliest proposed methods to address the slow SI/SV problem were pre-quantization (PQ) and pruning. In PQ, the test feature set is first compressed through subsampling (or another method) before likelihood computations [8]. PQ factors as high as 20 have been used without affecting SV accuracy. Application of PQ in order to speed-up SI has been investigated in [2] and results in a further real-time speed-up factor of as high as  $5\times$  with no loss in identification accuracy using the TIMIT corpus. In pruning [9], a small portion of the test feature set is compared against all speaker models. Those speaker models with the worst scores are pruned out of the search space. In subsequent iterations, other portions of the test feature set are used and speaker models are scored and pruned until only a single speaker model remains resulting in an identification. Using the TIMIT corpus, a speed-up factor of  $2\times$  has been reported with pruning [2]. Variants of PQ and pruning as well as combinations of the methods have been extensively evaluated in [2].

In [10], a hierarchical speaker identification (HSI) is proposed that uses *speaker clustering* which, for HSI purposes, refers to the task of grouping together feature sets from different speakers with similar acoustic data and modeling the superset, i.e. speaker cluster GMM. (In most other papers, speaker clustering refers to the task of grouping together unknown speech utterances based on a speaker's voice [11].) In HSI, a non-Euclidean distance measure between an individual speaker's GMM and the cluster GMMs is used to assign speakers to a cluster. Feature sets for intra-cluster speakers are then re-combined, cluster GMMs are re-built, distance measures are recalculated, and speakers are reassigned to "closer" clusters. The procedure iterates using the ISODATA algorithm until speakers have been assigned to an appropriate cluster. During this iterative procedure which uses the ISODATA algorithm, clusters with many speaker models are split and clusters with only a few speaker models are combined. The procedure continues until speakers have been assigned to an appropriate cluster. During the test stage, the cluster/speaker model hierarchy is utilized: first log-likelihoods are computed against the given cluster GMMs

in order to select the appropriate cluster for searching. Then log-likelihoods are computed against those speaker models in the cluster in order to identify the speaker. We note that a similar idea for reducing a search space using clusters or classes has long been used in the area of content-based image retrieval (CBIR) [12] but it appears that [10] was one of the first to use clusters for speeding up SI. Likewise, the use of speaker clusters have been used for fast speaker adaptation in speech recognition applications [13] as well as in the open-set speaker identification (OSI) problem [14].

Using a 40 speaker corpus, HSI requires only 30% of the calculation time (compared to conventional SI) while incurring an accuracy loss of less than 1% (details of the corpus and procedure for timing are not described). Unfortunately, HSI has a number of drawbacks including an extremely large amount of computation (which the authors acknowledge) required for clustering. Because of this required computation, the HSI method does not scale well with large population size. Although HSI was shown to speed up SI with little accuracy loss, the small number of speakers used in simulation does not provide any indication of how accuracy would degrade with much larger populations [15].

In a recent publication, a different approach toward efficient speaker recognition has been investigated. In [5], the authors *approximate* the required log-likelihood calculations in (1) with an approximate cross entropy (ACE) between a GMM of the test utterance and the speaker models; speed-ups are realized through reduced computation in ACE. The authors acknowledge potential problems with constructing a GMM of the test signal and offer methods to reduce this bottleneck. Also, if the test signal is short the GMM may not be accurate. The speaker verification results presented in [5] show a theoretical speed-up factor of 5 without any degradation in false acceptance. Open-set, speaker identification results show a theoretical speed-up factor of 62 for ACE.

In our research, the focus is *strictly* on fast speaker *identification*. In earlier work [16], we proposed a method to utilize clusters in order to speed-up SI, however, our work differs from [10] in two regards. First, rather than speaker clustering, we form clusters directly from the speaker models, i.e. speaker *model* clustering. This difference is important as it allows utilization of the simple  $k$ -means algorithm and leads to a scalable method for clustering which we demonstrated using the large population (630 speakers) TIMIT and NTIMIT corpora [16]. Second, we investigated searching more than one cluster so that any loss in identification accuracy due to searching too few clusters can be controlled; this allows a smooth trade-off between speed and accuracy. We demonstrated that search space could be reduced by 30% - 50% with little or no loss in accuracy; this search space reduction reduces the number of speaker models that (1) has to be computed over. Our work also differs from [5] in that we make no approximations to (1) relying instead on a reduction in the number of speaker models that (1) has to be calculated against. In addition, whereas the majority of the results presented in [5] are for SV, our focus is on fast SI.

In this paper, we extend our work by including log-likelihood criteria into  $k$ -means speaker model clustering. Our new results allow for further reductions in search space and thus higher speed-up factors while still maintaining acceptable accuracy levels. In addition, we provide new results which combine our method with PQ and pruning resulting in speed increases significantly greater than those cited in [10]

and comparable to those in [5] with little or no loss in SI accuracy. Finally, we note that although many techniques such as channel compensation and MAP adaptation of speaker models have been used to improve SI accuracy and robustness of GMM-based SI systems, our focus is on speeding-up identification of a baseline SI system. It is assumed that these techniques if applicable to the baseline system would also be applicable to our system as well.

This paper is organized as follows. In Section 2, we describe application of the  $k$ -means algorithm for speaker model clustering and two methods for selecting clusters to search. In Section 3, we describe the experimental evaluation and provide new results using the large population TIMIT (clean speech), NTIMIT (telephone-quality speech) and NIST2002 (cellular-quality speech) corpora. We also provide timing results for SI in order to evaluate the proposed method. In Section 4, we discuss our future directions with this research and in Section 5 we conclude the article.

## 2. SPEAKER MODEL CLUSTERING

### 2.1 Clustering

A direct method of determining clusters, taking into account all speaker models and training feature sets, leads to a difficult nonlinear optimization problem. As an alternate approach to [10], we previously proposed representing the speaker model as a point in  $L$ -dimensional space determined by its weighted mean vector (WMV) [16]

$$\bar{\mu} = \sum_{i=1}^W w_i \mu_i. \quad (2)$$

This representation allows for simple and efficient application of the  $k$ -means algorithm to cluster speaker models. The measure used in  $k$ -means to compute Euclidean distances from speaker model  $s$  to cluster centroid,  $\mathbf{r}_n$  is then

$$d(\lambda_s, \mathbf{r}_n) = \left[ (\bar{\mu}_s - \mathbf{r}_n)^T (\bar{\mu}_s - \mathbf{r}_n) \right]^{1/2}, \quad (3)$$

and  $\mathbf{r}_n$  is calculated as

$$\mathbf{r}_n = \frac{1}{K} \sum_{k=1}^K \bar{\mu}_k, \quad (4)$$

where  $K$  is the number of speakers in the cluster  $n$ , We call this approach to clustering “Euclidean  $k$ -means” since (3) is used as the measure. Because the cluster centroid does not have the required GMM parameters  $\{w_i, \mu_i, \Sigma_i\}$ , many distance measures as well as the Kullback-Liebnner (KL) divergence cannot be used in conventional  $k$ -means clustering.

### 2.2 Factoring in Log-Likelihoods into Clustering

Equations (2) and (3) provide a simple approach toward  $k$ -means-based speaker model clustering. However, the SI decision in (1) is based on log-likelihood and not on a Euclidean distance measure to the GMM. Therefore, we propose an additional step after Euclidean  $k$ -means clustering. First, we identify the speaker model,  $\lambda_n^{\text{CR}}$  which is nearest to each cluster centroid using (3) as in Fig. 1; this speaker model is called the cluster representative (CR). Second, we

measure the log-likelihood between a speaker model and CR as

$$d(\lambda_s, \lambda_n^{\text{CR}}) = - \sum_{m=1}^M \log p(\mathbf{x}_{s,m}^{\text{train}} | \lambda_n^{\text{CR}}) \quad (5)$$

where  $M$  is the number of training feature vectors and  $\mathbf{x}_{s,m}^{\text{train}}$  are the training feature vectors for speaker  $s$ . Third, speaker models are re-assigned to the nearest cluster using (5) as the distance measure. After the re-assignment step, cluster centroids are recomputed and new CRs (if any) are identified.

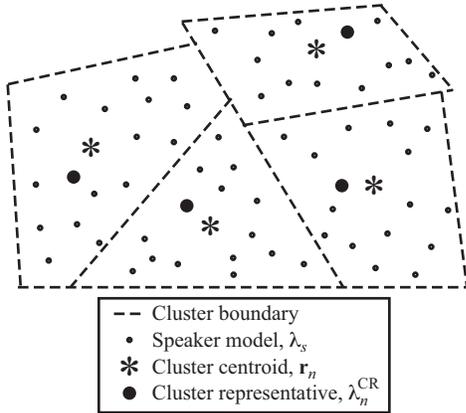


Figure 1: Space of speaker models and clusters.

If the distance measure in (3) is replaced by (5), a variation on  $k$ -means using the log-likelihood measure and CRs can be used for speaker model clustering instead of Euclidean distances and centroids; this overcomes the problem of the centroid not having the required GMM parameters for other distance measures. We call this alternate approach to clustering “log-likelihood  $k$ -means.” The algorithm for log-likelihood  $k$ -means clustering is listed in Algorithm 1.

---

**Algorithm 1** Speaker model clustering using a log-likelihood distance

---

- 1: Initialize cluster representatives,  $\lambda_n^{\text{CR}}, 1 \leq n \leq N$  using randomly-chosen speaker models
  - 2: Compute distance using (5) from  $\lambda_s$  to  $\lambda_n^{\text{CR}}, 1 \leq s \leq S$
  - 3: Assign each  $\lambda_s$  to the cluster with the minimum distance
  - 4: Compute new cluster centroids using (4) and determine  $\lambda_n^{\text{CR}}$
  - 5: Goto step 2 and terminate when cluster membership does not change.
- 

### 2.3 Cluster Selection and Testing

We propose two new methods to select the subset of clusters which will be searched during the test stage. Both methods are independent of how speaker models are clustered. In Method #1, the average of the test feature vectors is computed and those clusters as represented by their centroids nearest (Euclidean distance) to this average are searched. In Method #2, we use (5) with  $\mathbf{x}_m^{\text{test}}$  to identify the clusters which contain high likelihood speaker models. We note that both methods provide a relatively fast and efficient way to

select clusters for searching which is an important consideration for test-stage processing.

## 3. EXPERIMENTS AND RESULTS

For the TIMIT, NTIMIT, and NIST2002 corpora our SI system uses typical design parameters including a  $29 \times 1, 20 \times 1, 19 \times 1$ , respectively MFCC feature vector spanning the signal bandwidth and cepstral mean subtraction [1], [2]. For the TIMIT and NTIMIT corpora, we use  $W = 15$  component densities for the GMMs and approximately 24 s training signals and 6 s test signals. For the NIST2002 corpus, we use one speaker detection cellular data (330 speakers) with  $W = 15$  component densities and approximately 90 s training signals and 30 s test signals. With a complete calculation of (1), i.e. full search, our system has baseline identification accuracies of 99.84%, 70.79% for the 630-speaker TIMIT, NTIMIT corpus as shown by the dashed line in Figs. 2 and 3, respectively. These baseline accuracy rates agree with values published in recent literature [2]. For the NIST2002 corpus, our system has a baseline identification accuracy of 92.42% as shown by the dashed line in Fig. 4.

In order to evaluate the proposed approach, we measure SI accuracy as a function of the percentage of clusters searched. This percentage is an approximation to the search space reduction in (1), since the number of speaker models in each cluster are not exactly the same but are more or less equally-distributed. As in previous work, we use 100 clusters [16] for the TIMIT and NTIMIT corpora. For the NIST2002 corpus, we used 50 clusters.

### 3.1 Evaluation of Clustering and Cluster Selection

Results for TIMIT, NTIMIT and NIST corpora are shown in Figs. 2 - 4 and in Table 1. In evaluating the two measures used in clustering, Euclidean  $k$ -means and log-likelihood  $k$ -means, for a fixed method of cluster selection, log-likelihood  $k$ -means clustering produces higher SI accuracy results. In evaluating the two methods of cluster selection, we find that for the TIMIT, NTIMIT, and NIST2002 corpora, Method #2 generally produces higher SI accuracy results regardless of the measure used in clustering. The best results occur when using the combination of a log-likelihood measure for  $k$ -means clustering and Method #2 for cluster selection. In this case with TIMIT, we are able to search as few as 10%, 20% of the clusters with a 3.7%, 0.6% loss, respectively in SID accuracy; with NTIMIT, these losses are 3.7%, 0.3% loss, respectively and with NIST these losses are 3.0%, 0% respectively. The losses associated with 20% of the clusters are statistically insignificant. We note that searching 10%, 20% of the clusters *reduces* the speaker model space by about a factor of 10, 5 respectively. Finally searching more than 20% of clusters results in the same accuracy as the full search.

### 3.2 Clustering with Prequantization and Pruning

Using the proposed method of speaker model clustering, PQ, and pruning (static algorithm) [2], we carefully measured the actual time for a single speaker identification over several trials using all corpora. The average time for a single SI using no speed-up methods is normalized to  $1.0 \times$  and the speed-up factors with clustering, PQ and pruning are listed in Table 2 and associated accuracies are listed in Table 1. Speed-up gains using only PQ or pruning without speaker model clustering can be evaluated from the data in column 3 of Table

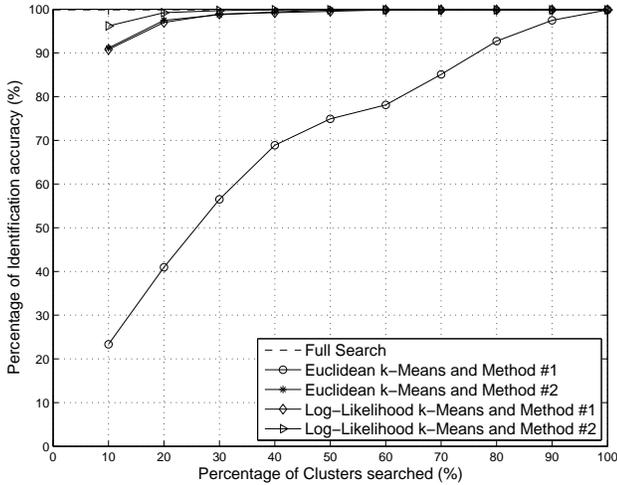


Figure 2: TIMIT SI accuracy vs. % clusters searched.

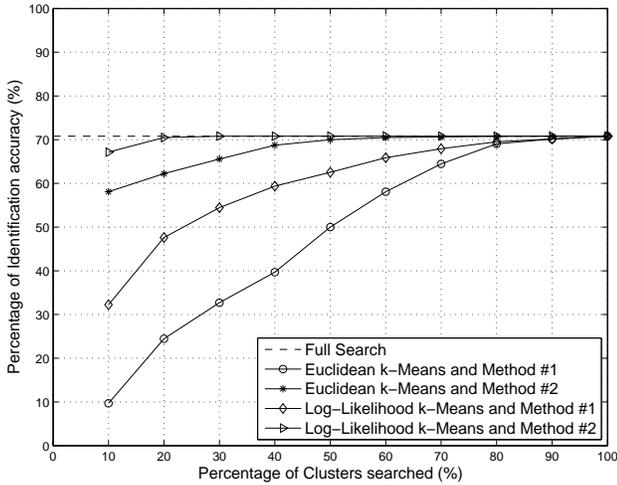


Figure 3: NTIMIT SI accuracy vs. % clusters searched.

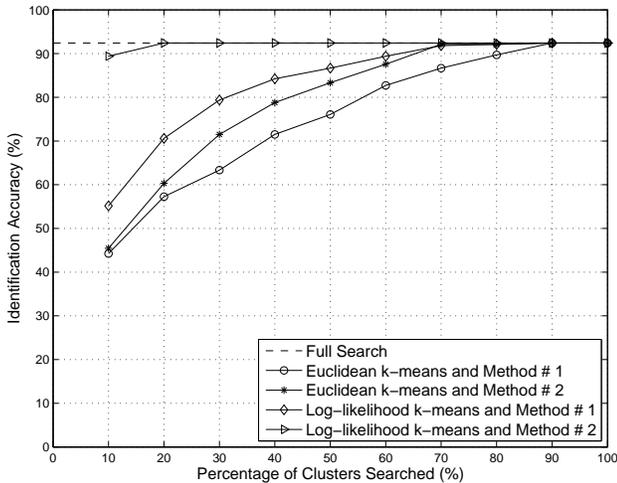


Figure 4: NIST SI accuracy vs. % clusters searched.

2 since utilizing 100% of the clusters amounts to using all speaker models. Although searching 20% of the clusters reduces the search space by a factor of  $5\times$ , due to the small overhead involved in Method #2 for cluster selection, the actual speed-up is  $4.4\times$ . We see from the table that searching 20% of clusters produces similar speed-ups to that of using the entire search space with pruning; combining the proposed method of clustering with PQ results in a real-time speed-up gain of  $63\times$ . The combination of the proposed method of clustering with PQ and pruning results in a speed-up gain of  $74\times$ .

Table 1: SI accuracies for TIMIT, NTIMIT, and NIST.

Corpus	10% of Clusters	20% of Clusters	100% of Clusters
TIMIT	96.2%	99.20%	99.84%
NTIMIT	67.14%	70.47%	70.79%
NIST	89.40%	92.42%	92.42%

Table 2: Average SI speed-up factors relative to baseline system for fixed SI accuracy.

Testing Method	10% of Clusters	20% of Clusters	100% of Clusters
Clustering only	$8.7\times$	$4.4\times$	$1.0\times$
Clustering + Pr	$8.8\times$	$6.6\times$	$4.4\times$
Clustering + PQ	$117.6\times$	$62.5\times$	$14.7\times$
Clustering + PQ + Pr	$149.2\times$	$74.0\times$	$31.6\times$

#### 4. FUTURE WORK

We are investigating a clustering approach which uses KL divergence. However, due to the asymmetry of KL divergence, our proposed methods for selecting clusters may not be appropriate [17]. In this case, a new method for cluster selection would have to be developed. In addition, we are investigating the use of speaker model clustering with GMM-UBM based SI system.

#### 5. CONCLUSIONS

In speaker identification, log-likelihood calculations in the test stage have been recognized as the bottleneck in terms of time complexity. In this paper, we have improved upon our earlier work which utilizes speaker model clusters for reducing the number of speaker models that have to be scored against, thus enabling faster and more efficient SI. In particular we have incorporated log-likelihood measures into the clustering algorithm and proposed new and efficient methods for cluster selection. Compared to other methods, our clustering approach scales well for large populations. For the TIMIT, NTIMIT and NIST corpora, we are able to search as few as 20% of the speaker model space and incur an insignificant loss in SI accuracy; Finally, SI times are given using the proposed clustering method together with other speed-up methods such as, pruning and pre-quantization resulting in actual speed-up factors as high as  $74\times$ . Higher speed-up factors (up to  $150\times$ ) are possible using 10% of the clusters but with slight decrease in SI accuracy.

## REFERENCES

- [1] D. Reynolds and R. Rose, "Robust text-independent speaker identification using gaussian mixture speaker models," *IEEE Trans. Signal Process.*, vol. 3, no. 1, pp. 72–83, Jan. 1995.
- [2] T. Kinnunen, E. Karpov, and P. Franti, "Real-time speaker identification and verification," *IEEE Trans. Audio, Speech, and Language Process.*, vol. 14, no. 1, pp. 277–288, Jan. 2006.
- [3] T. F. Quatieri, *Discrete-Time Speech Signal Processing Principles and Practice*. Prentice-Hall, Inc., 2002.
- [4] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted gaussian mixture models," *Digital Signal Processing*, vol. 10, pp. 19–41, 2000.
- [5] H. Aronowitz and D. Burshtein, "Efficient speaker recognition using approximated cross entropy (ACE)," *IEEE Trans. Audio, Speech, and Language Process.*, vol. 15, no. 7, pp. 2033–2043, Sep. 2007.
- [6] J. Makhoul, F. Kubala, T. Leek, L. Daben, N. Long, R. Schwartz, and A. Srivastava, "Speech and language technologies for audio indexing and retrieval," *Proc. IEEE*, vol. 88, no. 8, pp. 1138–1353, Aug 2000.
- [7] H. Aronowitz, D. Burshtein, and A. Amir, "Speaker indexing in audio archives using test utterance gaussian mixture modeling," in *Proc. Int. Conf. Spoken Lang. Proc. (ICSLP)*, 2004.
- [8] J. McLaughlin, D. A. Reynolds, and T. Gleeson, "A study of computation speed-ups of the GMM-UBM speaker recognition system," in *Proc. 6th European Conf. Speech Communication and Technology (Eurospeech)*, 1999, pp. 1215–1218.
- [9] B. L. Pellom and J. H. L. Hansen, "An efficient scoring algorithm for gaussian mixture model based speaker identification," *IEEE Signal Process. Lett.*, vol. 5, no. 11, pp. 281–284, Nov. 1998.
- [10] B. Sun, W. Liu, and Q. Zhong, "Hierarchical speaker identification using speaker clustering," in *Proc. Int. Conf. Natural Language Processing and Knowledge Engineering*, 2003.
- [11] W. Tsai, S. Cheng, and H. Wang, "Automatic speaker clustering using a voice characteristic reference space and maximum purity estimation," *IEEE Trans. Audio, Speech, and Language Process.*, vol. 15, no. 4, pp. 1461–1474, May 2007.
- [12] A. M. W. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, "Content-based image retrieval at the end of the early years," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 22, no. 12, pp. 1349–1380, Dec. 2000.
- [13] L. J. Rodriguez and M. I. Torres, "A speaker clustering algorithm for fast speaker adaptation in continuous speech recognition," *Lecture Notes in Computer Science: Text, Speech and Dialogue*, vol. 3206/2004, 2004, springer.
- [14] P. Angkititrakul and J. Hansen, "Discriminative in-set/out-of-set speaker recognition," *IEEE Trans. Audio, Speech, and Language Process.*, vol. 15, no. 2, pp. 498–508, Feb. 2007.
- [15] D. Reynolds, "Large population speaker identification using clean and telephone speech," *IEEE Signal Process. Lett.*, vol. 2, no. 3, pp. 46–48, Mar. 1995.
- [16] P. L. De Leon and V. Apsingekar, "Reducing speaker model search space in speaker identification," in *Proc. IEEE Biometrics Symposium*, 2007.
- [17] J. Hershey and P. Olsen, "Approximating the Kullback Leibler divergence between Gaussian mixture models," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing (ICASSP)*, 2007.