REDUCING SPEAKER MODEL SEARCH SPACE IN SPEAKER IDENTIFICATION

Phillip L. De Leon and Vijendra Apsingekar

New Mexico State University
Klipsch School of Electrical and Computer Engineering
Las Cruces, New Mexico USA 88003
{pdeleon, vijendra}@nmsu.edu

ABSTRACT

For large population speaker identification (SID) systems, likelihood computations between an unknown speaker's test feature set and speaker models can be very time-consuming and detrimental to applications where fast SID is required. In this paper, we propose a method whereby speaker models are clustered during the training stage. Then during the testing stage, only those clusters which are likely to contain high-likelihood speaker models are searched. The proposed method reduces the speaker model space which directly results in faster SID. Although there maybe a slight loss in identification accuracy depending on the number of clusters searched, this loss can be controlled by trading off speed and accuracy.

1. INTRODUCTION

The objective of speaker *identification* (SID) is to determine which voice sample from a set of known voice samples best matches the characteristics of an unknown input voice sample [1]. SID is a two-stage procedure consisting of training and testing. In the training stage shown in Fig. 1(a), speaker-dependent feature vectors, $\mathbf{Y}_m$ are extracted from a training speech signal and a speaker model, λ_s is built for each speaker's feature set. In the testing stage shown in Fig. 1(b), feature vectors $\mathbf{Y}_m^{\text{test}}$ are extracted from a test signal (speaker unknown). The test feature set is compared and scored against all S speaker models and the most likely speaker identity, $\hat{s}$ is decided. Of the various speaker modelling techniques, the Gaussian Mixture Model (GMM) based approach has shown to be very successful in accurately identifying speakers from a large population [1]. GMMs provide a probabilistic model of the distribution of feature vectors. A standard approach in estimating the parameters of the GMM (weights, mean vectors, and covariance matrices) is to use the Expectation Maximization (EM) algorithm [1]. In assessing an SID system we measure the identification accuracy, computed as the number of correct identification tests divided by the total number of tests.

In speaker *verification* (SV), MAP-adapted speaker models from an universal background model (UBM) with likelihood normalization are normally used, however UBMs are not used in SID [2]. There are also more advanced matching techniques like SVM-GLDS [3] and SVM-supervectors [4] which are also used in SV but use in SID has not yet been investigated.

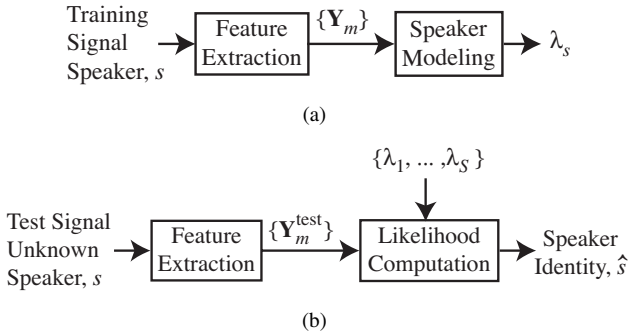


Fig. 1. (a) Training and (b) testing stages in SID

In this paper, we consider the problem of fast identification for large population SID systems. In such systems, likelihood computations between an unknown speaker's test feature set and all speaker models can be very time-consuming and detrimental to applications where fast SID is required. This can be very useful in speaker classification and speaker clustering applications where the exact identity of the speaker is not necessary but rather identifying the speaker class is sufficient [5].

The slow SID problem has been recognized and investigated as has a similar problem in speaker verification (SV); in this paper our focus is *strictly* on SID. Two previously proposed methods to speed-up SV and SID are pre-quantization (PQ) and pruning. In PQ, the test feature set is first compressed or reduced through downsampling (or another method) before likelihood computations [6]; a smaller feature set directly translates into faster verification. It has been found that reducing the test feature set by a factor as high as 20, does not affect SV performance. Application of PQ in

order to speed-up SID has been investigated in [5] and results in a speed-up factor of as high as $5\times$ with no loss in identification accuracy using the TIMIT corpus (clean speech, low noise and minimal channel distortions). In pruning, a small portion of the test feature set is compared against all speaker models [7]. 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 [5]. Variants of PQ and pruning as well as combinations of the methods have been recently evaluated in [5].

In this paper, we propose a method whereby speaker models are clustered according to their statistical similarity during the training stage. Then during the testing stage, only those clusters which are likely to contain high-likelihood speaker models are searched. The proposed method reduces the speaker model space which directly results in faster SID. Although there maybe a slight loss in identification accuracy depending on the number of clusters searched, this loss can be controlled by trading off speed and accuracy. Finally, our method can easily be combined with PQ and pruning for additional speed increases, however, in this paper we evaluate our method on a stand-alone basis.

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) [8], [9], [10]. In this application, only those images within a few pre-determined classes that are similar to the query image are searched rather than searching the entire image database. The use of speaker clusters has been used for fast speaker adaptation in speech recognition applications [11]. Here, the speaker adaptation methodology first determines speaker clusters in the training data, then estimates corresponding model parameters and applies a matching strategy to choose the optimal cluster for each test utterance. The use of clusters or classes for speaker recognition has also been used in the open-set speaker identification (OSI) problem. In this problem, the objective is to classify an unknown speaker into a predefined class of speakers or to recognize that the speaker does not belong to any class [12]. As far as applying speaker model clustering to the problem of slow SID, there does not appear to be any reported work in open literature.

This paper is organized as follows. In Section 2, we describe the GMM-based SID system. In Section 3, we describe our method of speaker model clustering and how speaker identification proceeds once a test signal is acquired. In Section 4, we describe the experimental evaluation and provide results using both the TIMIT and NTIMIT (telephone-quality speech) corpora; these corpora are two of the most common, large population speech databases used in SID research. In Section 5, we briefly describe possible future work. Finally, in Section 6 we conclude the article.

2. GMM-BASED SPEAKER IDENTIFICATION

2.1. Feature Extraction

Fig. 2 illustrates the steps involved in the feature extraction blocks of Fig. 1. First, silence is removed from the utterance and then the short-time Fourier transform (STFT), $X(m,k)$ is computed. In this work, the STFTs (1024-point) are computed using 20 ms Hamming-windowed segments with 50% overlap. Magnitude-squared data is computed from the STFT and weighted according to a mel-scale filterbank. The L-channel filterbank is designed with triangular responses over each frequency band and the filters, F_l are normalized according to their bandwidth. The log-energy at block time m for the lth channel, $y_l(m)$ is calculated and the DCT of the vector, $\mathbf{y}_m = [y_0(m), \cdots, y_{L-1}(m)]^T$ is computed for further decorrelation. Each resulting $L \times 1$ feature vector contains the mel-frequency cepstral coefficients (MFCCs), $\mathbf{Y}_m$ computed every 10 ms.

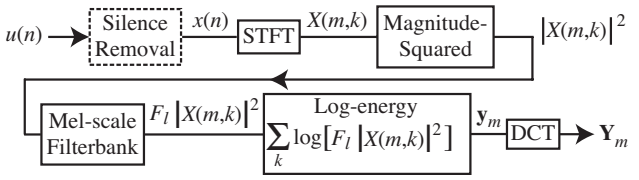


Fig. 2. Mel-scale cepstral feature analysis

2.2. Training the Speaker Identification System

The next step in building the SID system, is to statistically model each speaker's feature set, $\{\mathbf{Y}_m\}$. For this, we assume the probability density function (pdf) for the feature vector $\mathbf{Y}$ given speaker model λ_s can be modeled as a weighted mixture of Gaussian pdfs

$$p(\mathbf{Y}|\lambda_s) \quad = \quad \sum_{i=1}^{W} w_i p_i(\mathbf{Y}) \tag{1}$$

where W is the number of mixture components, w_i is the weight of the ith mixture component, and $p_i(\mathbf{Y})$ is the ith component density. In the GMM, the weights are constrained to sum to one. The component density in (1) is an L-dimension Gaussian pdf of the form

$$p_i(\mathbf{Y}) \quad = \quad \frac{1}{(2\pi)^{L/2}|\mathbf{\Sigma}_i|^{1/2}} \times \exp\left\{-\frac{1}{2}\left(\mathbf{Y}-\boldsymbol{\mu}_i\right)^T \mathbf{\Sigma}_i^{-1}\left(\mathbf{Y}-\boldsymbol{\mu}_i\right)\right\} \tag{2}$$

where $\boldsymbol{\mu}_i$ is the mean vector and $\mathbf{\Sigma}_i$ is the covariance matrix (assumed to be diagonal) corresponding to the ith mixture component. The weights, mean vectors, and covariance matrices collectively form the speaker model, λ_s [1].

ML estimation of λ_s is a difficult nonlinear optimization problem therefore iterative techniques, such as the EM algorithm, have been employed that guarantee convergence to local minima [1]. The EM algorithm begins with an initial estimate of λ_s, and improves this estimate until some convergence threshold is reached.

The EM update equations for a given speaker model, λ for the ith weight, mean vector, and covariance vector $\boldsymbol{\sigma}_i = \mathrm{diag}(\boldsymbol{\Sigma}_i)$ for $1 \leq i \leq W$ are given by

$$w_i = \frac{1}{M} \sum_{m=1}^{M} p(i|\mathbf{Y}_m, \lambda), \tag{3}$$

$$\boldsymbol{\mu}_i = \frac{\sum_{m=1}^{M} p(i|\mathbf{Y}_m, \lambda)\mathbf{Y}_m}{\sum_{m=1}^{M} p(i|\mathbf{Y}_m, \lambda)}, \tag{4}$$

$$\boldsymbol{\sigma}_i = \frac{\sum_{m=1}^{M} p(i|\mathbf{Y}_m, \lambda)\mathbf{Y}_m^2}{\sum_{m=1}^{M} p(i|\mathbf{Y}_m, \lambda) - \boldsymbol{\mu}_i^2} \tag{5}$$

where M is the number of training feature vectors, the *a posteriori* probability for the ith acoustic class is given by

$$p(i|\mathbf{Y}_m, \lambda) = \frac{w_i p_i(\mathbf{Y}_m)}{\sum_{i=1}^{W} w_i p_i(\mathbf{Y}_m)}, \tag{6}$$

and $\mathbf{Y}_m^2$ and $\boldsymbol{\mu}_i^2$ denote element-by-element squaring of the vector. The EM algorithm terminates when improvement of $\{w_i, \boldsymbol{\mu}_i, \boldsymbol{\sigma}_i\}$ at the current and previous iterations saturates.

2.3. Testing the Speaker Identification System

Once all S speaker models are obtained, SID proceeds through a maximum likelihood (log-likelihood) detection. We assume that all enrolled speakers are equally likely and that the test feature vectors are independent. In this case, it is well-known that maximum a posteriori (MAP) detection becomes the maximum likelihood (ML) detection for identification of the unknown speaker

$$\hat{s} = \arg \max_{1 \leq s \leq S} \prod_{m=1}^{M'} p(\mathbf{Y}_m^{\mathrm{test}}|\lambda_s) \tag{7}$$

or equivalently

$$\hat{s} = \arg \max_{1 \leq s \leq S} \sum_{m=1}^{M'} \log p(\mathbf{Y}_m^{\mathrm{test}}|\lambda_s) \tag{8}$$

where M' is the number of test feature vectors. Note that for test signals which are a few seconds long, (8) can require a great deal of computation if S is large.

3. SPEAKER MODEL CLUSTERING

In a SID system for a large and acoustically-diverse population, only a few speaker models actually give large log-likelihood values for (8). In fact, the basis for speaker pruning, is to quickly eliminate speaker models for which it is clear the log-likelihood score is going to be low thus reducing unnecessary computation in (8). We propose that after the training stage, the resulting speaker models should be clustered according to the pdf they represent and during the test stage, only those clusters likely to contain a high-scoring speaker model should be considered when computing (8). The following subsections describe how to accomplish this.

3.1. k-means Algorithm

The k-means algorithm is one of the simplest unsupervised clustering algorithms available [13]. The algorithm steps are as follows.

Algorithm 1 k-means Algorithm

1: Initially choose k cluster centroids $z_i(1), z_2(1), ..., z_k(1)$. These are arbitrary and are usually selected as the first k data points x of the set.
2: At the I^{th} iteration, distribute the data points x among the k cluster domains, using the relation,

$$x \in S_j(I) \text{ if } \|x - z_j(I)\| < \|x - z_i(I)\| \tag{9}$$

for all $i = 1, 2, ..., k,\ i \neq j$, where $S_j(I)$ denotes the set of data points whose cluster centroid is $z_i(I)$.
3: Compute the new cluster centroids $z_j(I+1)$, $j = 1, 2, ..., k$, such that the sum of squared distances from all points in $S_i(I)$ to $z_i(I+1)$ is minimized.
4: If $||z_j(I+1) - z_j(I)|| \leq \varepsilon$ for $j = 1, 2, ...k$ and small ε, the algorithm has converged and procedure is terminated. Otherwise, go to Step 2.

3.2. Data Vectors for Clustering

We use the k-means algorithm to cluster speaker models according to their weighted mean vector (WMV)

$$\bar{\boldsymbol{\mu}} = \sum_{i=1}^{W} w_i \boldsymbol{\mu}_i \tag{10}$$

or according to their covariance-normalized, weighted mean vector (NWMV)

$$\bar{\boldsymbol{\mu}}^{\mathrm{N}} \quad = \quad \sum_{i=1}^{W} w_i \boldsymbol{\Sigma}_i^{-1} \boldsymbol{\mu}_i. \tag{11}$$

The WMV can be thought of as the mean of the overall pdf represented by the GMM or "center" of the speaker model. A clustered speaker model space is illustrated in Fig. 3.

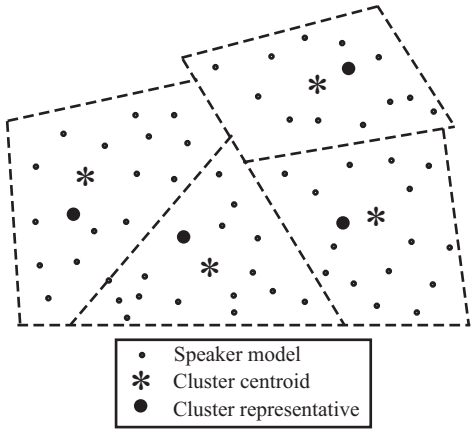


Fig. 3. Space of speaker models, clusters, and representatives.

3.3. Cluster Selection during the Testing Stage

For the test stage, we consider two ways in which to select a pre-determined percentage of clusters for evaluation of (8). In Method #1 shown in Fig. 4(a), a GMM is computed on the test feature set and those clusters (as represented by their centroids) nearest to the test (N)WMV are searched. In Method #2, prior to testing, we identify the speaker model in each cluster which is nearest to the centroid and call it the "cluster representative" (see Fig. 3). When the test signal is acquired, (8) is computed against all cluster representatives' speaker models as shown in Fig. 4(b) and clusters with the highest-scoring representatives are then searched. Method #1 clearly directs the search toward the clusters with candidate speaker models but requires computation of a test GMM; on the other hand, Method #2 does not require computation of a test GMM but is dependent on how well the representatives actually represent the speaker models in the cluster. For short test signals, there is the possibility with Method #1, that the EM alogrithm may not properly converge leading to an inaccurate test GMM and therefore may not direct our search to the proper cluster(s). Note that the cluster centroid itself is of no use in computing (8) since it does not have the necessary GMM parameters.

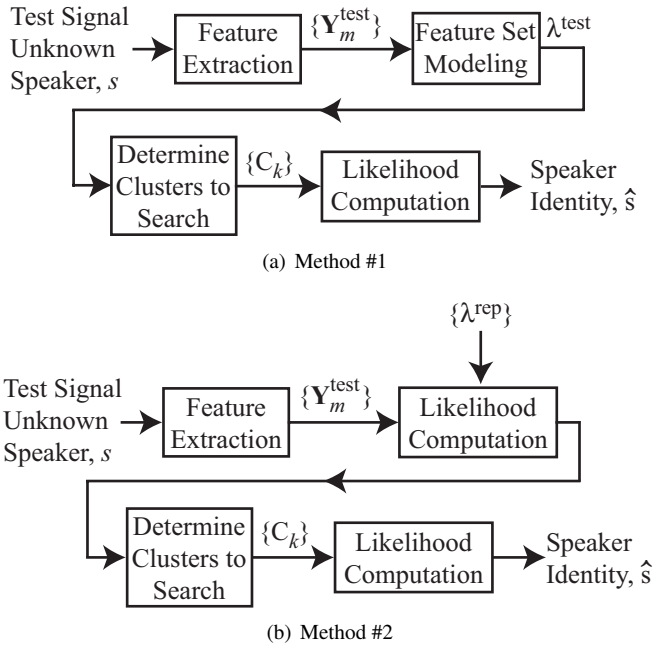


Fig. 4. Determining which speaker model clusters to search. (a) Method #1 uses a GMM from the test feature set and cluster centroids and (b) Method #2 compares the test feature set to cluster representatives' speaker models.

4. RESULTS

For the TIMIT corpus, we use a mel-scale filterbank which has the first nine center frequencies uniformly spaced from 100-1000 Hz and the next twenty center frequencies logarithmically spaced from 1000-8000 Hz resulting in a 29×1 feature vector. For the NTIMIT corpus (telephone-quality speech), the mel-scale filterbank which has the first seven center frequencies uniformly spaced from 300-1000 Hz and the next thirteen center frequencies are logarithmically spaced from 1000-3400 Hz resulting in a 20×1 feature vector [1]. In addition, we use $W = 15$ mixtures for the GMM as in [1]. With approximately 24 s training, 6 s test signals, and complete calculation of (8), i.e. full search, our SID system has baseline identification accuracies of 99.84%, 68.73% for the 630-speaker on TIMIT, NTIMIT corpus respectively. These baseline accuracies agree closely with those published in the current literature for TIMIT [5], and for NTIMIT [14].

In order to determine the SID accuracy when utilizing speaker model clusters, we evaluated WMV- and NWMV-based clustering and Method #1 and #2 for cluster selection. We measure SID accuracy as a function of the percentage of clusters searched. This percentage is an approximation to the search space reduction in (8), since the number of speaker models in each cluster are not exactly the same but are more or less equally-distributed.

Fig. 5 provides a comparison between the WMV and NWMV clustering methods using a total of 10 clusters and testing, with Method #1, over a selected subset of these. When using WMV and searching 50% of the clusters, identification accuracy decreases by 8.9%, 4.4% for the TIMIT, NTIMIT corpus respectively; when using NWMV these decreases are 4.2%, 15.2% (as compared to the baseline results). As a higher percentage of clusters are searched, accuracy increases up to the baseline results and thus any losses in accuracy can be controlled by trading off the number of clusters searched (search time) and accuracy. We find that NWMV produces better results for TIMIT (probably due to additional "spreading" of the speaker models) while the WMV produces better results for NTIMIT. For the remaining experiments, all TIMIT speaker models are clustered according to their NWMV in (11) while those for NTIMIT are clustered according to their WMV in (10).

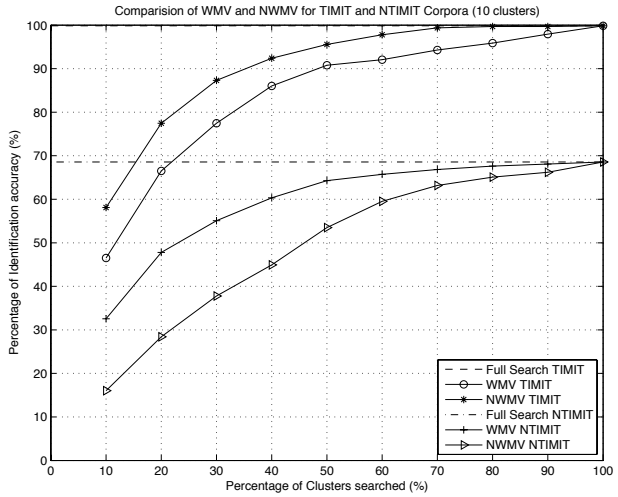
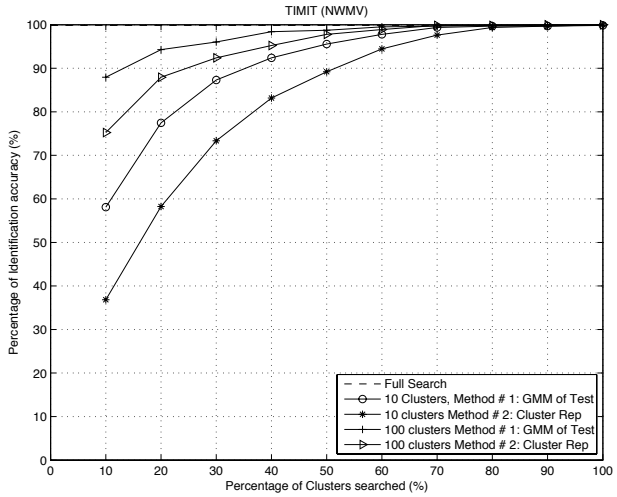
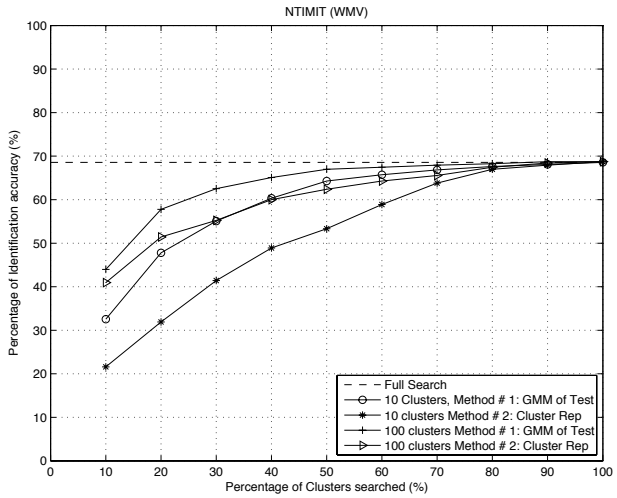


Fig. 5. Identification accuracies for the WMV and NWMV methods of clustering using ten clusters

Fig. 6 provides comparisons between Method #1 and #2 for cluster selection as well as for speaker model spaces partitioned into 10 and 100 clusters. For both corpora, there is a clear advantage to having more clusters available for testing. For TIMIT, we find the best performance with Method #1 and 100 clusters. In this case, we are able to search as few as 30%, 50% of the clusters with only a 3.7%, 1.0% loss, respectively in SID accuracy; searching 30%, 50% of the clusters reduces the speaker model space by about $1/3$, $1/2$ respectively. For NTIMIT, we again find the best performance with Method #1 and 100 clusters. In this case, we are able to search as few as 30%, 50% of the clusters with only a 6.2%, 1.7% loss, respectively in SID accuracy. These search space reductions directly translate into speed-up gains as good as pruning methods and PQ. In our research, we find that Method #1 results in a good balance between search space reduction and accuracy when the number of clusters is large (more than 70). On the other hand, Method #2 is good when number of clusters is smaller (less than 50).



(a)



(b)

Fig. 6. Identification accuracies versus the percentage of clusters searched for (a) TIMIT and (b) NTIMIT corpora.

An interesting side-effect was observed using cluster-based searching on the TIMIT corpus. When searching 70%-90% of the search space (100 clusters) using both Method #1 and #2, identification accuracy *increased* above the baseline (full search) accuracy—even to 100% in a couple of cases. The reason it is possible to increase accuracy with speaker model cluster clustering is that a speaker model which leads to an incorrect identification during a full search may not be present in the clusters which are being searched and thus not produce the incorrect identification. A similar effect could in theory occur with pruning (speaker model which could lead to an incorrect identification is pruned out early).

5. FUTURE RESEARCH

There are some possibilities for further improvements in reducing the search space in SID for fast identification. Since the proposed speaker model clustering is performed during the training stage, it can be combined with existing test-stage methods such as PQ and pruning (as mentioned in Section 1). The combination of speaker clustering with PQ and pruning could increase the SID times dramatically although it is not clear if accuracy would be significantly degraded. Another possibility includes utilization of a log-likelihood measure in the clustering procedure rather than a distance measure based on (N)WMV. This could potentially increase accuracy since the actual identification is based on a log-likelihood score. However, a direct method of determining clusters, taking into account all speaker models and training feature sets leads to a difficult nonlinear optimization problem [15]. Finally this research can be extended with other speech corpora like NIST which require channel compensation techniques like feature warping, factor analysis etc.

6. CONCLUSIONS

In SID, the testing stage requires log-likelihood calculations (scores) of the unknown speaker's test signal against all speaker models in the system. We have proposed the use of speaker model clusters for reducing the number of speaker models that have to be scored against, thus enabling faster SID. Using TIMIT, NTIMIT corpora and searching only 50% of speaker model clusters (search space) results in a 1.0%, 1.7%, respectively loss in SID accuracy. Greater reductions can made at the expense of SID accuracy which can be controlled and may be acceptable in applications where speaker-class identity and not the exact speaker identity is required. In some cases, the use of speaker model clusters resulted in slightly *higher* identification accuracy rates. Finally, our method can be combined with methods such as prequantization and pruning.

7. REFERENCES

[1] D. Reynolds, "Large population speaker identification using clean and telephone speech," *IEEE Signal Process. Lett.*, vol. 2, no. 3, pp. 46–48, Mar. 1995.

[2] 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.

[3] W. M. Campbell, D. E. Sturim, D. A. Reynolds, and J. Navratil, "The MIT-LL/IBM 2006 speaker recognition system: high-performance reduced-complexity recognition," *Proc. Int. Conf. on Acoustic, Speech and Signal Processing (ICASSP)*, 2007.

[4] W. M. Campbell, D. E. Sturim, and D. A. Reynolds, "Support vector machines using GMM supervectors for speaker verification," *IEEE Signal Process. Lett.*, vol. 13, no. 5, pp. 308–311, May 2006.

[5] 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.

[6] 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)*, 1999.

[7] 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.

[8] C. Faloutsos, R. Barber, M. Flickner, J. Hafner, W. Niblack, D. Petkovic, and W. Equitz, "Efficient and effective querying by image content," *J. Intelligent Information Systems*, vol. 3, no. 3/4, pp. 231–262, 1994.

[9] 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.

[10] R. O. Stehling, M. A. Nascimento, and A. X. Falcao, "An adaptive and efficient clustering-based approach for content-based image retrieval in image databases," in *2001 Int. Symp. Database Engineering & Applications*, 2001.

[11] 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.

[12] P. Angkititrakul and J. Hansen, "Discriminative in-set/out-of-set speaker recongition," *IEEE Trans. Audio, Speech, and Language Process.*, vol. 15, no. 2, pp. 498–508, Feb. 2007.

[13] T. F. Quatieri, *Discrete-Time Speech Signal Processing Principles and Practice*. Prentice-Hall, Inc., 2002.

[14] D. J. Mashao, "A hybrid GMM-SVM speaker identification system," *IEEE Africon*, Sep. 2004.

[15] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. Kluwer Academic Publishers, 1992.