Multimed Tools Appl (2018) 77:8273–8294
DOI 10.1007/s11042-017-4723-9

Client-wise cohort set selection by combining speaker- and phoneme-specific I-vectors for speaker verification

Waquar Ahmad[1] · Harish Karnick[2] · Rajesh M. Hegde[3]

Received: 16 August 2016 / Revised: 6 April 2017 / Accepted: 17 April 2017 /
Published online: 24 May 2017

Abstract This work explores the use of phoneme level information in cohort selection to improve the performance of a speaker verification system. In speaker verification, *cohort* is used in score normalization to get a better performance. Score normalization is a technique to reduce the undesirable variation arising from acoustically mismatched conditions. Proper selection of cohort significantly improves speaker verification performance. In this paper, we investigate cohort selection based on a speaker model cluster under the i-vector framework that we call the i-vector model cluster (IMC). Two approaches for cohort selection are proposed. First approach utilizes speaker specific properties and called speaker specific cohort selection (SSCS). In this approach, speaker level information is used for cohort selection. The second approach is phoneme specific cohort selection (PSCS). This method improves cohort set selection by using phoneme level information. Phoneme level information is further employed in a late fusion approach that uses a majority voting method on normalized scores to improve the performance of the speaker verification system. Speaker verification experiments were conducted using the TIMIT, HINDI and YOHO databases. An equal error rate improvement of 19.01%, 14.61% and 19.4%is obtained for the proposed method compared to the standard ZT-Norm method for TIMIT, HINDI and YOHO datasets.

✉ Waquar Ahmad
waquar@nitsikkim.ac.in

Harish Karnick
hk@cse.iitk.ac.in

Rajesh M. Hegde
rhegde@iitk.ac.in

[1] Department of ECE, NIT Sikkim, Ravangla, Sikkim, 737139, India

[2] Department of Computer Science and Engineering, IIT Kanpur, Kanpur, Uttar Pradesh, 208016, India

[3] Department of Electrical Engineering, IIT Kanpur, Kanpur, Uttar Pradesh, 208016, India

Reasonable improvements in performance are also obtained in terms of minimum decision cost function (min DCF) and detection error trade-off (DET) curves.

Keywords Speaker verification · Speaker recognition · Cohort selection

1 Introduction

Speaker verification is a method for verifying a claimed identity (target) using the speaker's voice [4, 5, 31, 34]. It has wide ranging applications in access control, forensics, and security. Speech is preferred over other biometrics in remote authentication cases because of speech is easier to capture and send over a voice channel. The classifications of speaker verification is broadly divided into two categories such as text-dependent and text-independent. In the case of text-dependent speaker verification, text of uttered speech is available as well as an extra information can be extracted from the available text which leads to better speaker verification system performance [33, 38], while there is no restriction on the uttered speech in text independent speaker verification [8, 20]. The Performance of a speaker verification system is adversely affected by training and testing mismatch conditions [32, 36]. This mismatch can be reduced at three level, i.e signal, model and score level.

The technique like score normalization is performed at the score level to reduce the effects of mismatch between training and testing conditions [3]. The cohort normalization method is one of the score normalization methods which significantly reducing the mismatch condition [35]. *Cohort* is defined as a set of people with a shared characteristic. In Speaker Verification literature, it refers to a speaker-dependent set of anti-speakers (impostors) [22]. The proper selection of a cohort is critical for implementing in a speaker verification system [22, 28]. In biometric authentication, there are two kinds of errors, i.e. false acceptance (FA) and false rejection (FR). FA means accepting an impostor and FR refers to rejecting a genuine speaker. In such circumstances the verification threshold is generally adjusted to obtain a balance between these two types of errors. So, the number of false rejections can be reduced by lowering the verification threshold, but at the cost of an increase in the number of false acceptances. The number of false acceptances can be reduced by setting a high threshold, but this causes to increases the false rejections. Therefore, this problem is challenging when highly trained impostors gain access to an online speaker authentication system.

Subsequently, the current state-of-art in speaker verification system utilizes the i-vector which is based on the total variability subspace [29]. I-vector is a representation of a speaker's utterance that are extracted using a low- dimensional total variability subspace. Speaker verification scores are obtained by the dot product between the test and speaker i-vector model. Further, the scores are normalized to minimize the acoustic variation between the training and testing conditions. The normalized scores are compared with the threshold to make the final decision. In forensic speaker verification, a relevant background population is used for estimating the likelihood of a random match based on the acoustic evidence collected. Given a large number of speakers, the relevant background population can be found for cohort selection. In a speaker verification system, this approach is realized using the T-Norm (Test normalization) method. This technique is test dependent, where the impostor score distribution is estimated for each test utterance by performing non-target trials with the cohort [30]. Then the similarity between the test and target utterance is normalized by using this distribution.

Motivation A phoneme contains valuable information about the speaker and has been used to improve the performance of a speaker verification system [16, 24, 25]. Most of the work related to normalization in speaker verification is based on the Gaussian Mixture Model-Universal Background Model (GMM-UBM) method. Recently, the concept of i-vector was proposed, which outperforms the GMM-UBM based speaker verification method [27, 29]. In general, score normalization is applied after the scores are obtained. A new scoring method is proposed in [10], where the score normalization step is merged with the final scoring technique. This normalized scores are computed by using the average of the impostors i-vectors. The overall success of this scoring method greatly depends on the proper selection of the impostors in order to obtain the average i-vector parameters.

Contributions Contribution of this work is as follows

- Determining the effects of incorporating phoneme level information into text-dependent speaker verification. This has been implemented for English as well as for Hindi speakers.
- A novel technique for cohort selection in the i-vector framework has been proposed and its performance has been compared against the standard normalization methods.
- A new speaker verification system based on majority voting method which uses multiple threshold in final scoring has been proposed.

It is found that the proposed methods lead to significant improvement in performance over the conventional methods.

The remainder of this paper is structured as follows. Section 2 presents an overview of the speaker verification and score normalization methods in the i-vector framework. The channel compensation methods to reduce the channel and inter-session variability at the model level is also discussed. Section 3 describes the method for cohort selection based on i-vector model cluster. It includes the selections based on speaker specific, phoneme specific and a combination of both methods. A late fusion method that utilizes the majority voting on normalized scores are also discussed herein. Section 3 discuss the performance evaluation of the proposed method based on experiments conducted on TIMIT, HINDI and YOHO databases. The performance of the proposed method is evaluated in terms of equal error rate (EER) and minimum decision cost function (min DCF). Section 5 concludes the paper with a discussion on the results obtained and possible future applications and extensions.

2 Speaker verification and score normalization in the I-vector framework

In this section, an overview of the speaker verification system in the i-vector framework has been presented. It also discussed the standard normalization methods used in the i-vector framework. Standard methods of channel compensation techniques, i.e Linear Discriminant Analysis (LDA) and Within-Class Covariance Normalization (WCCN) are also explained.

The task of speaker verification is a hypothesis testing problem and can be formulated in the following manner. For the given test utterance Y, with a claimed identity S. The null hypotheses, $H0 : Y$ is from the claimed speaker S and the alternative hypothesis, $H1 : Y$ is not from the claimed speaker S. The verification selects one of the hypothesis. In Fig. 1, the block diagram for the speaker verification system in the i-vector framework is illustrated. Extraction of i-vectors from the speech utterances is explained in the ensuing section.

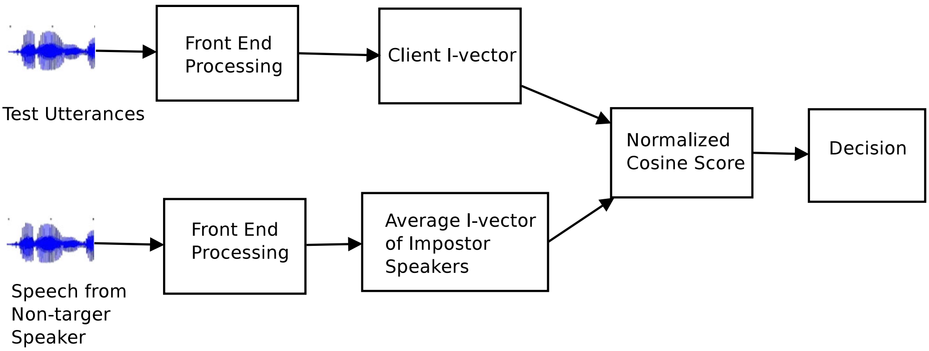

Fig. 1 Block diagram of the i-vector based speaker verification system

2.1 I-vector extraction

I-vector is a low dimensional representation of the utterances. In GMM-UBM system, a speaker model is obtained by adapting a background model to data of a target speaker. A background model is a GMM with large number of mixture components trained on the features from non-target speakers speeches. The main problem of this adaptation technique is the adaptation of non-speaker parameter (channel and other non-speaker factor) along with the speaker specific parameter. In the factor analysis method, a GMM mean supervector of target model is assumed to have speaker independent, speaker dependent, channel dependent and speaker dependent residual components [19]. The supervector is obtained by concatenation of means of all the mixture components of the target model [7].

Combining all the variable component in one matrix is given in [29] and termed as total variability subspace. In this approach, a supervector is assumed to have the following structure.

$$\mathbf{s} = \mathbf{m} + \mathbf{T}\mathbf{w} \tag{1}$$

Where **m** is the UBM supervector and **T** is the total variability matrix and **w** is the total variability factor, termed as the i-vector. The training of the matrix **T** is done in exactly the same way as that of the speaker subspace in the JFA approach with a slight modification by assuming the speech utterances from the same training speaker as from the different speakers. Matrix **T** is a low rank matrix. For the given matrix **T**, i-vector **w** is obtained for the given utterance. The i-vector representing the utterance is calculated by:

$$\boldsymbol{w} = \left(\boldsymbol{I} + \boldsymbol{T}^t \boldsymbol{\Sigma}^{-1} \boldsymbol{N} \boldsymbol{T}\right)^{-1} \boldsymbol{T}^t \boldsymbol{\Sigma}^{-1} \boldsymbol{F} \tag{2}$$

where **I** is the identity matrix and **N** is a diagonal matrix of dimension $CF \times CF$, its diagonal blocks are $N_c\mathbf{I}$, $(c = 1, 2, ..C)$ where c is the Gaussian index and **F** is the supervector formed by concatenating all the centralized first order statistics. $\boldsymbol{\Sigma}$ is a diagonal covariance matrix of dimension $CF \times CF$. The block diagram in Fig. 2 illustrates the procedure for extracting the i-vector. The i-vector extraction itself does not perform any channel and inter-session compensation method. LDA and WCCN methods are used to compensate the channel and inter-session variability from the i-vector which is explained in the next section.

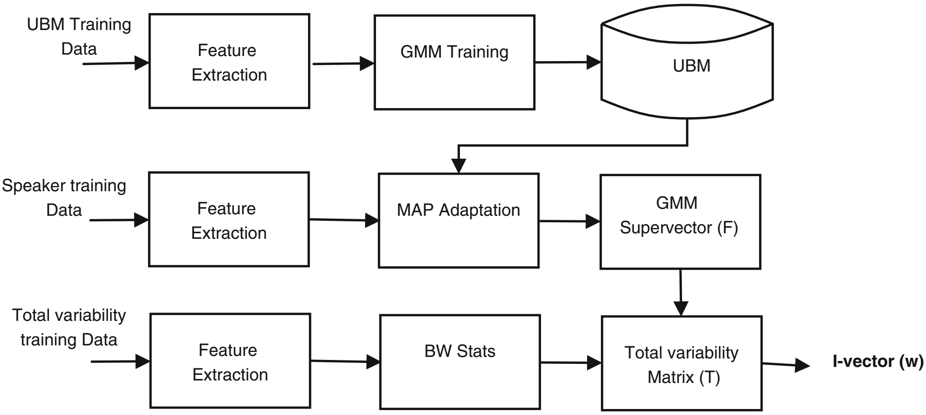

Fig. 2 Diagram illustrating the i-vector extraction from speech dataset

2.2 Linear discriminant analysis method

Linear Discriminant Analysis (LDA) is applied to compensate for the inter-session and channel variability in speech data. The main objective of LDA method is to find new orthogonal axes which maximizes between class variation and minimizes within class variations. This leads to getting better discrimination between different classes and reduces the dimensionality of the data. The LDA transformation matrix $\mathbf{A}_{LDA}$ consists of the eigenvectors having the largest eigenvalues of the eigenvalue problem $\boldsymbol{S}_B \boldsymbol{v} = \lambda \boldsymbol{S}_W \boldsymbol{v}$, where the between and within speaker scatter matrices, $\boldsymbol{S}_B$ and $\boldsymbol{S}_W$ respectively are calculated using

$$\boldsymbol{S}_B = \sum_{s=1}^{S} N_s \left(\boldsymbol{\mu}_s - \boldsymbol{\mu}\right)\left(\boldsymbol{\mu}_s - \boldsymbol{\mu}\right)^t \tag{3}$$

$$\boldsymbol{S}_W = \sum_{s=1}^{S} \sum_{i=1}^{N_s} \left(\boldsymbol{w}_i{}^s - \boldsymbol{\mu}_s\right)\left(\boldsymbol{w}_i{}^s - \boldsymbol{\mu}_s\right)^t \tag{4}$$

where $\boldsymbol{\mu}_s$ is the mean i-vector of each speaker, S denotes the total number of speakers in consideration and N_s stands for the total number of utterances for speaker s. N is the total number of sessions. The mean i-vector for each speaker $\boldsymbol{\mu}_s$ and mean i-vector across all the speaker $\boldsymbol{\mu}$ is defined as

$$\boldsymbol{\mu}_s = \frac{1}{N_s} \sum_{i=1}^{N_s} \boldsymbol{w}_i{}^s \tag{5}$$

$$\boldsymbol{\mu} = \frac{1}{N} \sum_{s=1}^{S} \sum_{i=1}^{N_s} \boldsymbol{w}_i{}^s \tag{6}$$

The matrix $\mathbf{A}_{LDA}$ is calculated as follows:

$$\boldsymbol{A}_{LDA} = \arg \max_{\boldsymbol{A}} \frac{|\boldsymbol{A}^T \boldsymbol{S}_B \boldsymbol{A}|}{|\boldsymbol{A}^T \boldsymbol{S}_W \boldsymbol{A}|} \tag{7}$$

2.3 Within-class covariance normalization

In sequence to LDA Within-Class Covariance Normalization (WCCN) [15] method is applied to reduce the within speaker variance that remains after LDA. This helps in reducing the within class split and leads to an improved system performance. The WCCN matrix B is found by the Cholesky decomposition of $\mathbf{W}^{-1} = \mathbf{B}\mathbf{B}^{t}$, where the within-class covariance matrix is calculated by:

$$\boldsymbol{W} = \frac{1}{S}\sum_{s=1}^{S}\sum_{i=1}^{N_s}\left(\boldsymbol{A}_{LDA}{}^{t}\boldsymbol{w}_i{}^{s} - \hat{\boldsymbol{\mu}}_s\right)\left(\boldsymbol{A}_{LDA}{}^{t}\boldsymbol{w}_i{}^{s} - \hat{\boldsymbol{\mu}}_s\right)^{t}. \tag{8}$$

Where, $\mathbf{A}_{LDA}$ is the LDA projection matrix, $\hat{\boldsymbol{\mu}}_s$ is the mean of the LDA projected i-vector of each speaker s and S is the total number of speakers.

2.4 Cosine scoring

The cosine score for a trial between a set of test and target i-vectors $\boldsymbol{w}_{target}$ and $\boldsymbol{w}_{test}$ is given by the dot product $\left(\boldsymbol{w}'_{target}, \boldsymbol{w}'_{test}\right)$ between the inter-session compensated normalized i-vectors.

$$score\left(\boldsymbol{w}'_{target}, \boldsymbol{w}'_{test}\right) = \frac{\langle \boldsymbol{w}'_{target}, \boldsymbol{w}'_{test}\rangle}{\|\boldsymbol{w}'_{target}\|\,\|\boldsymbol{w}'_{test}\|} \tag{9}$$

Where $\|\boldsymbol{w}'_{target}\|$ and $\|\boldsymbol{w}'_{test}\|$ are the L_1-norm of the w'_{target} and w'_{test} respectively. This score is compared with the threshold to make a final decision for speaker verification. It should be noted that, both target and test i-vectors are estimated exactly in the same manner, i.e. using the same UBM and same total variability subspace. The use of the cosine score as a decision score for speaker verification makes the process faster and less complex.

2.5 Score normalization in the I-vector framework

In the i-vector framework the normalization (Both T-Norm and Z-Norm) can be incorporated [10] in the score computation step as

$$s_{znorm} = \frac{\left(\boldsymbol{w}'_{target}\right)^{t}\left(\boldsymbol{w}'_{test} - \boldsymbol{w}'_{z\_imp}\right)}{\|\boldsymbol{C}_{z\_imp}.\boldsymbol{w}'_{target}\|} \tag{10}$$

$$s_{tnorm} = \frac{\left(\boldsymbol{w}'_{target} - \boldsymbol{w}'_{t\_imp}\right)^{t}\left(\boldsymbol{w}'_{test}\right)}{\|\boldsymbol{C}_{t\_imp}.\boldsymbol{w}'_{target}\|} \tag{11}$$

In (10), s_{znorm} is the Z-Norm score, $\boldsymbol{w}'_{target}$ is the target i-vector, $\boldsymbol{w}'_{test}$ is the test i-vector. $\boldsymbol{w}'_{z\_imp}$ is the average impostor i-vector and $\boldsymbol{C}_{z\_imp}$ is a diagonal matrix containing the square root of diagonal Z-Norm impostor's covariance matrix. Similarly, in (11), s_{tnorm} is T-Norm score, $\boldsymbol{w}'_{target}$ is the target i-vector, $\boldsymbol{w}'_{test}$ is the test i-vector. $\boldsymbol{w}'_{t\_imp}$ is average impostor i-vector and $\boldsymbol{C}_{t\_imp}$ is a diagonal matrix containing the square root of diagonal T-Norm impostor covariance matrix. A combined scoring method is given as in (12). It combines the effect of both Z-Norm and T-Norm method.

$$s_{ztnorm} = \frac{\left(\boldsymbol{w}'_{target} - \boldsymbol{w}'_{t\_imp}\right)^{t}\left(\boldsymbol{w}'_{test} - \boldsymbol{w}'_{z\_imp}\right)}{\|\boldsymbol{C}_{t\_imp}.\boldsymbol{w}'_{target}\|\,\|\boldsymbol{C}_{z\_imp}.\boldsymbol{w}'_{test}\|} \tag{12}$$

The performance of the i-vector based speaker verification system depends on the average impostor i-vectors $\boldsymbol{w}_{t\_imp}$ and $\boldsymbol{w}_{z\_imp}$ estimation . If this estimation is made based on speakers that are closer to the target speakers then it leads to improved speaker verification system performance. $\boldsymbol{w}_{z\_imp}$ is the average impostor i-vector for z-norm which is computed over a set of utterances from speakers different than the target speaker (i.e., impostor). Here in this work, the training i-vectors of a target model are used for selecting the impostors close to the target model using cosine similarity. Once the set of impostor is selected, $\boldsymbol{w}_{z\_imp}$ is computed as

$$\boldsymbol{w}_{z\_imp} = \frac{w_{z\_imp1} + w_{z\_imp2} + ... + w_{z\_impN}}{N} \tag{13}$$

where $wz_{imp1}, wz_{imp2}, ...wz_{impN}$ are set of N impostor for the target model. In T-Norm scoring, the task is to select the speakers for the cohort, which are nearest to the given target model and diverse from each other [31]. By selecting cohort for the same IMC as that of the target it is guaranteed that impostors near to the target model are selected. In this work, $\boldsymbol{w}_{t}\_imp$ is estimated by using the test i-vectors of the target model and a set of impostor i-vectors. The next section will describe the cohort selection method and the combined method for speaker verification using phoneme level information.

3 Cohort selection using speaker and phoneme specific I-vectors

In this section, we present the speaker clustering method in the i-vector framework. Cohort selection is discuses by using speaker and phoneme specific properties. After that Majority voting method is presented which is fused with cohort normalization to improve the speaker verification system performance. Iterative proportional fitting procedure method for normalizing the confusion matrix is also discussed herein.

3.1 I-vector based clustering method for cohort selection

Earlier, in the GMM-UBM based speaker verification method, speakers were clustered using the k-means clustering algorithm to improve the performance of the system [1]. This approach gives speedups in the speaker identification process as well as improvements in speaker verification [1, 2]. We propose an algorithm to cluster speaker models by using i-vectors that utilize the test utterances for grouping the similar set of speakers. This leads to more meaningful clustering of the speakers. We presented it in Fig. 3 that shows the space of the i-vector model cluster, claimant speaker i-vector and cohort i-vectors.

To cluster the speakers, a confusion matrix is generated by scoring the test utterances with the target models using the dot product.The confusion matrix is then normalized using the Iterative proportional fitting method [12]. The IPFP algorithm normalizes the confusion matrix so that each row and each column individually sums to one and that is necessary for appropriately measuring the similarity between the rows. A simple distance metric, that quantifies the similarity between speakers is utilized to cluster similar sets of speakers in multiple passes. In first pass two closest speakers are grouped into one class and this process is repeated until a threshold in terms of accuracy is achieved. The i-vector representation of the clustered speaker is achieved by taking the average of the i-vectors of a similar set of speakers. This i-vector based model cluster aids efficient speaker identification and speaker verification scoring. Figure 4 illustrates the flow diagram of the proposed method for speaker clustering in the i-vector framework for speaker verification.

Algorithm 1 lists the steps involved in speaker clustering in the i-vector framework.

Algorithm 1 Speaker clustering method in the i-vector framework

```
1: Input: I-vectors of test utterances.
2: Obtain the confusion matrix by scoring the test utterances against the target model.
3: Normalize the confusion matrix using IPFP method.
4: L1 metric applied to obtain an upper triangular matrix that gives a pairwise similarity
   between the target models.
5: Repeat step 2 with the clustered speakers obtained in step 4 until a threshold in terms
   of accuracy is obtained.
6: Output: Cluster of similar speakers.
```

Once similar group of speakers is clustered, speaker specific cohorts are selected for normalization which is discussed in the next subsection.

3.2 Speaker specific cohort selection (SSCS)

The SSCS is an online method for selection of the cohort and it is a data driven approach [28]. In [28], a method called client-wise cohort set selection (CWCS) was proposed, which uses client-wise speaker specific properties to obtain the cohort set in a GMM-UBM based system. For the i-vector framework, it is termed as SSCS method. In this method, speakers for cohort are selected during the matching phase. This method has an additional advantage over offline cohort selection as it selects the speakers which are test dependent. Figure 5 shows a case where the offline selection fails.

The closest impostors to the target model are on the wrong side of the test i-vectors. Consequently, the score for the target model will be higher than for any of the cohort.

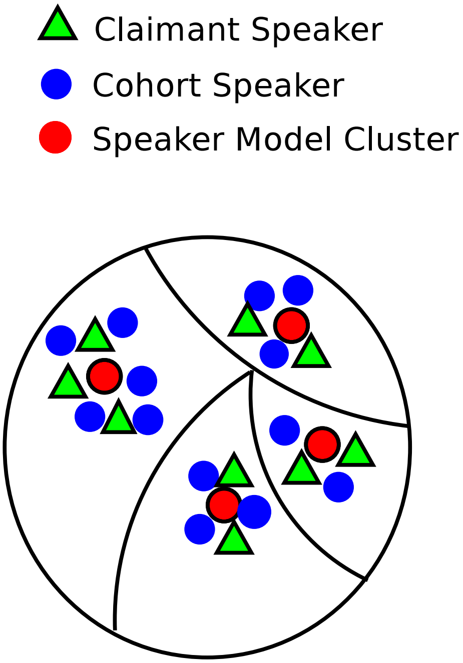

Fig. 3 Space of i-vector model clusters, cohort and claimant speakers

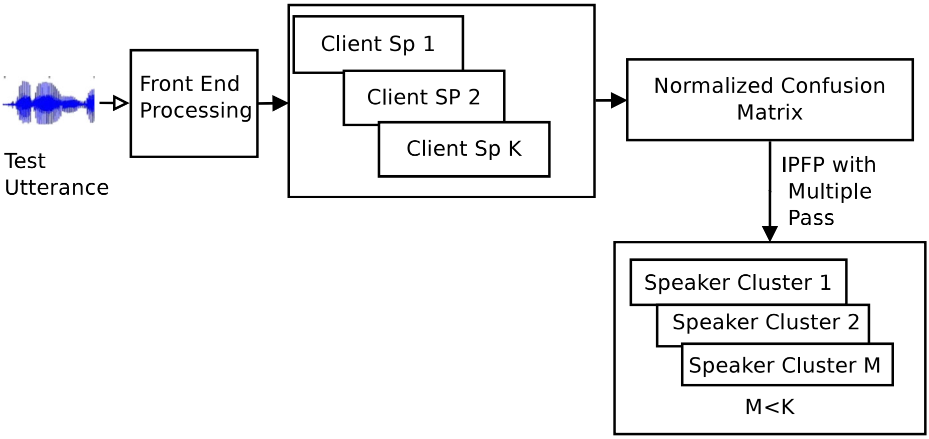

Fig. 4 Flow diagram illustrating the Speaker Model Clustering in the i-vector Framework

Thus, it wrongly accepts the claimant. However, it should be noted that the probability of the test i-vector belongs to the right target model which is very high. But, when an online selection is performed, the i-vector corresponding to the closest impostor to the test sequence (client speaker) are selected for scoring. Therefore, it is expected that the number of false acceptances is reduced by using an online approach. The block diagram is illustrated in Fig. 6.

Algorithm 2 lists the steps involved in speaker specific cohort selection.

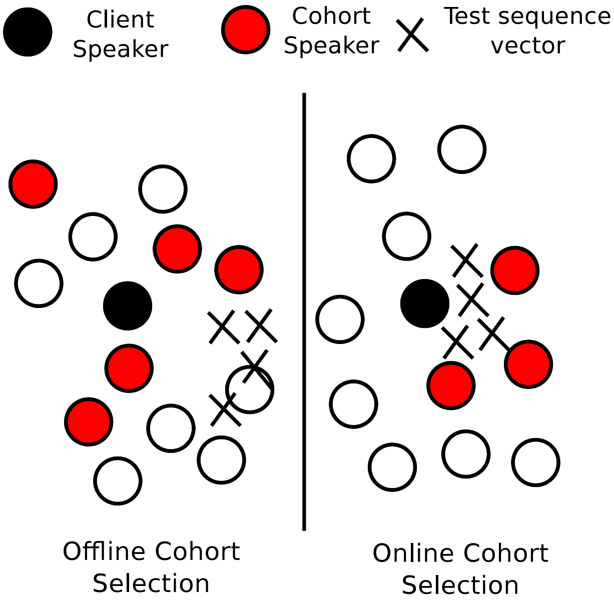

Fig. 5 Illustration of online and offline cohort speakers

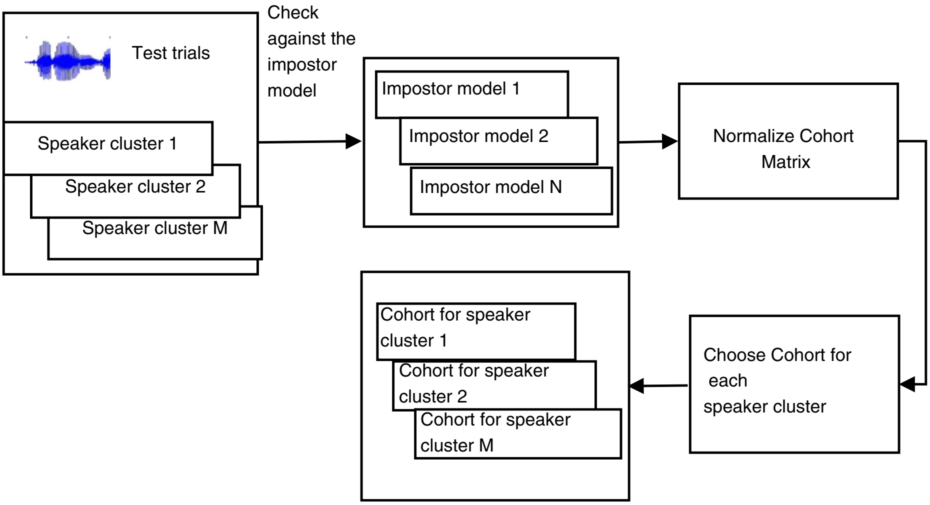
Test trials
Speaker cluster 1
Speaker cluster 2
Speaker cluster M
Check against the impostor model
Impostor model 1
Impostor model 2
Impostor model N
Normalize Cohort Matrix
Choose Cohort for each speaker cluster
Cohort for speaker cluster 1
Cohort for speaker cluster 2
Cohort for speaker cluster M

Fig. 6 Block diagram illustrating the cohort set selection using SSCS method

Algorithm 2 Speaker specific cohort selection

```
1: Input: The input test utterances.
2: Generate cohort matrix by scoring the test utterance with the impostor models.
3: Normalize the cohort matrix using IPFP method.
4: Apply L1 distance metric that quantifies the similarity in terms of cell probability
   values.
5: Closest cohorts are pruned from the row wise candidates of the confusion matrix.
6: Output: Cohort of corresponding i-vector based speaker cluster model is obtained.
```

To get the best cohort, the impostor trials for each speaker model cluster has been started and corresponding cohort matrix is obtained. Here the name cohort matrix is assigned as row wise and column wise models are different as compared to confusion matrix where we have the same model both in rows and column. The cohort matrix is then normalized using the iterative proportional fitting procedure (IPFP) method [18]. In order to select the cohort set from normalized cohort matrices, L1 distance metric [13] is used as it sums the absolute differences of the corresponding coordinate values between two row vectors. It must be noted here that, the row wise candidates of the cohort matrix are the claimed identities. Hence, the possibility of pruning one closest speaker in the selection of the cohort is possible.

The most closest speakers to the target speakers are selected from the cohort set. After the selection, each i-vector model cluster will have its own set of cohort. This cohort selection method leads to better speaker verification system performance compared to conventional cohort selection methods, which do not perform any similarity modeling and use the same set cohort for all claimed speaker models.

3.3 Phoneme specific cohort selection (PSCS)

A phoneme is the smallest distinct unit of sound and has been used to discriminate between two utterances. Speaker specific phoneme models, i.e. modeling the way individuals produce different sounds can give important information about the identity of the claimed speaker [11].

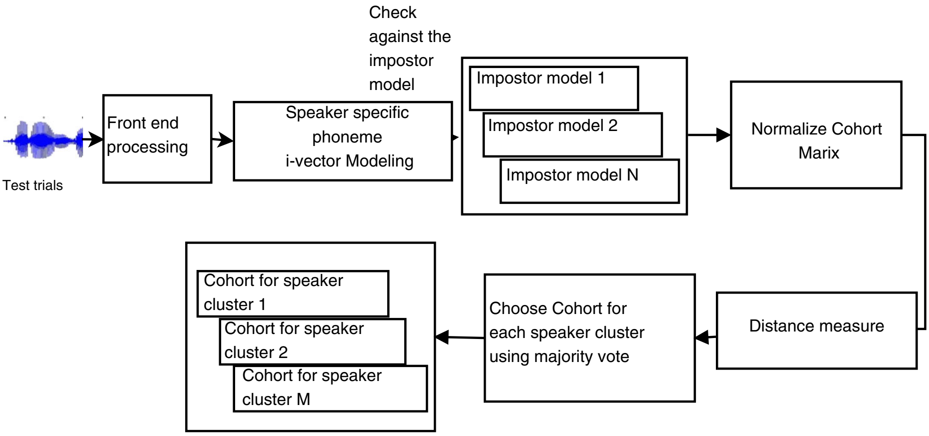
Check against the impostor model
Test trials
Front end processing
Speaker specific phoneme i-vector Modeling
Impostor model 1
Impostor model 2
Impostor model N
Normalize Cohort Marix
Distance measure
Choose Cohort for each speaker cluster using majority vote
Cohort for speaker cluster 1
Cohort for speaker cluster 2
Cohort for speaker cluster M

Fig. 7 Block diagram illustrating the cohort selection using PSCS method

Phoneme based cohort set selection uses speaker specific phoneme level information for the selection of a cohort for every clustered speaker model. For speaker specific phoneme i-vector modeling, most frequently occurring phonemes are selected by ranking them according to their frequency in the training database. Acoustic features corresponding to these phonemes are extracted. Baum-Welch statistics are then estimated from these acoustic features. These statistics along with the total variability subspace are used in extracting the speaker specific phoneme i-vectors. Instead of test i-vectors of the clustered speaker, the speaker specific phoneme i-vectors of the clustered speaker are used to generate the cohort matrices by scoring it against the non-target speakers. This cohort matrix gives the proximity between the phoneme of the test utterance and cohort models. The cohort matrix is normalized using IPFP to get the cohort sets. The final cohort set for each i-vector model cluster is selected using majority voting for frequently occurring phonemes. Phoneme level transcription of HINDI and TIMIT database are available to extract the speaker specific phoneme features. Experiments are also performed where time level phoneme alignment is not provided. To explore the significance of PSCS in such a case, a speech recognizer is used to get the time level phoneme alignment. YOHO database is used for this method. The speech recognizer is trained using the training utterances of YOHO database. HTK toolkit is used herein for obtaining the phoneme level time alignment for YOHO database [37]. Figure 7 shows the block diagram and Algorithm 3 for selecting the cohort set using speaker specific i-vectors.

Algorithm 3 lists the steps involved in the implementation of the proposed method.

Algorithm 3 Phoneme specific cohort selection method

```
1: Input: I-vectors of the input test utterances.
2: Phoneme test i-vectors are scored against the impostor models.
3: Repeat step 2 for 5-6 most occurring phonemes.
4: Obtain the cohort matrices corresponding to the phonemes.
5: Normalize the cohort matrices using IPFP method.
6: L1 metric applied in the normalized cohort matrix to prune the closest cohort from row
   wise candidates for the corresponding phonemes.
7: Cohorts corresponding to the clustered speaker are obtained by majority voting.
8: Output:Cohort of corresponding claimant speaker.
```

3.4 Combining the speaker and phoneme specific method for cohort selection

In this method, combined information from both SSCS and PSCS methods is utilized to get the final cohort set. First set of cohort is obtained using SSCS method as described in the earlier section, which uses speaker level information. The second set of cohort is obtained using PSCS method, which uses phoneme level information. A final set of cohort is obtained by combining the cohort sets obtained using both these methods. Hence, the final set of cohort is based on both speaker specific properties and phoneme specific properties . Figure 8 illustrates the combined approach.

The final cohort set obtained will be used for score normalization in the speaker verification method as discussed in the ensuing section.

3.5 Cohort normalization in the I-vector framework

For each speaker the test data set is divided into two sets of trials known as target trials and non-target trials. The average impostor i-vector is estimated using the cohorts selected by the SSCS and PSCS methods. Subsequently, normalized scores are obtained using (12). It is noted here that, the cohort used in obtaining impostor utterances for Z-Norm scoring is different than the cohort set selected for T-Norm scoring. We have used the training i-vectors to score against the non-target i-vectors to generate the cohort matrix to estimate the $\boldsymbol{w}_{z\_imp}$.

In T-Norm scoring, the task is to select the cohort models, which are nearest to the given claimant model (target model) and diverse from each other [31]. In the method proposed here, $\boldsymbol{w}_{z\_imp}$ is estimated offline (using training i-vectors) and $\boldsymbol{w}_{t\_imp}$ is estimated online (using the test i-vectors). $\boldsymbol{C}_{t\_imp}$ and $\boldsymbol{C}_{z\_imp}$ are diagonal matrices, that contain the square root of the diagonal covariance matrix of the impostor i-vectors.

Further, to improve the performance of the speaker verification system the majority voting method is used in this work [21]. Once normalized scores have been obtained with the new scoring method, the only step remaining is the final decision in which the claimant speaker is either accepted or rejected. This is accomplished by setting a threshold,

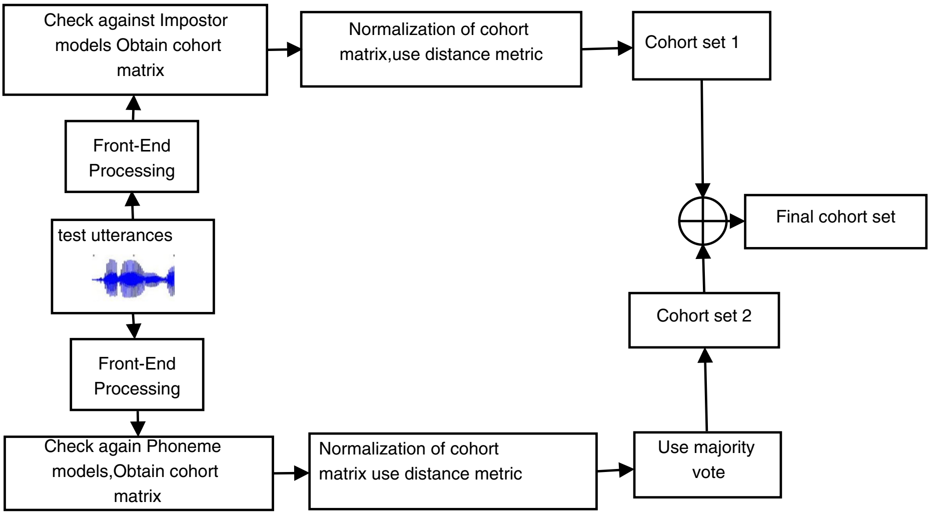
Check against Impostor models Obtain cohort matrix
Normalization of cohort matrix,use distance metric
Cohort set 1
Front-End Processing
Final cohort set
test utterances
Cohort set 2
Front-End Processing
Check again Phoneme models,Obtain cohort matrix
Normalization of cohort matrix use distance metric
Use majority vote

Fig. 8 Block diagram illustrating the combined approach for cohort selection

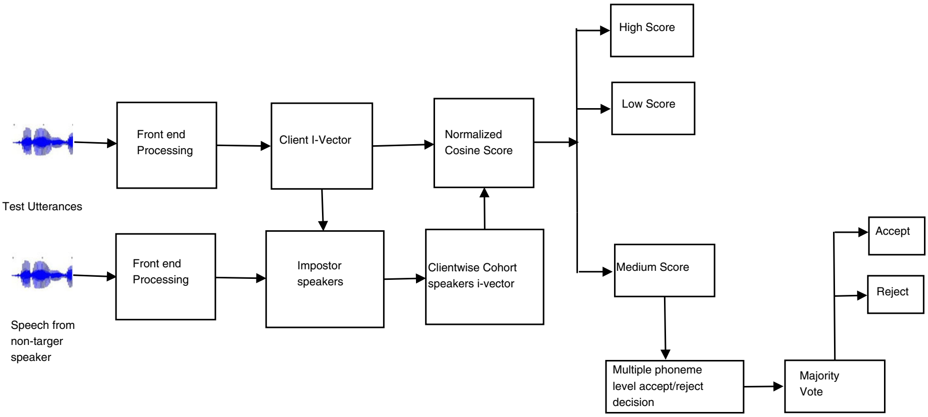

Fig. 9 Block diagram illustrating the recognition phase with normalization in the proposed method

all utterances producing scores above the set threshold are accepted as coming from the claimant speaker, while all utterances producing scores below the set threshold are rejected as coming from the impostor speakers. In this method, use of multiple thresholds is proposed to improve speaker verification decisions. Two thresholds are set, a low threshold and high threshold. Utterances producing scores above the high threshold are accepted, while those producing scores below the lower threshold are rejected. For scores, lying between the two thresholds further processing is done using phoneme specific i-vector models. Multiple phoneme level accept and reject decisions are made and a final decision to accept or reject is made by majority voting across all phonemes. Figure 9 shows the recognition phase of the speaker verification system using the majority voting method. In the next section, IPFP method is described which is used for normalizing the confusion matrix.

3.6 Iterative proportional fitting procedure (IPFP)

The IPFP method was proposed to estimate the cell probabilities in a contingency table [12]. This method estimates cell probabilities in a confusion matrix by forcing each row and column sum equal to one. Based on some marginal constraints these cell probabilities are estimated. Let us consider $n_{ij}, n_{ij} > 0$ observations in a confusion matrix (r × c), where

$$\sum_{i=1}^{r}\sum_{j=1}^{c} n_{ij} = n \tag{14}$$

initial values are taken as.

$$p_{ij}^{0} = n_{ij}/n \tag{15}$$

The following minimization criterion has been applied in estimating the cell probabilities.

$$\sum_{i=1}^{r}\sum_{j=1}^{c}(n_{ij} - np_{ij})^2/n_{ij} \tag{16}$$

The following fixed marginal constraints have been assumed.

$$p_{i+} = \sum_{j=1}^{c} p_{ij} \;(i = 1, 2, ..., r) \tag{17}$$

$$p_{+j} = \sum_{i=1}^{r} p_{ij} \; (j = 1, 2, ..., c) \tag{18}$$

where $p_{i+} = 1$ and $p_{+j} = 1$, since the sum of cell probabilities in every individual row and column in the resultant normalized confusion matrix should be equal to one. Using the above theory, the following iterative steps have been used for matrix normalization.

(1) At every even step (m $\geq$ 1) and to estimate the probability values, following expression is evaluated:

$$p_{ij}^{2m-1} = p_{ij}^{2m-2} . p_{i+} / p_{i+}^{2m-2} \tag{19}$$

(2) Similarly, at every odd step the probability values are estimated as:

$$p_{ij}^{2m} = p_{ij}^{2m-1} . p_{+j} / p_{+j}^{2m-1} \tag{20}$$

The iterations are continued until two successive sets of values for the cell probabilities agree sufficiently well. As the no. of iterations become infinitely large, i.e., N → ∞

$$p_{ij}^{N} \rightarrow p_{ij} \tag{21}$$

where the p_{ij} satisfies the marginal total condition.

4 Performance evaluation

In this section, the performances of the proposed methods are evaluated through speaker verification experiments conducted on the TIMIT [14], HINDI [9] and YOHO [6] databases. DET curves and EER are used to evaluate the performance of the proposed methods. They are compared with the raw cosine scoring and ZT-Norm scoring methods. The significant improvements obtained in terms of the DET curves and the EER using the proposed method.

4.1 Description of the data sets

Three data sets, TIMIT, HINDI and YOHO database are used in testing the proposed methods. A combination of subsets of theses databases has been used for UBM development.

- **TIMIT Database**: TIMIT database contains a total of 6300 sentences, 10 sentences are spoken by each of 630 speakers from 8 major dialect regions of the United States. The text material in the TIMIT prompts consists of 2 dialects “shibboleth” sentences designed at SRI, 450 phonetically-compact sentences designed at MIT, and 1890 phonetically-diverse sentences selected at TI. The phonetically-compact sentences were designed to provide a good coverage of pairs of phones. The phonetically-diverse sentences (the SI sentences) were selected from existing text sources - the Brown Corpus [23] and the Playwrights Dialog [17] - so as to add diversity in sentence types and phonetic contexts. The selection criteria maximized the variety of allophonic contexts found in the texts.
- **HINDI Database**: The HINDI database is taken from the Database of Indian Languages [9]. The database contains recordings of 20 news bulletins and features both male and female speakers. Phoneme level transcription of HINDI database is available to extract the phoneme level features.

- **YOHO Database**: YOHO Database: YOHO database is primarily used for performing speaker verification system performance. It consists of 108 male and 30 female speakers. The database was collected during testing of ITT'S speaker verification system in an office environment. Speaker variation spanned a wide range over attributes like age, job description and educational background. Most of the speakers are from the New York city area with some non-native English speakers. Each Speaker has four enrollment session in a well defined train and test scenarios. They are prompted to read a series of twenty four combination loch phrases. Each phrase is a sequence of three two-digit numbers. There are ten verification trials per speaker, consisting of four phrases per trial. The data were collected using a high quality telephone handset (Shure XTH-383), but did not pass through the telephone channel [6].

4.2 Experimental conditions

The features used for the experiments in the system are the MFCCs. The features are extracted as follows. First, the speech signal is sampled at 8 kHz, followed by silence removal using the parameter is given in Table 1.

After this, the speech signal is divided into frames of length 20 ms with a frame overlap of 10 ms. Then each frame of speech is multiplied by a Hamming window to reduce any discontinuities at the edges of the frames. 22 mel filter banks are used to obtain the features. The features used in the experiments are the thirteen dimensional MFCCs appended with velocity and acceleration coefficient's, resulting in thirty nine dimensional feature vectors.

A subset of the TIMIT, HINDI and YOHO databases are used for the background modeling. The background model thus contains speech from a large number of non-target speakers speaking diverse languages. The background model is a GMM having 512 mixture component is trained using the iterative EM algorithm. Generally five to ten iterations are sufficient for parameter convergence. A total variability factor of 400 is used as defined by the total variability matrix **T** for I-vector extraction using large amount of non target speakers from these databases. The client i-vector models are derived by using the total variability subspace and MAP adapted GMM supervector. The ratio of true trials to false trails of 0.157 and 0.148 is used for evaluating the proposed method for TIMIT and HINDI database. Also a true to the false ratio of 0.124 is used for evaluating the proposed method on the YOHO database. Cohort i-vector models were randomly selected for TIMIT, HINDI and YOHO databases.

4.3 Performance measure

In this section, performance measure methods are described for the experiments. A speaker verification system makes the decision based on the following criteria.

$$D(x) = \begin{cases} \text{accept} & \text{if } D(x) > \Delta \\ \text{reject} & \text{otherwise} \end{cases}$$

where Δ is the threshold and $D(x)$ is the score obtained. Two types of error are possible in decision making step, false acceptance (FA) and false rejection (FR). FA means accepting

Table 1 Parameters used for pre-processing of the speech signal

Min silence duration	1 ms
Max silence duration	170 ms
Silence compression	1:1
Threshold for silence	−30 dB

an impostor, and FR refers to rejecting a genuine speaker. False Acceptance Rate (FAR) and False Rejection Rate (FRR) are the normalized version of false alarm (FA) and false rejection (FR) and defined as

$$FAR = \frac{FA}{N^I} \tag{22}$$

$$FRR = \frac{FR}{N^C} \tag{23}$$

where FA and FR are the number of false errors and the number of false miss errors, N^I is the total number of impostor attempts and N^C is the total number of legitimate attempts. Equal Error Rate (ERR) is defined as the FAR that is equal to FRR. Thus, smaller ERR signifies better system performance. Another performance measure such as DCF is defined as

$$DCF = (C_{FRR} \times FRR \times P_T) + (C_{FAR} \times FAR \times (1 - P_T)) \tag{24}$$

where C_{FAR}, C_{FRR} are defined as the cost of false acceptance and cost of false rejection respectively. P_T defined as the prior probability of the likelihood of the test utterance belongs to the claimed speaker. Min DCF is defined as the minimum value of the DCF that can be achieved on the test data. It can be found by choosing the score threshold such that it minimizes the (24) on the test data. These DCF parameters were set as C_{FRR}=10, C_{FAR}= 1, and P_T = 0.01 to evaluate the proposed method. In addition to the single measurement of DCF and EER value, more information can be shown as a graph. The graph plotting has all the operating point corresponds to a likelihood threshold for separating the actual decision of true and false.

The DET curve was first introduced by Martin in 1998 [26]. It replaced the ROC (Receiver Operating Characteristic) curve, which was used for visualizing the relationship between the true positive rate and false positive rate. The ROC curves are not in general a good representation for measuring the performance of the speaker verification system with

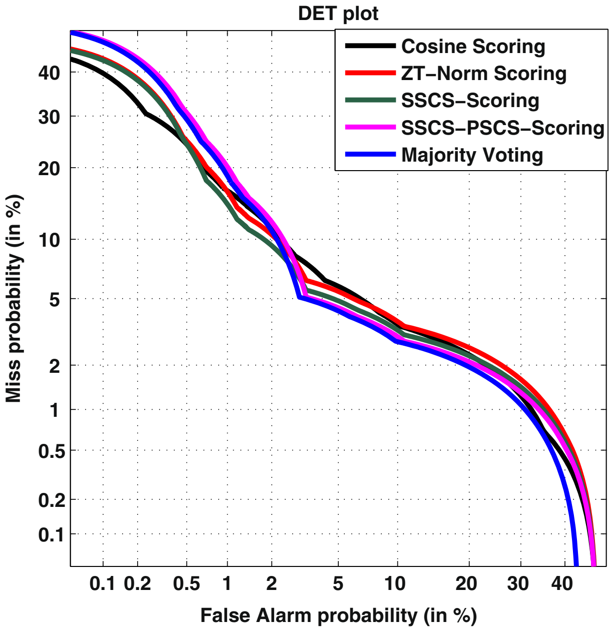

Fig. 10 DET plots of speaker verification experiment for the TIMIT database

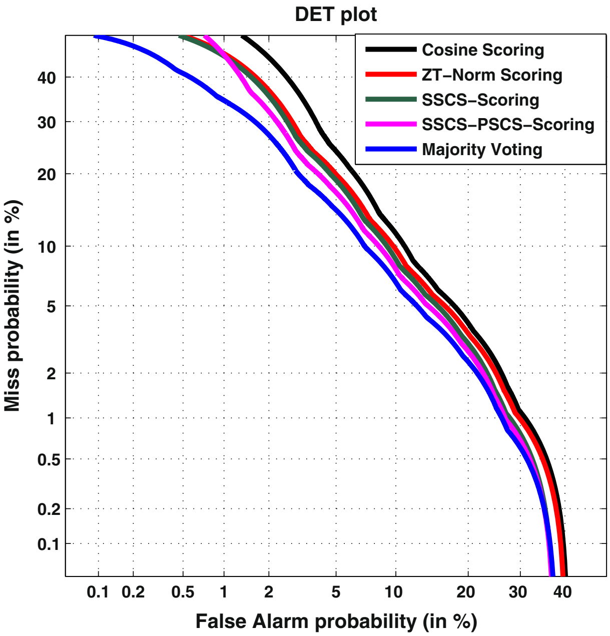

Fig. 11 DET plots of speaker verification experiment for the HINDI database

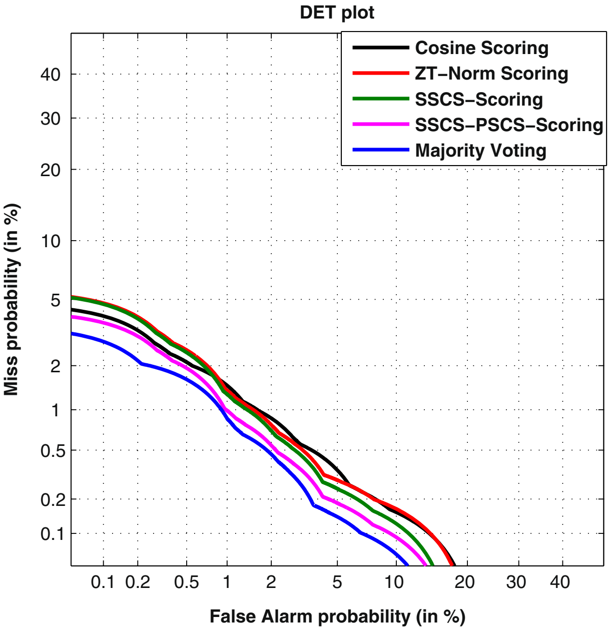

Fig. 12 DET plots of speaker verification experiment for the YOHO database

Table 2 min DCF and EER values for for TIMIT database

Scoring method	EER	min DCF
Cosine Scoring	5.53	0.1488
ZT-Norm Scoring	5.32	0.1388
SSCS-Norm Scoring	4.92	0.1319
SSCS-PSCS-Norm Scoring	4.59	0.1272
Majority Voting	4.47	0.1204

regard of ease in reading and ability to draw conclusions from it. The DET curves are plotted against two types of error, false acceptance versus false rejection unlike ROC curves which is plotted false acceptance versus true acceptance.

In DET curves, the two errors are plotted on the **x** and **y** axes on a normal deviating scale which makes the curve look linear. They are obtained by varying the threshold from one extreme to the other and record the false alarm and miss rate for each case. The curves obtained in this fashion have the complete information about the system performance at all possible threshold value. In this way, the DET curves allow us to compare the system depends on the algorithm used to choose the decision threshold.

4.4 Experimental results

Experiments on the TIMIT, HINDI and YOHO database are conducted using five methods.

- The cosine scoring method: In this method, normalization is not carried out. Scores are computed from the channel compensated I-vectors. This is the baseline system considered under the i-vector framework.
- ZT-Norm scoring method : This method uses the standard ZT-Norm scoring approach [29]. The $\boldsymbol{w}_{t\_imp}$ and $\boldsymbol{w}_{z\_imp}$ are estimated offline and online respectively from a large number of speakers.
- The speaker specific cohort set selection followed by ZT-Norm scoring method (SSCS-Norm-Scoring): In this method, the client-wise cohort set selection method followed by ZT-Norm scoring is used. Note that a common cohort set is not used for score normalization as in the ZT-Norm scoring schemes. The $\boldsymbol{w}_{t\_imp}$ and $\boldsymbol{w}_{z\_imp}$ are estimated offline and online respectively from speakers that are close to the target speakers.
- The phoneme based cohort set selection followed by ZT-Norm scoring method (SSCS-PSCS-Norm-Scoring): In this method, the phoneme-based cohort set selection approach is used and final cohorts are selected by combining with the cohort selected by the SSCS approach. In this method, a common cohort set is not used unlike the ZT-Norm scoring approach.

Table 3 min DCF and EER values for HINDI database

Scoring method	EER	min DCF
Cosine Scoring	10.58	0.3348
ZT-Norm Scoring	9.85	0.3070
SSCS-Norm Scoring	9.46	0.2954
SSCS-PSCS-Norm Scoring	9.04	0.2816
Majority Voting	8.41	0.2604

Table 4 min DCF and EER values for YOHO database

Scoring method	EER	min DCF
Cosine Scoring	1.22	0.0324
ZT-Norm Scoring	1.18	0.0366
SSCS-Norm Scoring	1.14	0.0354
SSCS-PSCS-Norm Scoring	0.99	0.0305
Majority Voting	0.95	0.0255

- Phoneme based majority voting method: In this method phoneme level accept and reject decisions are made. A final decision is taken by the majority voting method for medium score.

The experimental results obtained are presented in the form of DET curves [26], EER and min DCF values. The DET plots for the TIMIT, HINDI and YOHO databases are shown in Figs. 10, 11 and 12. The Tables 2, 3 and 4 shows the min DCF and EER values for TIMIT, HINDI and YOHO databases respectively. From Figs. 10–12, and Tables 2–4, it is observed that the proposed SSCS-PSCS-Norm and Majority Voting scoring methods show considerable improvement over the standard cosine scoring and ZT-Norm scoring methods.

5 Conclusion

The proposed phoneme based methods are utilizing speaker specific and claimant-specific phoneme models for improving the speaker verification performance. The PSCS-Norm approach uses a client-wise cohort set selection approach using phonemic level comparison of the test signal with speaker specific phoneme models. The cohort set selection selects the closest set of impostor speakers via normalized confusion matrices and takes into account hitherto unused classification errors. Hence, this technique differs from the standard ZT-Norm scoring technique in which the same set of cohort models is used for all test utterances. This approach involves meaningful selection of the cohorts using the test utterances. Further, phoneme information is used with multiple thresholds for medium scores. This leads to further improvement in speaker system verification system performance. Experiments are conducted on two types of database. In one type (TIMIT and HINDI) of database phoneme level transcription is available to extract the phoneme features, in other database (YOHO) we have used the speech recognizer to extract the phoneme features. In both the cases we have got significant improvement in speaker verification system performance.

The usage of phoneme models and cohort sets, if thoroughly formalized has the potential to be of significance in the field of text-dependent speaker verification. Applying better similarity measures for cohort selection and better methods for score computation can significantly improve the speaker verification system performance.

References

1. Apsingekar V, DeLeon P (2009) Speaker model clustering for efficient speaker identification in large population applications. IEEE Trans Acoust Speech Signal Process 17(4):848–853
2. Apsingekar V, DeLeon P (2011) Speaker verification score normalization using speaker model clusters. Speech Comm 53:110–118

3. Auckenthaler R, Carey M, Lloyd-Thomas H (2000) Score normalization for text-independent speaker verification systems. Digital Signal Process 10(1–3):42–54
4. Bimbot F, Bonastre J-F, Fredouille C, Gravier G, Magrin-Chagnolleau I, Meignier S, Merlin T, Ortega-García J, Petrovska-Delacrétaz D, Reynolds DA (2004) A tutorial on text-independent speaker verification. EURASIP J Appl Signal Proc 2004:430–451
5. Campbell J Jr (1997) Speaker recognition: A tutorial. Proc IEEE 85(9):1437–1462
6. Campbell JP (1995) Testing with the yoho cd-rom voice verification corpus. In: 1995 international conference on acoustics, speech, and signal processing, 1995. ICASSP-95, vol 1. IEEE, pp 341–344
7. Campbell WM, Sturim DE, Reynolds DA (2006) Support vector machines using gmm supervectors for speaker verification. Signal Proc Lett IEEE 13(5):308–311
8. Das RK, Jelil S, Prasanna SM (2016) Significance of constraining text in limited data text-independent speaker verification. In: 2016 international conference on signal processing and communications (SPCOM). IEEE, pp 1–5
9. (2001) Database for indian languages, Speech and vision lab, IIT Madras, Chennai
10. Dehak N, Dehak R, Glass J, Reynolds D, Kenny P (2010) Cosine similarity scoring without score norMalization techniques. In: Proceedings Odyssey speaker and language recognition workshop
11. Eatock S, Mason J (1994) A quantitative assesment of the relative speaker discriminating properties of phonemes. In: Proceedings of the ICASSP 1994, pp 133–136
12. Fienberg SE (1970) An iterative procedure for estimation in contingency tables. Annals of Mathematical Statistics 41(3):907–917
13. Fukunaga K (1990) Introduction to statistical pattern recognition. Academic Press Professional
14. Garofolo JS (1993) Timit acoustic-phonetic continuous speech corpus. Linguistic Data Consortium, Philadelphia
15. Hatch AO, Kajarekar SS, Stolcke A (2006) Within-class covariance normalization for svm-based speaker recognition. In: INTERSPEECH, pp 1471–1474
16. Hosom J-P, Vermeulen PJ, Shaw J (2016) Speaker verification and identification using artificial neural network-based sub-phonetic unit discrimination, uS Patent 9,230,550
17. Hultzen I, Jr JA, Miron M (1964) Tables of transitional frequencies of english phonemes. University of Illinois Press, Urbana, Il
18. Jirouek R, Peuil S (1995) On the effective implementation of the iterative proportional fitting procedure. Comput Stat Data Anal 19(2):177–189
19. Kenny P (2005) Joint factor analysis of speaker and session variability: Theory and algorithms. CRIM Montreal (Report) CRIM 06:8–13
20. Kenny P, Stafylakis T, Alam J, Kockmann M (2015) An i-vector backend for speaker verification. In: Proceedings interspeech, pp 2307–2310
21. Kinnunen T, Hautamäki V, Fränti P (2004) Fusion of spectral feature sets for accurate speaker identification. In: 9th conference speech and computer
22. Kinnunen T, Kärkkäinen I, Fränti P Report series a, the mystery of cohort selection
23. Kucera H, Francis WN (1967) Computational analysis of present day american english. Brown University Press
24. Larcher A, Bousquet P, Lee K. A, Matrouf D, Li H, Bonastre J-F (2012) I-vectors in the context of phonetically-constrained short utterances for speaker verification. In: 2012 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 4773–4776
25. Lei Y, Scheffer N, Ferrer L, McLaren M (2014) A novel scheme for speaker recognition using a phonetically-aware deep neural network. In: 2014 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 1695–1699
26. Martin A, Doddington G, Kamm T, Ordowski M, Przybocki M (1997) The det curve in assessment of detection task performance. In: Proceedings eurospeech, vol 97, pp 1895–1898
27. Matějka P, Glembek O, Castaldo F, Alam MJ, Plchot O, Kenny P, Burget L, Černocky J (2011) Full-covariance ubm and heavy-tailed plda in i-vector speaker verification. In: 2011 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 4828–4831
28. Nagineni S, Hegde R (2010) On line client-wise cohort set selection for speaker verification using iterative normalization of confusion matrices. In: Proceedings eursipco, pp 576–580
29. Najim D, Patrick K, Réda D, Pierre D, Pierre O (2011) Front-end factor analysis for speaker verification. IEEE Trans Audio Speech Lang Process 19(4):788–798
30. Ramos-Castro D, Fierrez-Aguilar J, Gonzalez-Rodriguez J, Ortega-Garcia J (2007) Speaker verification using speaker-and test-dependent fast score normalization. Pattern Recogn Lett 28(1):90–98
31. Reynolds DA (1995) Speaker identification and verification using gaussian mixture speaker models. Speech Comm 17(1–2):91–108

32. Reynolds DA (1997) Comparison of background normalization methods for text-independent speaker verification. In: Eurospeech
33. Reynolds DA, Campbell WM (2008) Text-independent speaker recognition. In: Springer handbook of speech processing. Springer, pp 763–782
34. Rosenberg AE (1976) Automatic speaker verification: A review. Proc IEEE 64(4):475–487
35. Sturim DE, Reynolds DA (2005) Speaker adaptive cohort selection for tnorm in text-independent speaker verification. In: ICASSP, pp 741–744
36. Vincent E, Watanabe S, Nugraha AA, Barker J, Marxer R An analysis of environment, microphone and data simulation mismatches in robust speech recognition. Computer Speech & Language
37. Young SJ, Young S (1993) The HTK hidden Markov model toolkit: Design and philosophy. University of Cambridge Department of Engineering
38. Zeinali H, Sameti H, Burget L, Černockỳ J, Maghsoodi N, Matějka P (2016) i-vector/hmm based text-dependent speaker verification system for reddots challenge. Interspeech 2016:440–444

Waquar Ahmad received his B.Tech. (2005) in Electronics Engineering from Aligarh Muslim University, Aligarh, and M.Tech. (2007) & PHD (2015) from Indian Institute of Technology Kanpur. His research interest include speaker verification, speech processing, and Biometrics.

Harish Karnick is a Professor in the department of Computer Science and Engineering at IIT Kanpur. His interests are machine learning, cognition and more generally artificial intelligence.

Rajesh M. Hegde is a Professor and P K Kelkar Research Fellow with the Dept of Electrical Engineering at IIT Kanpur. His current areas of research interest include multi media signal processing, multi microphone speech processing, pervasive multi media computing, ICT for socially relevant applications in the Indian context, and applications of signal processing in wireless networks with specific focus on emergency response and transportation applications. He has also worked on NSF funded projects on ICT and mobile applications at the University of California San Diego, USA, where he was a researcher and lecturer in the Dept of Electrical and Computer Engineering between 2005-2008. He is also a member of the National working group of ITU-T (NWG-16) on developing multi media applications. Additional biographic information can be found at the URL : http://home.iitk.ac.in/rhegde.