Speaker verification using speaker-specific-text

B. BHARATHI
SSN college of Engineering
Department of Computer Science and Engg.,
Kalavakkam, Chennai 603110
India
bharathib@ssn.edu.in

T. NAGARAJAN
SSN College of Engineering
Department of Information Technology
Kalavakkam, Chennai 603110
India
nagarajant@ssn.edu.in

Abstract: In speaker recognition tasks, one of the reasons for reduced accuracy is due to closely resembling speakers in the acoustic space. In conventional GMM-based modeling technique, since the model parameters of a class are estimated without considering other classes in the system, features that are common across various classes may also be captured, along with unique features. If the system is designed to use only the unique features of a given speaker with respect to his/her acoustically resembling speaker, then the system is expected to perform better. In this proposed work, the effect of a subset of phonemes, reasonably distinct (unique) to a speaker, in the acoustic sense, on a speaker verification task is investigated. This paper proposes a technique to reduce the confusion errors, by finding speaker-specific phonemes and formulate a text using the subset of phonemes that are unique, for speaker verification task using GMM-based approach and i-vector based approach. We have experimented with three techniques namely, product of likelihood-Gaussians-based distance, Bhattacharyya distance and average log-likelihood-based distance to find out acoustically unique phonemes. Experiments have been conducted on speaker verification task using speech data of 50 speakers collected in a laboratory environment. The experiments show that the Equal Error Rate (EER) has been decreased by 4% and 4.5% using speaker-specific-text when compared to that of GMM and i-vector technique with random-text respectively.

Key–Words: Speaker verification, Product of Gaussian, Gaussian Mixture Model, i-vector, acoustic likelihood

1 Introduction

Gaussian Mixture Modeling (GMM) and Hidden Markov Modeling (HMM) techniques have been successfully used in many classification tasks. Maximum Likelihood Estimation (MLE) and Expectation Maximization (EM) algorithms can be used to estimate the model parameters efficiently. However, a major drawback in this type of modeling techniques is that the modeling is carried out in isolation, i.e., the modeling technique, when modeling a class, does not consider the information from other classes. In other words, out-of-class data is not used to optimize the classifier performance. This may lead to poor models with parameters that are common to other classes, in addition to the unique parameters of a class. This may, in turn, increase the classification (or confusion) error. Better classification accuracy can be achieved if the training technique is able to capture the unique features of a class, i.e., the features that discriminate a class from other classes, efficiently.

Many research works have been reported in the literature to increase the classification accuracy of a classifier by increasing the discriminative power of the classifier. Such techniques can be grouped into mainly two classes as follows:

1. Discriminating the classes in the feature level itself by identifying and removing the common features between two classes under consideration.

2. Adjusting the model parameters themselves such that two classes, in the feature space itself, are well separated.

In [20], the use of GMM for speaker identification was shown to provide good performance with several existing techniques. However, this criterion only utilizes the labeled utterances for each speaker model and very likely leads to a local optimization solution.

To improve the discriminative qualities of Gaussian mixture models, several approaches have been proposed. Universal Background Model-Gaussian Mixture Model (UBM-GMM) is a popular one among them. UBM is a base model from which all speaker models are adapted by a form of Bayesian adaptation [21]. A UBM is built from a large data set containing all probable speakers. During training, speaker specific model is adapted from this UBM by performing Maximum A Posteriori (MAP) adaptation. In [9], segmental Generalized Probabilistic Descent (GPD) algorithm has been used to estimate model parameters of a class considering the competing speakers. Maximum Mutual Information Estimation (MMIE)-based methods have been used to model a class considering the rest of the models [3] or a subset of remaining models [17, 6]. In [12], Maximum model distance algorithm for GMM is described. This approach [6] tries to maximize the distance between each model and a set of competitive speakers models. In [22], GMMs have been built for each speaker discriminatively based on the available positive and negative examples for each speaker. In this approach [22], speaker models are trained by moving the mean values of the mixture components in such a way as to maximize the likelihood of speaker data while also minimizing the likelihood of negative examples for the speaker.

Minimum Classification Error (MCE) approach for speaker verification is proposed in [16]. In this approach [16], all the competing speakers are used to evaluate the score of the anti speaker which is found to be effective. However, it is not practical for verification test over a large population. An interesting method has been proposed in [1] where the outliers are de-emphasized. Product of likelihood Gaussians (POG) has been used to estimate the bias of a model in [19]. Product of likelihood Gaussians has been used to identify the most probable confusing features between two classes in [2]. Then the common features are removed from the training data. By eliminating the confusing features, during testing, evidence is derived only from the features that are unique to a class.

To avoid playback of recorded voice of the genuine speaker, a text prompted speaker verification task using HMM and Multilayer Perceptron (MLP) is described in [10]. The set of context-independent phoneme HMMs is used to provide a segmentation of the speech signal into phonemes with a simple Viterbi forced alignment. The feature vectors, labeled with the corresponding phonemes, are then used to train MLPs, one per phoneme and per speaker. The discriminative power of the most frequently appearing phonemes was investigated. However, those phonemes are not unique to the particular speaker.

The i-vector systems have become the state-of-the-art technique in the speaker verification field [7]. They provide an elegant way of reducing large-dimensional input data to a small-dimensional feature vector while retaining most of the relevant information. The technique was originally inspired by Joint Factor Analysis framework introduced in [14]. JFA [14] is based on the decomposition of a speaker dependent GMM super vector into separate speaker-and channel-dependent part. This separation allows for learning the channel characteristics in the form of separate model, hence producing pure channel independent speaker models. This proves to be an efficient technique for handling channel and session variability.

In our proposed work, the classes are discriminated at the phoneme level, i.e., acoustically unique phonemes of a speaker when compared to his/her closely resembling speakers were derived. In this paper, test utterances are formulated using speaker-specific-text and random text. The speaker-specific-text is formulated using the acoustically unique phonemes. The random-text is formulated without considering the acoustically unique phonemes. The major focus of this paper is to show that the speaker verification task using speaker-specific-text greatly reduces the confusion errors when compared to speaker verification task using random-text.

In this paper, we investigated the usefulness of speaker-specific-text on speaker verification task using two approaches namely GMM and i-vector. The major advantage of i-vector approach in the speaker verification task is its additional ability to handle channel and session variabilities. However, our proposed work uses same channel for all the speakers during training as well as testing and the session variability is also not considered. In this work, i-vector approach is used, to check the effect of acoustically unique phonemes in a classification task.

The organization of this paper is given below. The next section describes the importance of speaker-specific-text and the techniques used to derive acoustically unique phonemes in our proposed system. Our proposed system using GMM-based approach are described in Section 3. The introduction about i-vector approach is presented in Section 4. Section 5 describes the details of our speech corpus, and experimental setup of the proposed system. Section 6 deals with the performance analysis of the proposed technique on speaker verification task. Finally, section 7 concludes the paper.

2 Relevance of speaker-specific-text

In [2], a GMM-based technique was proposed to equip a classifier to capture the unique features of a class and to make decisions based on the unique features alone. During testing, feature vectors that are unique to a class have been derived and used, thereby the classification accuracy is increased. One of the drawbacks is that, if the test utterance does not contain reasonable number of unique features, then the discrimination power cannot be ensured. Another drawback is that the unique features have to be identified from the test utterances during testing thus increases the computation time. If the speaker is able to utter the word which contains only the unique features then the computation time will be reduced. Even though the unique feature vectors are known, one cannot expect a speaker to utter speech segments, that contain these features alone. On the other hand, if we know unique phoneme list apriori, one can formulate a text to be uttered using such phonemes alone.

In this proposed work we investigate the effect of a subset of phonemes, that are unique to a speaker in the acoustic sense on a speaker verification task. The proposed technique involves three main steps:

1. To find out confusing speakers for each speaker
2. To derive acoustically unique phoneme set for each speaker when compared to his/her confusing speakers
3. To perform testing using speaker-specific-text

The proposed technique was experimented in [4] on speaker identification task using TIMIT speech corpus. The authors of [4] have demonstrated the improvement in classification accuracy, by considering only one confusing speaker for each of the speakers, during training as well as testing phases. In [4], one of the reasons identified for misclassification is that the acoustically unique phonemes set is derived using average of log-likelihood values. If some of the phonemes have less number of examples, then considering the statistical parameter like the mean of likelihoods, is not appropriate. In [4], since the TIMIT speech corpus is used, many phonemes have very less number of examples (even just two) it is not appropriate to use the mean value and this might have led to false set of phonemes as unique. This error can be avoided by creating our own phonetically balanced speech corpus[1]. Hence, in this proposed work, we have created our own speech corpus in which we have ensured that all the phonemes have reasonable number of (minimum 30) examples.

In [4], only one confusing speaker is considered for each of the speakers. During testing, if the confusing speaker is not present in the first position, we will not have the chance to improve the performance. On the other hand, if we consider more than one confusing speaker for each of the speakers, then common set of unique phonemes can be derived from all of the confusing speakers. One may assume that these phoneme set is, to certain extent, unique to the other speakers too. Let us consider a closed-set speaker recognition task, with N speakers. For any speaker, in a given set of N speakers, unique phonemes can be derived in the following two ways:

1. Considering the rest of the $N-1$ speakers as competing speakers.
2. Considering a smaller set of speakers (say m speakers, where $m \ll N$) as competing speakers.

In case(1), when N is very large, deriving unique phonemes is computationally expensive. It is reasonable to assume that most of the speakers in the total set N will not be acoustically closer to the test speaker. One more reason is that, when the number of confusing speakers is increased, the number of common unique phonemes is decreased, hence it may not be possible to formulate the speaker-specific-text. Considering these reasons, in our work, only a subset of speakers is considered. For this purpose, any minimum distance classifier or maximum likelihood classifier can be used. For the current study, conventional GMM testing is used to derive this subset by considering m-best results of GMM technique.

For each pair of speakers, where the pair consists of the intended speaker and one of the competing speakers, the unique phonemes can be derived as follows: Given the speech segments for each of the phonemes and the model (GMM) for the speakers in the pair, the unique phonemes (or acoustically unique phonemes) can be derived. In our proposed work, m(for this work, m=3) competing speakers are considered. Therefore, a common set of unique phoneme is derived from all of the competing speakers.

In this paper, speaker verification task using speaker-specific-text is implemented using GMM and i-vector approach. For both GMM and i-vector approach, the method to find out confusing speakers for each speaker and the method to derive acoustically unique phoneme for each speaker when compared to his/her confusing speaker are same which is explained in Section 2.1 and 2.2 respectively.

2.1 Identifying confusing speakers

Given the training utterances and GMM of each speaker, the confusing speaker list for each speaker(S_i) is derived using the algorithm described below:

[1] NIST SRE corpora cannot be used for the proposed approach due to the reason that our approach requires speech data to be collected for speaker-specific-text and used during testing

Input: Let us consider the training utterances of speaker S_i and λ_1,λ_2, ... λ_n be the GMMs of the speakers S_1,S_2, ... S_n.
Output: Confusing speaker list of speaker S_i.

1. The training utterances of speaker S_i are tested against the model λ_j, where $j = 1, 2, ..n$ & $j \neq i$.
2. Estimate the likelihoods of the training utterances of speaker S_i being produced by j^{th} speaker model λ_j, $j = 1, 2, ..n$ & $j \neq i$ and compute the average likelihood value.
3. Sort the average likelihood values of speaker S_i in descending order.
4. The speaker's correspond to first m(for this work, $m = 3$) are considered as confusing speakers for the speaker S_i.

The above algorithm is used to derive confusing speaker list for all the speakers.

2.2 Creation of acoustically unique phoneme set

The following three distance metrics have been experimented to derive acoustically unique phoneme set for each speaker when compared to his / her confusing speakers. The methods are:

1. Product of likelihood-Gaussians-based distance
2. Bhattacharyya-based distance
3. Average Log-likelihood

2.2.1 Product of likelihood-Gaussians

The product of likelihood-Gaussians technique presented in [2] tries to identify the most probable common features in likelihood space. In our proposed work, acoustically unique phonemes are derived by using product of likelihood-Gaussians. The procedure to estimate product of likelihood-Gaussians is explained as follows:

Let us consider the feature vectors of two different classes (C_i and C_j) as x_k^i and x_k^j. Let λ_i and λ_j be the models of the classes, C_i and C_j, respectively. Let the likelihoods of the feature vectors of the class C_i for the given models λ_i and λ_j be $p(x_k^i|\lambda_i)$ and $p(x_k^i|\lambda_j)$ respectively. We can assume that these likelihoods are distributed normally in likelihood space with suitable parameters. Let these two Gaussians be $N_{ii}(\mu_{ii}, \sigma_{ii}^2)$ and $N_{ji}(\mu_{ji}, \sigma_{ji}^2)$. Similarly, for the feature vectors of the class C_j, the likelihood-Gaussians are $N_{jj}(\mu_{jj}, \sigma_{jj}^2)$ and $N_{ij}(\mu_{ij}, \sigma_{ij}^2)$. The overlap in the feature space is reflected in the likelihood space. The overlapped region between N_{ji} and N_{ii} indicate that a subset of x_k^i gives likelihood in the same range for the models λ_i and λ_j. As the overlap increases, the number of feature vectors of x_k^i that give likelihood in the same range for both the models increases. This increases the probability of an unseen common feature vector, belonging to class C_i, giving a better likelihood for the model λ_j. Therefore, the overlap can be used as a measure of the number of features that class C_i shares with class C_j.

A method to quantify the amount of overlap between two Gaussians was proposed in [18] and was used in [19] to calculate the amount of bias. The same method is used here to calculate the commonality between two classes and the μ_k(PoG mean) to identify the most probable confusing features. The details of the method presented in [18] is given below for clarity purposes and it is used to identify the common feature vectors.

Let $N_{ii}(\mu_{ii}, \sigma_{ii}^2)$ and $N_{ji}(\mu_{ji}, \sigma_{ji}^2)$ be the Gaussian distribution of the utterances of the class C_i for the given models λ_i and λ_j
Let $N_k(\mu_k, \sigma_k^2)$ be

$$N_k(\mu_k, \sigma_k^2) = N_{ii}(\mu_{ii}, \sigma_{ii}^2) \ . \ N_{ji}(\mu_{ji}, \sigma_{ji}^2). \quad (1)$$

For the product of the Gaussians, the mean (μ_k) and its variance (σ_k^2) can be given as

$$\mu_k = \frac{\sigma_{ji}^2 \mu_{ii} + \sigma_{ii}^2 \mu_{ji}}{\sigma_{ii}^2 + \sigma_{ji}^2}, \quad (2)$$

$$\sigma_k^2 = \frac{\sigma_{ii}^2 \sigma_{ji}^2}{\sigma_{ii}^2 + \sigma_{ji}^2}. \quad (3)$$

In order to quantify the amount of overlap between two different Gaussians, the following ratio ($\mathcal{O}_{ij}$) is defined:

$$\mathcal{O}_{ij} = \frac{\max[N_{ii}(\mu_{ii}, \sigma_{ii}^2) \ . \ N_{ji}(\mu_{ji}, \sigma_{ji}^2)]}{\max[N_{ii}(\mu_{ii}, \sigma_{ii}^2) \ . \ N_{ii}(\mu_{ii}, \sigma_{ii}^2)]} \quad (4)$$

$$\mathcal{O}_{ij} = \frac{\sigma_{ii}}{\sigma_{ji}} \ e^{-\left[\frac{(\mu_k - \mu_{ii})^2}{2\sigma_{ii}^2} + \frac{(\mu_k - \mu_{ji})^2}{2\sigma_{ji}^2}\right]}. \quad (5)$$

If $\mu_{ii} = \mu_{ji}$, then Equation (4) reduces to

$$\mathcal{O}_{ij} = \frac{\sigma_{ii}}{\sigma_{ji}}. \quad (6)$$

However, for this case we expect the overlap $\mathcal{O}_{ij}$ to be equal to 1. To achieve this, Equation (4) is further normalized as given below:

$$\begin{aligned}\mathcal{O}^{\mathcal{N}}_{ij} &= \mathcal{O}_{ij}\frac{\sigma_{ji}}{\sigma_{ii}} \\ &= e^{-\left[\frac{(\mu_k-\mu_{ii})^2}{2\sigma_{ii}^2}+\frac{(\mu_k-\mu_{ji})^2}{2\sigma_{ji}^2}\right]}. \quad (7)\end{aligned}$$

The resultant $\mathcal{O}^{\mathcal{N}}_{ij}$ is used as a measure to estimate the amount of overlap between two gaussians (For further details, [2] can be referred).

To derive speaker-specific-text of a speaker, the common phonemes (i.e., the corresponding speech segments) of the speaker and his/her confusing speaker, available in the training utterances, are tested with his/her model and his/her confusing speaker's model. Log-likelihood of each phoneme is computed for the intended speaker and the confusing speaker. Using the Equation (7), the amount of overlap between the two Gaussians for each phoneme is estimated. Based on the sorted values of $\mathcal{O}^{\mathcal{N}}_{ij}$, the first 30 phonemes are considered as acoustically unique phonemes (Thirty phonemes are chosen so that reasonable number of acoustically unique phonemes can be made available while deriving common acoustically unique phonemes across multiple confusing speakers). For each speaker with respect to his/her confusing speakers, different subset of acoustically unique phonemes are derived.

2.2.2 Bhattacharyya-based distance

The Bhattacharyya-based distance [23] to measure the distance between two GMM distribution is used to find out the acoustically unique phonemes. Let $N_{ii}(\mu_{ii},\sigma_{ii}^2)$ and $N_{ji}(\mu_{ji},\sigma_{ji}^2)$ be the Gaussian distribution of the utterances of the class C_i for the given models λ_i and λ_j. The Bhattacharyya-based GMM-distance measure between the two Gaussian distributions as shown in Equation (8).

$$\begin{aligned}k &= \frac{1}{8}(\mu_{ji}-\mu_{ii})^t\left[\frac{\sigma_{ii}^2+\sigma_{ji}^2}{2}\right]^{-1}(\mu_{ji}-\mu_{ii})\frac{1}{2} \\ &\quad + \left[ln\frac{\left|\frac{\sigma_{ii}^2+\sigma_{ji}^2}{2}\right|}{\sqrt{\left|\sigma_{ii}^2\right|\left|\sigma_{ji}^2\right|}}\right] \quad (8)\end{aligned}$$

To derive speaker-specific-text of a speaker, the common phonemes (i.e., corresponding speech segments) of the speaker and his/her confusing speaker, available in the training utterances, are tested with his/her model and his/her confusing speaker's model. Log-likelihood of each phoneme is computed for the intended speaker and the confusing speaker. Here also the likelihood values are assumed to posses normal distribution. Using the Equation (8), the bhattacharya distance between the two Gaussians for each phoneme is estimated. The estimated distance values are sorted in descending order. If the bhattacharya distance value is high then the corresponding phoneme is considered to be unique. Therefore, the first thirty phonemes are considered as acoustically unique phonemes. For each speaker with respect to his/her closely resembling speakers different subset of acoustically unique phonemes are derived.

2.2.3 Average log-likelihood

Given the training utterances of the speaker S_i, GMMs of the speaker(S_i) and his/her confusing speakers, the acoustically unique phonemes are derived using the algorithm described below:

Input:Let us consider the training utterances of speaker S_i. Let us denote GMM of speaker S_i as λ_{S_i} and its confusing speakers $C^j_{S_i}$as $\lambda_{C^j_{S_i}}$ (where, $j = 1, 2, ..m$) and U be the phoneme set.
Output:Acoustically unique phonemes of speaker S_i.

1. For each phoneme in the set U, the corresponding speech segments of speaker S_i, available in the training utterances are tested against the speaker model λ_{S_i} and his / her confusing speaker model $\lambda_{C^j_{S_i}}$.
2. Estimate the average likelihood of the phoneme of speaker S_i being produced by the speaker model λ_{S_i} as $\lambda^{avg}_{S_i}$.
3. Estimate the average likelihood of the phoneme of speaker S_i being produced by the confusing speaker model $\lambda_{C^j_{S_i}}$ as $\lambda^{avg}_{C^j_{S_i}}$.
4. Calculate the difference in average likelihood d_j= $\lambda^{avg}_{S_i} - \lambda^{avg}_{C^j_{S_i}}$.
5. Repeat the steps (1) to (4) for all the phonemes in the set U.
6. Based on the sorted values of d_j, the first 30 phonemes are considered as acoustically unique phonemes.

The above algorithm is used to derive acoustically unique phonemes for all the speakers.

We used consistency as the metric to find out the best method among the methods listed above, to derive acoustically unique phoneme set for each speaker. In our proposed work, our speech corpus consists of 130 training utterances and 12 test utterances. The number of speakers taken for this experiment is 10. In order to find out the consistency, the training utterances have been divided into two sets. Each set consists of 65 training utterances. Using the above three methods, acoustically unique phoneme set for each speaker is derived with the two sets of training utterances separately. If the acoustically unique phoneme set derived using the two sets of training utterances are almost same, then the above algorithm is said to have the consistency. In order to find out the consistency, number of acoustically unique phonemes that are common between the two sets of training utterances are computed for 10 speakers. Using the number of common and acoustically unique phonemes between the two sets of training utterances, the mean value and standard deviation are estimated and tabulated in table 1

Table 1: Mean and standard deviation of the number of common and acoustically unique phonemes between the two sets of training utterances using the 3 methods

Method	mean (μ)	Standard deviation(σ)
Average log-likelihood	16.2	4.13
Product of likelihood-Gaussians	19.6	2.75
Bhattacharyya-based distance	11.4	7.8

For better consistency, the mean value should be high and standard deviation should be low. It implies that the number of acoustically unique phonemes that are common between the two sets of training utterances are more. From table 1, it can be noted that, product of likelihood-Gaussians-based distance method, gives higher mean value and lower standard deviation value. From this result, It has been decided to use product of likelihood-Gaussians-based distance method for deriving acoustically unique phonemes for the given speaker. The speaker-specific-text is formulated using this acoustically unique phonemes and, performance of the proposed work is tabulated in Section 6.

The reason for choosing the number of confusing speaker as 3, is explained as given below Fig.(1)

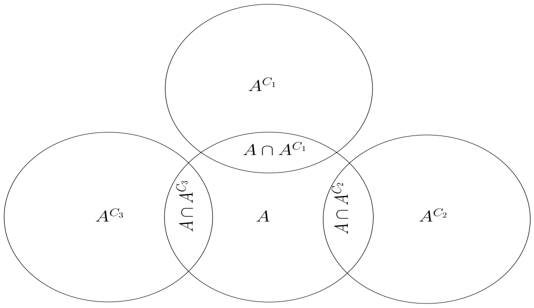

Figure 1: Representation of unique phoneme space and common phoneme space of a speaker A by considering more than one confusing speakers (A^{C_1}, A^{C_2}, A^{C_3})

In Fig.1, let the phoneme space of speaker A is represented by a circle A (middle circle). The phoneme space of the confusing speakers of speaker A are represented by A^{C_1}, A^{C_2} and A^{C_3}. The common unique phoneme space of speaker A (U^p) by considering their confusing speakers is represented by,

$$U^p = A - \sum_{i=1}^{m} A \cap A^{C_i} \tag{9}$$

where,
m - number of confusing speakers
$A \cap A^{C_i}$ - common phonemes between the speaker$A$ and his / her confusing speaker A^{C_i}, i = 1,2,. . . ,m.

From Equation (9) it can be noted that when the second term of RHS increases, the value of U^P decreases. From fig. 1, we can conclude that when the number of confusing speakers is increased, the number of common unique phonemes is decreased, hence it may not be possible to formulate the required number of speaker-specific-text for some speakers. Because of this reason the number of confusing speaker for each speaker is fixed to 3 in this proposed work.

3 GMM-based approach using speaker-specific-text

Gaussian Mixture Models (GMM) [20] are popular statistical models due to their ability to form good approximations of data and the ease in computation. It is a linear combination of multiple Gaussian distributions.

A Gaussian mixture density is a weighted sum of M component densities given by the equation

$$p(\vec{x} \mid \lambda) = \sum_{i=1}^{M} p_i b_i(\vec{x}), \qquad (10)$$

where
$\vec{x}$ - is a D-dimensional feature vector
$b_i(\vec{x})$ - i th mixture component density, i=1,2,..,M
p_i - i th mixture weight, i=1,2,..,M

Each component density is a D-variate Gaussian function of the form

$$b_i(\vec{x}) = \frac{1}{(2\pi)^{\frac{D}{2}} |\Sigma_i|^{\frac{1}{2}}} \exp\{-\frac{1}{2}(\vec{x} - \vec{\mu_i})^t \Sigma_i^{-1} (\vec{x} - \vec{\mu_i})\} \qquad (11)$$

with mean vector $\vec{\mu_i}$ and covariance matrix Σ_i .
The mixture weight must satisfy the constraint that

$$\sum_{i=1}^{M} p_i = 1 \qquad (12)$$

The complete Gaussian mixture density is parametrized by the mean vectors, covariance matrix, and mixture weights of all the component densities. These parameters are collectively represented by

$$\lambda = [p_i, \vec{\mu_i}, \vec{\Sigma_i}], \quad i = 1, .., M$$

Given a collection of training vectors, maximum likelihood is estimated by using the iterative EM algorithm. The EM algorithm iteratively refines the GMM parameters to monotonically increase the likelihood of the estimated model for the observed feature vectors. Generally, five to ten iterations are sufficient for parameter convergence. The advantage of using the GMM as the likelihood function is that, it is computationally inexpensive and is based on a well-understood statistical model. For text-independent tasks, it is insensitive to the temporal aspects of the speech and only the underlying distribution of acoustic observations from a speaker has been modeled. The latter is also a disadvantage, because higher-levels of information about the speaker, conveyed in the temporal speech signal have not used by this approach. In this proposed work GMM has been used for speaker verification task.

Given the test utterance, GMMs of claimed speaker and impostor, testing using speaker-specific-text is carried out using the algorithm described below:

Input:Let us consider the test utterance of claimed speaker S_i. Let GMM of speaker S_i as λ_{S_i} and GMM of its impostor as $\lambda_{impost_{S_i}}$.
Output:Accept or reject the speaker S_i.

1. For speaker S_i, his/her m confusing speakers are considered as impostors.
2. Using the training utterances of confusing speakers $C_{S_i}^{j}$ (where, $j = 1, 2, ..m$), impostor model is created for speaker S_i.
3. The test utterance (which is created using speaker-specific-text using Product of likelihood-Gaussians-based distance as explained in Section 2.2.1)of the claimed speaker S_i is tested against the claimed speaker model(λ_{S_i}) and their corresponding impostor model($\lambda_{impost_{S_i}}$).
4. Calculate the score(S), by finding out the difference between the likelihood value of claimed speaker model(λ_{S_i}) and their impostor model($\lambda_{impost_{S_i}}$).
5. If score(S) $\geq$ empirically fixed threshold(θ) then accept the speaker S_i otherwise reject the speaker S_i.

The above algorithm is used to perform testing with GMM using speaker-specific-text for all the speakers.

4 i-vector approach

In i-vector based system, a variable length speech pattern is projected onto a low-dimensional linear subspace. The basis vectors of this subspace are estimated from the EM algorithm. This low dimensional representation of a speech utterance is termed as the i-vector (identity vector). The main idea in traditional JFA, is to find two subspaces which represent the speaker and channel-variabilities, respectively. The experiments in [8] show that JFA is partially successful in separating speaker and channel variabilities. The authors of [8] found that the channel space contains some information that can be used to distinguish between speakers. For this reason, the authors of [8] propose a single space that models the two variabilities and named it as the total variability space. The basic assumption is that, a given speaker- and channel-dependent GMM super vector M can modeled as follows:

$$M = m + Tw \tag{13}$$

where
m - is a Universal Background Model(UBM) super vector
T - is a low rank matrix, which represents a basis of the reduced total variability space
w - is the i-vector

T is named the total variability matrix. The components of w are the total factors and they represent the coordinates of the speaker in the reduced total variability space. These feature vectors are referred to as identity vectors or i-vectors. The feature vector associated with a given recording is the Maximum-a-Posteriori(MAP) estimate of w, whose calculation is explained in [11]. The matrix T is estimated using the EM algorithm described in [11]. However, our proposed work uses same channel for all the speakers during training as well as testing and the session variability is also not considered. In this work, i-vector and GMM approach is used, to check the effect of acoustically unique phonemes in a classification task.

The steps involved in testing with i-vector using speaker-specific-text as follows:

1. Gender-independent UBM is built using the training feature vectors of all the speakers.
2. For each speaker, the training utterances are concatenated and the total variability matrix is estimated.
3. The dimension of i-vectors is 400 (determined empirically).
4. i-vectors are extracted from the training utterances.
5. Similarly, i-vectors are extracted from the test utterances(which is created using speaker-specific-text using Product of likelihood-Gaussians-based distance as explained in Section 2.2.1)
6. The score is computed using Cosine Similarity Score(CSS). The CSS is computed by comparing the cosine angle between a test i-vector w_{test} and a target i-vector w_{target}:

$$score(w_{target}, w_{test}) = \frac{\langle w_{target}, w_{test} \rangle}{\|w_{target}\| \, \|w_{test}\|} \tag{14}$$

5 Experimental setup

In our proposed work, due to the requirement that all the phonemes should have enough examples, we have created our own speech corpus. We have collected 142 English sentences (from TIMIT corpus), that have enough number (minimum 30) of examples for all the 45 phonemes. The number of phonemes taken for this work is 45. The speech data is recorded using 16kHz sampling rate. Speech utterances are collected from 50 speakers which includes 43 female speakers and 7 male speakers. All the speakers uttered the same 142 sentences. The speakers age group is between 20 and 35. Each utterance is approximately of 3 second duration. The entire speech data is automatically segmented at phoneme-level using Forced Viterbi alignment algorithm[5] .

For each speaker, among 142 sentences, 130 are used for training and 12 are used for testing. For each speaker, a GMM with 128 mixture components has been trained, considering Mel-frequency cepstral coefficients (13 static + 13 dynamic + 13 acceleration) as the features. For each speaker, the confusing speaker list is derived by using the method described in Section 2.1. The methods to derive acoustically unique phonemes for each speaker is as explained in Section 2.2. In our proposed work, we have used product of likelihood-Gaussians-based distance method to derive acoustically unique phonemes. For each speaker, common and acoustically unique phonemes have been derived by considering 3 confusing speakers. To derive speaker characteristics, the constraint that is set in our work is that the test utterances should have at least six phonemes. For each speaker, the speaker-specific-text is formulated by combining sequence of consonant phoneme followed by vowel phoneme(CV words) using 6 common acoustically unique phonemes. The formulated speaker-specific-text need not be a meaningful word however it will be a readable text.

The method to find out confusing speakers for each speaker and deriving acoustically unique phoneme for each speaker when compared to his/her confusing speaker is common for both GMM-based approach and i-vector approach. In GMM-based approach, For each speaker, a GMM with 128 mixture components is trained for claimed speaker model. Similarly, using the training utterances of confusing speakers, impostor model is created for each speaker. For i-vector approach, gender-independent 128 mixture component UBM is built using the training utterances of all the 50 speakers. For each speaker, 130 training utterances are concatenated and the total variability matrix is estimated using the concatenated training utterances. When the system is tested using speech utterances that correspond to speaker-specific-text, the confusion error is found to be reduced considerably than that of the GMM and i-vector approach with random-text, as discussed below:

6 Performance Analysis

Speaker verification performance is compared between the utterances using speaker-specific-text (the utterances with acoustically unique phonemes) and the utterances using random-text (the utterances without considering the acoustically unique phonemes). The random-text is formulated by dividing the test utterance into words that contain 6 continuous phonemes. The words(random-text) are formed in such a way that, it should not contain more than two acoustically unique phonemes of the corresponding claimed speaker. The number of speakers taken for this experiment is 50. The performance of speaker verification task with random-text and speaker-specific-text using GMM and i-vector approach are tabulated in table 2.

Table 2: Speaker verification performance with Random-text and speaker-specific-text using GMM and i-vector approach

Approach	EER in %	
	Random-text(Conventional method)	Speaker-specific-text
GMM	7	3
i-vector	37	32.5

From Table 2, it can be noted that there is a 4% reduction in Equal Error Rate using GMM with speaker-specific-text when compared to GMM with random-text. Similarly, there is a 4.5% reduction in Equal Error Rate using i-vector with speaker-specific-text when compared to i-vector with random-text. From Table 2, we can notice that the EER is considerably higher in the case of i-vector approach. The proposed approach using GMM with random-text, the EER is only 7% as specified in row 1 of Table 2. This shows that i-vector approach may not be ideal when the duration of the test utterance is short [13, 15]. In any sense, in both GMM-based or i-vector based technique, if the test utterance is from speaker-specific-text, the performance is found to be better.

The speaker verification performance is compared between GMM with random-text and GMM with speaker-specific-text for test utterances with different duration. The number of speakers taken for this experiment is 50. For each speaker, the number of test utterances is 12. For GMM with random-text, the test utterances are formulated using 6, 12 and 18 randomly chosen phonemes respectively. For GMM with speaker-specific-text, the test utterances are formulated using 6, 12 and 18 acoustically unique phonemes respectively.

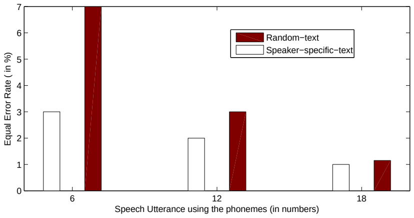

Figure 2: The speaker verification performance is compared between GMM with random-text and GMM with speaker-specific-text for test utterances with different duration

From Fig. 2 it can be noted that Equal Error Rate using GMM with speaker-specific-text is lower than that of GMM with random-text approach for different speech utterance duration also. We can also notice that, when the duration increases, performance difference between random-text and speaker-specific-text is decreasing. This is due to the fact that, longer duration test utterances using random-text may contain more number of acoustically unique phonemes.

The speaker verification performance is compared between i-vector with random-text and i-vector with speaker-specific-text for test utterances with different duration. The number of speakers taken for this experiment is 50. For each speaker, the number of test utterances is 12. For i-vector with random-text, the test utterances are formulated using 6, 12 and 18 randomly chosen phonemes respectively. For i-vector with speaker-specific-text, the test utterances are formulated using 6, 12 and 18 acoustically unique phonemes respectively.

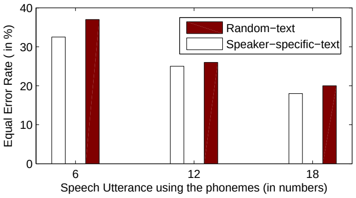

Figure 3: The speaker verification performance is compared between i-vector with random-text and i-vector with speaker-specific-text for test utterances with different duration

From Fig. 3 it can be noted that Equal Error Rate using i-vector with speaker-specific-text is lower than that of i-vector with random-text approach for different speech utterance duration also.

Speaker verification performance of the system is analysed using i-vector with random-text approach for test utterances with different duration have been tabulated in Table 3.

Table 3: Speaker verification performance using i-vector with random-text approach for test utterances with different duration

S.No	Speech utterance duration (in seconds)	EER in %
1	0.5	37
2	3	11
3	36	6

From Table 3, it can be noted that, if the utterance length decreases, speaker verification performance degrades at an increasing rate using i-vector based approach as described in [13, 15].

The limitation of our proposed system is that, every time a new speaker is introduced, confusing speakers list has to be generated. Then acoustically unique phoneme set may be changed for few speakers (only the speaker, who have the newly introduced speaker as confusing speaker in the confusing speaker list).

7 Conclusions

In this paper, we have proposed to use unique phonemes of a speaker, in other words, a set of phonemes that are acoustically unique when compared with that of a competing (acoustically closely resembling) speakers to reduce the confusion error on speaker verification task. A novel method has been explored to derive speaker-specific-text using Product of likelihood-Gaussians. The experiments show that the Equal Error Rate has been decreased by 4% and 4.5% using speaker-specific-text when compared to that of GMM and i-vector approach with random-text respectively. The experimental results show that the i-vector approach may not be ideal when the duration of the test utterance is short. In any sense, in both GMM-based or i-vector based technique, if the test utterance is from speaker-specific-text, the performance is found to be better.

Acknowledgment

The authors would like to sincerely thank Dr. Hema A Murthy, Professor and M.R. Srikanth, PhD Research scholar of Indian Institute of Technology (IIT) Madras for providing the codes for i-vector approach.

References:

[1] Levent M. Arslan and John H. L. Hansen. Selective training for hidden Markov models with applications to speech classification. *IEEE Transactions on Speech and Audio Processing*, 7(1):46–54, 1999.

[2] C. Arun Kumar, B. Bharathi, and T. Nagarajan. A discriminative GMM technique using product of likelihood Gaussians. In *Proc.TENCON 2009 - IEEE Region 10 Conference*, pages 1–6, 2009.

[3] L. R. Bahl, P. F. Brown, P. V. de Souza, and R. L. Mercer. Maximum mutual information estimation of hidden Markov model parameters for speech recognition. In *Proc. ICASSP*, pages 49–52, 1986.

[4] B. Bharathi, P. Vijayalakshmi, and T. Nagarajan. Speaker identification using utterances correspond to speaker-specific-text. In *Proc.*

IEEE Students' Technology Symposium (TechSym), 2011 , pages 171–174, 2011.

[5] Fabio Brugnara, Daniele Falavigna, and Maurizio Omologo. Automatic segmentation and labeling of speech based on hidden Markov models. *Speech Communication*, 12(4):357–370, 1993.

[6] J.-K. Chen and F.K. Soong. An N-best candidates-based discriminative training for speech recognition applications. *IEEE Transactions on Speech and Audio Processing*, 2(1):206–216, 1994.

[7] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet. Front-end factor analysis for speaker verification. *IEEE Transactions on Audio, Speech, and Language Processing,*, 19(4):788–798, 2011.

[8] Najim Dehak, Rda Dehak, Patrick Kenny, Niko Brmmer, Pierre Ouellet, and Pierre Dumouchel. Support vector machines versus fast scoring in the low-dimensional total variability space for speaker verification. In *Proc. INTERSPEECH*, pages 1559–1562, 2009.

[9] C.M. Del Alamo, F.J. Caminero Gil, C. dela Torre Munilla, and L. Hernandez Gomez. Discriminative training of GMM for speaker identification. In *Proc. ICASSP*, volume 1, pages 89–92 vol. 1, 1996.

[10] D.P. Delacretaz and J. Hennebert. Text-prompted speaker verification experiments with phoneme specific MLPs. In *Proc. ICASSP*, volume 2, pages 777–780 vol.2, 1998.

[11] O. Glembek, L. Burget, P. Matejka, M. Karafiat, and P. Kenny. Simplification and optimization of i-vector extraction. In *Proc. ICASSP*, pages 4516–4519, 2011.

[12] Q. Y. Hong and S. Kwong. Discriminative training for speaker identification based on maximum model distance algorithm. In *Proc. ICASSP*, volume 1, pages I–25–8 vol.1, 2004.

[13] Ahilan Kanagasundaram, Robbie Vogt, David Dean, Sridha Sridharan, and Michael Mason. i-vector based speaker recognition on short utterances. In *Proc. INTERSPEECH*, pages 2341–2344, 2011.

[14] P. Kenny, G. Boulianne, P. Ouellet, and P. Dumouchel. Joint factor analysis versus eigenchannels in speaker recognition. *IEEE Transactions on Audio, Speech, and Language Processing,*, 15(4):1435–1447, 2007.

[15] Anthony Larcher, Pierre-Michel Bousquet, Kong-Aik Lee, Driss Matrouf, Haizhou Li, and Jean-Franois Bonastre. I-vectors in the context of phonetically-constrained short utterances for speaker verification. In *Proc. ICASSP*, pages 4773–4776, 2012.

[16] Chi-Shi Liu, Chin-Hui Lee, Biing-Hwang Juang, and A.E. Rosenberg. Speaker recognition based on minimum error discriminative training. In *Proc. ICASSP*, volume i, pages I/325–I/328 vol.1, 1994.

[17] K. Markov, S. Nakagawa, and S. Nakamura. Discriminative training of HMM using maximum normalized likelihood algorithm. In *Proc. ICASSP*, volume 1, pages 497–500 vol.1, 2001.

[18] T. Nagarajan and D. O'Shaughnessy. Discriminative MLE training using a product of gaussian likelihoods. In *Proc. INTERSPEECH*, pages 601–604. ISCA, 2006.

[19] T. Nagarajan and D. O'Shaughnessy. Bias estimation and correction in a classifier using product of likelihood-Gaussians. In *Proc. ICASSP*, volume 3, pages 1061–1064, 2007.

[20] D.A. Reynolds and R.C. Rose. Robust text-independent speaker identification using Gaussian mixture speaker models. *IEEE Transactions on Speech and Audio Processing,*, 3(1):72–83, 1995.

[21] Douglas A. Reynolds, Thomas F. Quatieri, and Robert B. Dunn. Speaker verification using adapted Gaussian mixture models. *Digital Signal Processing*, 10(1-3):19–41, 2000.

[22] M.R. Srikanth and H.A. Murthy. Discriminative training of Gaussian mixture speaker models: A new approach. In *Proc. National Conference on Communications (NCC), 2010* , pages 1–5, 2010.

[23] Chang Huai You, Kong-Aik Lee, and Haizhou Li. GMM-SVM kernel with a Bhattacharyya-based distance for speaker recognition. *IEEE Transactions on Audio, Speech, and Language Processing,*, 18(6):1300–1312, 2010.