International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 3 Issue 3, March - 2014

# Reducing the Computational Complexity of the GMM-UBM Speaker Recognition Approach

Farnaz Ganjeizadeh
Dept. of Engineering,
CSU – East Bay, Hayward, CA

Howard Lei
Dept. of Engineering
CSU – East Bay, Hayward, CA

Andrew Maganito
Dept. of Engineering
CSU – East Bay, Hayward, CA

Gopikrishnan Pallipatta
Dept. of Engineering
CSU – East Bay, Hayward, CA

***Abstract*—Speaker recognition approaches rely heavily on the use of Gaussian Mixture Models (GMMs) for speaker modelling. The models can represent arbitrary distributions of feature vectors extracted from speech waveforms, and are easy to train. Traditional GMM-based implementations are computationally complex, utilizing large numbers of parameters. These approaches have typically been used in "offline" settings, where results are not generated in real-time. This work seeks to reduce the computational complexity of the GMM-UBM approach, allowing it to produce results in shorter amounts of time. The preliminary results of this study demonstrate the feasibility of reducing the number of parameters while maintaining speaker recognition performance. This work provides a foundation for further work in reducing computational complexity, which we intend to use for future real-time hardware implementations.**

***Keywords—Speaker Recognition; GMM-UBM***

## I. INTRODUCTION

Speaker recognition has been an established area of research for the past 15 years, and involves the application of signal processing, statistical, and machine learning algorithms to the recognition of speaker identities in audio recordings. The technology is applicable to high-tech applications, such as voice-based biometrics[1], and forensics [2][3]. The traditional speaker recognition approach that has been widely popular until around 2007 is one that uses Gaussian mixture models (GMMs) to model the feature vectors extracted from the speech waveforms of speakers [4]. It is referred to as the GMM-UBM approach, which involves the use of a Universal Background Model (UBM) to represent feature vectors from a large set of speakers. The feature vectors are extracted using acoustic signal processing techniques. While more recent approaches for speaker recognition have relied on advanced techniques such as Joint Factor Analysis (JFA) [5] and i-vectors [6][7], the classical approach involving GMM models is still viable in environments where speaker data is limited. This is because the JFA and i-vector techniques rely on large amounts of development data for modelling purposes. Such data, especially those matching the noise and recording environments of the target data, are not always available,

The GMM models consist of a mixture of multivariate Gaussians probability distributions which are easy to obtain (or train) using the feature vectors. Given limited knowledge of the data, GMMs can model feature vector distributions that are difficult to precisely characterize, such as feature vectors resulting from speech waveforms. GMM models are trained using the Expectation-Maximization (EM) algorithm, an iterative algorithm that finds a maximum-likelihood estimate of the model parameters given the feature vectors [8]. The EM algorithm is similar to the K-means clustering algorithm [9], except that it uses soft clustering assignments. In soft clustering, each feature vector is assigned a likelihood of belonging to each GMM mixture. The mixture means, covariance's, and weights are updated based on the likelihoods of its MFCC vectors.

In every speaker recognition system, UBM is needed to represent the distribution of a general population of speakers [4]. The UBM is a speaker-independent GMM model that is used for score normalization and speaker-dependent GMM training, and is itself trained using the EM algorithm given feature vectors from a large number of speakers. GMM models are widely used not only for the classical GMM-UBM approach but also for the more advanced JFA and i-vector approaches. The i-vector approach seeks to obtain low-dimensional vectors from speech waveforms representing speaker "voiceprints" [6][7]. A UBM is used for the statistical algorithms.

The standard approaches such as the GMM-UBM are computationally complex, which prevents them for performing real-time voice processing. They have traditionally been designed to process the voice recordings "offline," meaning that a waiting period is required before the system can to obtain the voice identities. This is because the systems often utilize millions of parameters, requiring many megabytes of memory and billions of algebraic computations per audio recording. One way in which the speaker recognition approaches can be improved is to allow them to perform better "real-time" voice processing by finding effective ways to reduce the number of parameters and computations needed. Such reductions would also enable more effective implementations on small-scale hardware platforms, where the memory and computational resources are limited. The aim of this work is to first set up and investigate existing complex

speaker recognition approaches to find ways in which the number of parameters and computational complexity can be effectively reduced. Based on the findings, the next step is to implement a speaker recognition approach suitable for small-scale embedded hardware platforms.

The article is structured as followed: Section 2 discusses the data collected, and Section 3 describes the baseline GMM-UBM approach. Section 4 describes the methodologies used to reduce the computational complexity and speaker recognition performance measures, and Section 5 describes the experiments and results, and provides a discussion. Section 6 provides a summary and discussion of future work.

## II. DATA

The data consists of recorded speech from California State University, East Bay students and faculty. The recorded subjects include nine females and 33 males. Each subject was asked to read two paragraphs carefully selected from a textbook containing numbers and simple wording. The duration of the first paragraph takes roughly two minutes to read, while the second paragraph takes roughly one minute. The recordings were taken using a Blue Snowball USB microphone with the omni-directional microphone setting, with a sampling frequency of 44,100 samples per second. The total amount of speech used in all experiments is roughly one hour. We note that the dataset we used using is significantly smaller compared to standard datasets, such as the NIST Speaker Recognition Evaluation Datasets [10].

## III. BASELINE GMM-UBM APPROACH

The baseline for our experiments is the classical GMM-UBM approach, which is based on training GMMs to model the distribution of feature vectors of extracted from speech waveforms. The feature vectors are Mel-Frequency Cepstral Coefficients (MFCCs) C0-C19, a total of 20 dimensions. In addition, the first and second time derivatives of the coefficients of each feature vector dimension are appended to generate vectors of 60 dimensions. The typical feature extraction approach extracts one MFCC feature vector for every 10ms of speech using 25ms windows of speech, such that an entire speech waveform is represented by a sequence of feature vectors. Every minute of speech should hence contain 100 vectors. Each MFCC feature vector dimension is mean and variance normalized across the duration of each waveform. Because our work is focused on the modelling approaches and not on the MFCC feature vectors, we will omit a full description of the feature vector extraction process from the acoustic and signal processing standpoint. For those interested, the work of [11] describes the MFCC features in detail.

The GMM-UBM approach involves first training a UBM via the EM algorithm on a set of speech data across multiple speakers. The UBM represents the speaker-independent model. In our particular implementation of the system, speaker-dependent GMM models are trained using the EM algorithm from each speaker's data, and the UBM is used to initialize the algorithm. The following equation describes the probably density function (pdf) of a GMM model:

$$p(x;\mu,\Sigma,\omega)=\sum_{m=1}^{M}\omega_m N(x;\mu_m,\Sigma_m) \quad (1)$$

where **x** is a vector, N(x; $\mu_m$, $\Sigma_m$) is a pdf of a Gaussian distribution with mean $\mu_m$ and covariance matrix $\Sigma_m$, and $\omega_m$ are the mixture weights. M is the number of Gaussian mixtures. The UBM is trained using the first paragraph of speakers 1-10, while the speaker-dependent models are trained using the first paragraphs of each speaker. Hence, the first paragraphs of each speaker comprise the training data.

In our experiments, we used eight mixtures for each GMM (M=8), with full covariance matrices. The number of mixtures is small compared to those used in a typical GMM-UBM system, with 512 to 2,048 mixtures. However, the dataset we are using (1 hour of total speech) is also significantly smaller compared to the typical datasets, and hence fewer mixtures are needed. Figure 1 illustrates the GMM training process.

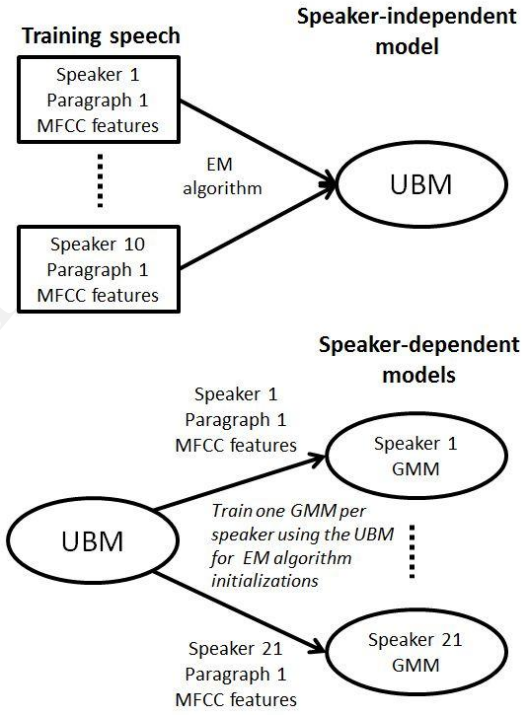


Fig. 1. Training of Speaker-Independent UBM and speaker-dependent GMM models

Once the speaker-dependent GMM models were trained, data from the second paragraphs of each speaker (i.e. the test data) was used to generate test MFCC feature vectors for each speaker. The test feature vectors are scored against each speaker-dependent model. Specifically, given a speaker A for which test MFCC feature vectors are generated, and a speaker B for which a speaker-dependent GMM is generated, a log-likelihood ratio (LLR) is computed to generate a speaker-similarity score, as shown in the equation below:

$$score(A,B)=\log\left(\frac{\sum_{i=0}^{N}p(x_{Ai};\mu_B,\Sigma_B,\omega_B)}{\sum_{i=0}^{N}p(x_{Ai};\mu_{UBM},\Sigma_{UBM},\omega_{UBM})}\right) \quad (2)$$

where p(•) is a pdf of a GMM, score(A,B) is the similarity score between speakers A and B, $\mu_B$, $\Sigma_B$, and $\omega_B$ are the parameters of the GMM trained for speaker B, and $\mu_{UBM}$, $\Sigma_{UBM}$, and $\omega_{UBM}$ are the parameters of the UBM. $x_{Ai}$ is MFCC feature vector i from the test data (second paragraph) of speaker A, which has a total of N feature vectors. Figure 2 illustrates score computation.

Note that our implementation of the GMM-UBM speaker recognition system was completed using publically available MATLAB scripts under the BSD license.

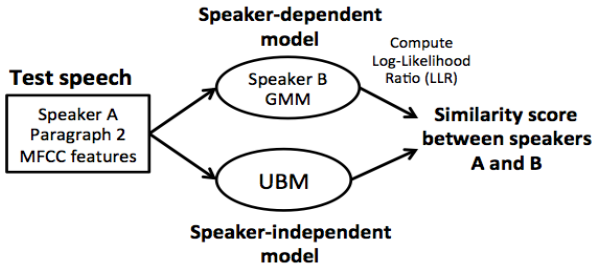


Fig. 2. Log-Likelihood Ratio (LLR) speaker similarity score computation

## IV. Methodologies and Performance Measures

The goal of this work is to reduce the computational complexity of the baseline GMM-UBM speaker recognition approach. This was achieved by examining the importance of different parameters and seeing how the number of parameters can be reduced based on their relative importance to the performance results. The following parameters were examined to evaluate their individual effects on performance.

- MFCC analysis window (Tw): 25 ms
- MFCC feature vector shift (Ts): 10 ms
- MFCC feature vector dimension (N): 60
- GMM-UBM mixtures (M): 8
- UBM training data (D): Paragraph 1 of speakers 1-10

Three of the parameters (Ts, M, and D) allowed us to reduce computational complexity while achieving comparable results to the baseline approach. Results were obtained using the following two performance measures, which are widely used in Speaker Recognition and Identification research.

- Closed-Set Speaker Identification Accuracy
- Equal Error Rate (EER)

The closed-set speaker identification accuracy is the percentage that the test data's speaker (test speaker) is correctly identified given the set of all speakers in the dataset, and the knowledge that the test speaker is included among the set of all speakers. A test speaker is correct identified if the LLR score is highest for the speaker-dependent GMM of the same speaker. The higher the accuracy percentage, the better the speaker recognition approach.

The second performance measure is the Equal Error Rate (EER). The EER occurs at a scoring threshold where the rate at which non-target speaker scores are misclassified as target speaker scores (false alarms), equals the rate at which target speaker scores are misclassified as non-target speaker scores (misses). The lower the EER, the better the speaker recognition approach at separating the target and non-target speaker scores.

Note that 42 speakers are used for all experiments in this work, and each speaker provides both training and test data. Hence, the speaker recognition approaches all generate 42 target speaker scores, and 1,722 non-target speaker scores (42*42 = 1,764; 1764-42 = 1,722), where each test data is scored against every speaker-dependent model obtained from the training data.

## V. Experiments, Results and Discussions

We first implemented the GMM-UBM baseline approach with the parameter values discussed in Section 5. The following table shows the results of the baseline implementation. Further experiments are based on parameter modifications of the baseline approach.

Results for the baseline GMM-UBM approach

TABLE I. Results For The Baseline GMM-UBM Approach

| Approaches | Acc | EER |
|---|---|---|
| Baseline | 61.904 | 0.217 |

We performed a set of 10 experiments (Exp 1 – Exp 10) examining the MFCC feature vector shift parameter (Ts). The purpose is to determine if the number MFCC feature vectors can be reduced by spaced them farther apart across time. For larger Ts values, the MFCC feature vectors are spaced farther apart, and for smaller values, the vectors are spaced closer together. Hence, larger values would lead to reduced number of MFCC vectors, resulting in reduced computational complexity. Table 2 shows our results.

TABLE II. Results Based on changing the MFCC Feature Vector Shift Parameter

| Exp | Ts | Acc (%) | EER |
|---|---|---|---|
| 1 | 25 | 61.904 | 0.095 |
| 2 | 35 | 92.857 | 0.093 |
| 3 | 45 | 80.952 | 0.071 |
| 4 | 55 | 71.428 | 0.077 |
| 5 | 65 | 73.809 | 0.077 |
| 6 | 75 | 76.190 | 0.070 |
| 7 | 80 | 92.857 | 0.097 |
| 8 | 85 | 73.809 | 0.093 |
| 9 | 95 | 88.095 | 0.092 |
| 10 | 105 | 69.047 | 0.118 |

According to Table 2, the Exp 2 and Exp 7 have among the best combination of accuracies (92.857 and 92.857) with acceptable EERs (0.093 and 0.097). The experiments used Ts values of 35 ms and 80 ms, which is greater than the baseline Ts value of 10 ms. This suggests that MFCC feature vectors can be spaced farther apart across time while maintaining speaker recognition performing using the GMM-UBM approach. We note that the results in Table 2 have generally

lower EERs and Accuracies than the baseline system, which uses more MFCC frames. This seems to contradict what we know about speaker recognition, and we are currently investigating the root cause of this outcome.

Using the Ts value of 35 ms, we next examined the number of speakers used for the UBM training data (D) to see if data reductions can be made to improve UBM training speed. Accuracy and EER results are shown in Table 3. The numbers under the "D" column represent the speaker identities used to train the UBM. Each of the 42 speakers has a unique identity, labelled from 1 through 42.

TABLE III. RESULTS BASED ON CHANGING THE UBM TRAINING DATA

| Exp | # of spkrs used | D | Acc (%) | EER |
|---|---|---|---|---|
| 11 | 10 | 1 – 10 | 92.857 | 0.093 |
| 12 | 20 | 1 – 20 | 83.333 | 0.071 |
| 13 | 20 | 1 – 10<br>31 – 40 | 85.741 | 0.069 |
| 14 | 10 | 31 – 40 | 85.714 | 0.071 |
| 15 | 10 | 1,5,9,13,<br>17,21,25,<br>29,33,38 | 76.190 | 0.071 |

Results from Table 3 suggests that there is an inconclusive effect of altering the speaker identities used for UBM training. Using speaker identities 1-10 resulted in the best Accuracy (92.867%), while using speaker identities 1-10 and 31-40 resulted in the best EER (0.069). However, it is preferable to use fewer speakers for UBM training to reduce computational complexity (i.e. Exp 11, Exp 14, and Exp 15).

Afterwards, we examined the effect of the number of UBM mixtures (M) to speaker recognition performance, using a Ts of 35, and speakers 1-10 for UBM training. Results are shown in Table 4.

TABLE IV. RESULTS BASED ON CHANGING THE GMM MIXTURE AMOUNTS

| Exp | M | Acc (%) | EER |
|---|---|---|---|
| 16 | 4 | 80.952 | 0.095 |
| 17 | 8 | 92.857 | 0.093 |
| 18 | 16 | 85.714 | 0.07 |

These results suggest that using eight UBM mixtures (Exp 17) produces the best Accuracy (92.857%), and the second best EER (0.093). Eight UBM is also used in the baseline GMM-UBM approach. Increasing the number of mixtures to 16 does produce better EER, but requires more computational complexity. It should also be noted that the use of only 4 mixtures (Exp 16) produces an Accuracy of 80.952% and EER of 0.095, while requiring the least computational cost.

Lastly, we examined different parameter combinations in an attempt to arrive at the optimal tradeoff between computational complexity and performance. Table 5 shows the results that generated the best EERs, which is the most widely used performance measure in practice.

TABLE V. RESULTS BASED ON CHANGING MULTIPLE PARAMETERS

| Exp | Ts | D | Acc (%) | EER |
|---|---|---|---|---|
| 19 | 75 | 1,5,9,13,<br>17,21,25,<br>29,33,38 | 83.333 | 0.045 |
| 20 | 80 | 1,5,9,13,<br>17,21,25,<br>29,33,38 | 83.333 | 0.047 |

Results suggest that using large MFCC vector shifts of 75 and 80, along with a uniform distribution of 10 speakers for UBM training resulted in the optimal EERs (0.045 and 0.047), and decently Accuracy (83.333%)

## VI. SUMMARY AND FUTURE WORK

This work explored parameter reduction approaches to the widely established GMM-UBM speaker recognition approach on a set of 42 speakers, each with roughly 3 minutes of speech. The aim was to reduce the computational complexity of the approach by using fewer parameters. We found that by increasing the MFCC feature vector shifts, the computational cost can be significantly reduced while even improving speaker recognition performance. The number of UBM training speakers and GMM mixtures were also examined, but did not result in significant performance improvements while reducing computational complexity. This can be attributed to the lack of statistical significance in some results. Future work will incorporate more speakers to generate greater statistical significance in the results, and quantify the reduction in computation time for the different experiments. Afterwards, approaches with lowest computational complexity (i.e. time) will be used for real-time speaker recognition implementations on embedded processor hardware platforms.



## REFERENCES

[1] J.F. Bonastre, F. Bimbot, L.J. Boe, J.P. Campbell, D.A. Reynolds, and I. Magrin-Chagnolleau, "Person Authentication by Voice: A Need for Caution", in 8th European Conference on Speech Communication and Technology, Geneva, Switzerland, 2003.J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68-73.

[2] P. Rose, *Forensic Speaker Identification*. London: Taylor & Francis, 2002.

[3] J.P. Campbell, W. Shen, W.M. Campbell, R. Schwartz, J.F. Bonastre, and D. Matrouf, "Forensic Speaker Recognition", in *IEEE Signal Processing Magazine*, 2009, pp. 95-103.

[4] D.A. Reynolds, T.F. Quatieri, and R. Dunn, "Speaker Verification using Adapted Gaussian Mixture Models," in *Digital Signal Processing*, Vol. 10 No. 3, 2000, pp. 19–41.

[5] [5] P. Kenny, G. Boulianne, P. Ouellet, and P. Dumouchel, "Joint Factor Analysis Versus Eigenchannels in Speaker Recognition", in *IEEE Transactions on Audio, Speech, and Language Processing*, Vol. 15(4), 2007, pp. 1435-1447.

[6] N. Dehak, R. Dehak, P. Kenny, N. Brummer, P. Ouellet, and P. Dumouchel, "Support Vector Machines versus Fast Scoring in the Low-Dimensional Total Variability Space for Speaker Verification", in Proceedings of *Interspeech*, Brighton, UK, 2009, pp. 1559-1562.

[7] L. Burget, P. Oldrich, C. Sandro, G. Oldrej, M. Pavel, and N. Brummer, "Discriminantly Trained Probabilistic Linear Discriminant Analysis for Speaker Verification", in Proceedings of ICASSP, Brno, Czech Republic, 2011.

[8] A.P. Dempster, N.M. Laird, D.B. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm", *Journal of the Royal Statistical Society, Series B*, Vol. 39(1), 1977, pp. 1-38.

[9] J.B. MacQueen, "Some Methods for Classification and Analysis of Multivariate Observations", Proceedings of 5th Berkeley Symposium on

Mathematical Statistics and Probability 1. University of California Press. 1967, pp. 281-297.

[10] The NIST Year 2012 Speaker Recognition Evaluation Plan", http://www.nist.gov, 2012.

[11] S. Davis and P. Mermelstein, "Comparison of Parametric Representations of Monosyllabic Word Recognition in Continuously Spoken Sentences", in Proceedings of *ICASSP*, 1980.