PAN ARCHIVES OF ACOUSTICS
Vol. 42, No. 1, pp. 127–135 (2017)

DOI: 10.1515/aoa-2017-0014

Speaker Model Clustering to Construct Background Models for Speaker Verification

Gökay DİŞKEN[(1)], Zekeriya TÜFEKCİ[(2)], Ulus ÇEVİK[(3)]

[(1)] *Department of Electrical-Electronics Engineering*
Adana Science and Technology University
Adana, Turkey; e-mail: gdisken@adanabtu.edu.tr

[(2)] *Department of Computer Engineering*
Çukurova University
Adana, Turkey

[(3)] *Department of Electrical-Electronics Engineering*
Çukurova University
Adana, Turkey

(*received October 6, 2016; accepted December 27, 2016*)

Conventional speaker recognition systems use the Universal Background Model (UBM) as an imposter for all speakers. In this paper, speaker models are clustered to obtain better imposter model representations for speaker verification purpose. First, a UBM is trained, and speaker models are adapted from the UBM. Then, the k-means algorithm with the Euclidean distance measure is applied to the speaker models. The speakers are divided into two, three, four, and five clusters. The resulting cluster centers are used as background models of their respective speakers. Experiments showed that the proposed method consistently produced lower Equal Error Rates (EER) than the conventional UBM approach for 3, 10, and 30 seconds long test utterances, and also for channel mismatch conditions. The proposed method is also compared with the i-vector approach. The three-cluster model achieved the best performance with a 12.4% relative EER reduction in average, compared to the i-vector method. Statistical significance of the results are also given.

Keywords: Gaussian mixture models; k-means; imposter models; speaker clustering; speaker verification.

1. Introduction

Automatic speaker recognition is a process where a machine is used to verify, or identify a person's identity from his/her voice (Campbell, 1997). In the verification, decision is made by using two models: one represents the claimed speaker, and the other represents the imposters. In the identification, models of all enrolled speakers in a system are evaluated to find the identity of an unknown speaker. A speaker recognition system can be text-dependent, or text-independent. In the text-dependent, the speaker is limited in phonetic sense (a fixed sentence, prompted digits etc.) (Bimbot *et al.*, 2004). In the text-independent systems, a speaker can talk to the system without a constraint. Text-independent recognition is the more challenging one, since training and test speeches for a speaker may have different phonetic contents (Kinnunen, Li, 2010).

Gaussian Mixture Models (GMM) are extensively used for modeling the feature distributions of speakers in text-independent systems (Reynolds, Rose, 1995). They have become the fundamental tool, because of their ability to model arbitrary shapes with a good accuracy (Reynolds, Rose, 1995; Reynolds, 1995; 1997). The Universal Background Model (UBM) approach, introduced in (Reynolds *et al.*, 2000), further improved the popularity of the GMMs for speaker recognition systems. The UBM consists of many Gaussian components (usually 512 to 2048) to represent the acoustic space of all available speakers. The GMM-UBM framework provides the opportunity to adapt speaker models from the UBM with a little adaptation data, and almost halves the scoring duration by invoking the connection between the UBM and the adapted speaker models. Moreover, a UBM model is needed to extract sufficient statistics for state-of-the-art speaker recognition systems using Support Vector Machines

(SVM) (Campbell *et al.*, 2006), joint factor analysis (Kenny *et al.*, 2007; Kenny, 2005), and *i*-vectors (Dehak *et al.*, 2011; Richardson *et al.*, 2015).

On the other hand, the identification process can be very time consuming, especially for a system with a large population. Therefore, many methods have been proposed to achieve speed-ups. One of these methods is the GMM hashing (Auckenthaler, Mason, 2001; McClanahan, De Leon, 2012; McClanahan, De Leon, 2015), where top scoring mixtures for a feature vector can be predicted by using a GMM that is smaller than the UBM. Another method is hierarchically clustering the UBM mixtures (Xiang, Berger, 2003; Saeidi *et al.*, 2010). Some of the other methods are speaker clustering at feature level (Xiong *et al.*, 2006), and speaker clustering at model level (Beigi *et al.*, 1999; De Leon, Apsingekar, 2007; Apsingekar, De Leon, 2009). However, there is a tradeoff between the identification rate and identification time, since not all the mixtures are scored, or not all the speakers' models are considered. Speaker clustering method is also used to compensate speaker-related effects in speech recognition recently (Hossa, Makowski, 2016).

In this paper, it is suggested that the verification systems may also benefit from the speaker model clustering methods. The main idea of the model clustering in the identification is to reduce the number of candidate speakers. Since only the claimed ID's model, and an imposter model is taken into account for verification, the goal of this paper is to create more accurate imposter models by clustering the speaker models. For this purpose, first the speaker models are obtained by adapting the means of a UBM. Then, speaker models are represented as mean supervectors, and k-means algorithm is applied to cluster them. The cluster centers are used as the imposter model of the respective clusters. The main advantage of this method is to acquire better imposter models, and decreasing the false positive rates of speaker verification systems. Also, the proposed algorithm shows comparable, or better, results compared to the *i*-vector approach, without demanding an excessive training procedure such as *i*-vectors. This paper is organized as follows: In Sec. 2, the traditional GMM-UBM method is revised, and then the proposed approach is introduced. The experimental results are given in Sec. 3, the results are discussed in Sec. 4, and Sec. 5 concludes the paper.

2. Conventional and proposed methods

In this section, first the conventional approach for modeling the speaker feature distributions is given according to (Reynolds *et al.*, 2000), since the proposed model clustering algorithm is based on the GMM-UBM method. Then, the proposed approach for clustering the speaker models is given. Also, differences between the proposed method and some of the other model clustering approaches for speaker recognition are mentioned.

2.1. GMM-UBM speaker modeling

A GMM is defined by its mixture parameters, which consists of mixture weights, mean vectors, and covariance matrices. An M-mixture model can be written as

$$\lambda = \{p_i, \overline{\mu}_i, \Sigma_i\}, \qquad i = 1, \ldots, M, \tag{1}$$

where λ is the GMM model, i is the mixture index, p_i is the weight, $\overline{\mu}_i$ is the mean vector, and Σ_i is the covariance matrix (usually diagonal) of mixture i, respectively. The weights in a GMM model must sum to one. These parameters are learnt from the training data by using the expectation maximization algorithm.

A UBM is intended to represent the acoustic space of all available speakers, so it is speaker-independent. In some situations, a speaker's data may not be sufficient for training his/her own GMM. The speaker's model can be adapted from a well-trained UBM, by using the available speaker-dependent data. Sufficient statistics, obtained from the speaker-dependent training data, are used to update the UBM parameters for mixture i, hence an adapted model is created. All mixture parameters (weights, means, and variances) can be adapted, but adapting only the means is found to be more effective (Reynolds *et al.*, 2000). Therefore, the main approach in the speaker recognition literature is to adapt only the means, and use the same weights and variances of the UBM components in the speaker model. In the verification phase, a feature vector from an unknown speaker is scored with the UBM model first. Then, indexes of top scoring N mixtures (where N is much smaller than the number of all mixtures) are extracted for the given feature vector, and speaker model's score is calculated with only these N mixtures, instead of scoring all mixtures. In the decision stage, if the difference between these scores exceeds a threshold, the unknown speaker is verified as whom he/she claims to be, otherwise, as an imposter. This process is illustrated in Fig. 1.

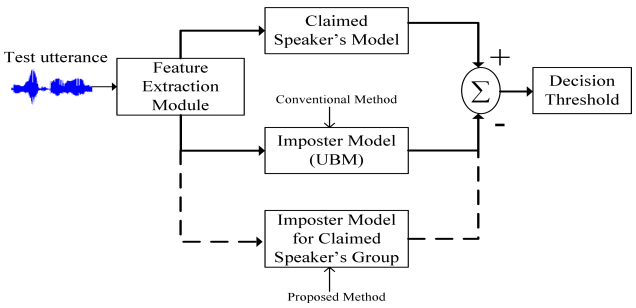

Fig. 1. Scoring algorithms of the conventional GMM-UBM (solid line), and the proposed cluster based method (dashed line).

2.2. Clustering speaker models

In the proposed method, a UBM is trained, and speaker models are adapted (means only) by following the conventional procedure. After models for all enrolled speakers are obtained, their means are divided (element-wise) by their respective standard deviations to achieve normalization (Eq. (2))

$$\widetilde{\mu}_{i,s} = \frac{\boldsymbol{\mu}_{i,s}}{\boldsymbol{\sigma}_i}, \tag{2}$$

where i is the mixture index, s is the speaker index, $\widetilde{\mu}_{i,s}$ is the normalized mean vector of the i-th mixture for speaker s, and $\boldsymbol{\sigma}_i$ is the standard deviation vector of the i-th mixture.

The normalized means for each mixture are concatenated to construct a mean supervector per speaker. Then, these supervectors are clustered by using the traditional k-means algorithm with the Euclidean distance as the similarity measure (Eq. (3))

$$J_{s,c} = \sum_{s=1}^{S} \|\widetilde{\mu}_s - \mathbf{v}_c\|^2, \tag{3}$$

where c is the cluster index, $\mathbf{v}_c$ is the vector representing the center of cluster c, $\widetilde{\mu}_s$ is the mean supervector of speaker s, S is the total number of speakers, and $J_{s,c}$ represents the distance of speaker s to cluster c. Each speaker is assigned to the cluster which gives the minimum $J_{s,c}$ value. Then, the cluster centers are recalculated by using Eq. (4)

$$\mathbf{v}_c = \left(\frac{1}{N_c}\right) \sum_{s=1}^{N_c} \widetilde{\mu}_{s,c}, \tag{4}$$

where Nc is the number of speakers assigned to the cluster c, and $\widetilde{\mu}_{s,c}$ is the mean supervector of speakers assigned to the same cluster.

Final values of cluster centers ($\mathbf{v}_c$ supervectors) are decomposed into mixture mean vectors, and multiplied (element-wise) by the standard deviation vector of their respective components (Eq. (5)). Hence, the imposter models for each cluster are created. Note that the mixture weights, and variances are the same as the speaker models. This process is illustrated in Fig. 2

$$\boldsymbol{\mu}_{i,c} = \mathbf{v}_{i,c}\boldsymbol{\sigma}_i, \tag{5}$$

where $\mathbf{v}_{i,c}$ is the i-th mixture's mean vector of cluster c, and $\boldsymbol{\mu}_{i,c}$ is the final (denormalized) values for the i-th mixture's mean vector of cluster c. With this approach, speakers sharing the similar acoustic space are assumed to be gathered in the same group by clustering their models, and this space will be the imposter model for the speakers in that group, as indicated by the dashed lines in Fig. 1.

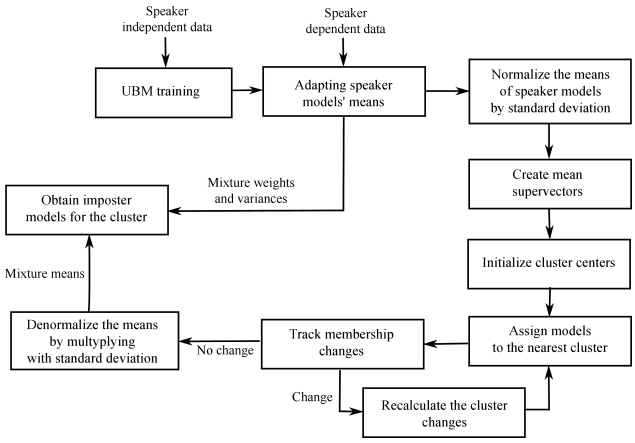

Fig. 2. Block diagram of the proposed method.

Similar clustering methods are proposed in (Apsingekar, De Leon, 2009; De Leon, Apsingekar, 2007) for speaker identification, where the Euclidean distances are calculated with weighted mean vectors, and covariance normalized weighted mean vectors. However, since the speaker models in our system share the same weights (copied from the UBM), the distance is calculated solely between the model mean vectors.

2.3. Comparison with cohort modeling approach

The proposed algorithm is a combination of the UBM, and cohort methods, from a point of view. The cohort model represents the acoustic space around a target speaker by combining the closest speakers to the target. A cohort model for each individual speaker is constructed, which is a drawback in the means of storage, and fair scoring (a speaker's cohort model may not accurately define the acoustic space around him/her). Although the UBM method is reported to perform better than the cohort approach in (Reynolds, 1997), and extensively used for the reasons mentioned before, cohort imposter models are still investigated by researchers (Zhu *et al.*, 2011; McLaren *et al.*, 2010). Combining the UBM, and the cohort is considered before in the score space by employing SVM to find an optimum decision value (Brew, Cunningham, 2009; 2010). However, conventional GMM-UBM scoring process is preferred in this paper, since it is hypothesized that by increasing the performance of the traditional method, it is also possible to achieve a higher performance with more complex methods that are based on the traditional UBM (such as SVM machines with mean supervectors, i-vectors, etc.).

Comparing to the conventional UBM method, the imposter models now represent not all the acoustic space, but the acoustic space defined by the speakers in the cluster. A cohort-like representation is achieved

by this modeling approach. Furthermore, there is no need to construct an imposter model for each speaker, since an imposter model is shared between the speakers in a group. Therefore, the computational and memory loads are also reduced in the proposed method.

3. Experiments and results

The NIST SRE 1998 database (available at: www.nist.gov/speech/tests/spk/1998/current_plan.htm, access date: 05.10.2016) was used to evaluate the performance of the proposed method. The database contains 250 male speakers and 250 female speakers. In (Doddington *et al.*, 2000), it is reported that the difference in the system performance for the male, and the female speakers is fairly small (Fig. 9, DET curves named Fem (All), and Male (All) shows the performances of female, and male speakers, respectively). Also, the NIST evaluations do not include cross-gender tests (Doddington *et al.*, 2000). However, one can create a gender-independent system by separately training male, and female models, then combining them (Reynolds *et al.*, 2000). Therefore, only the male speakers were used in the experiments. In the database, one-session, two-session, and two-session-full training conditions are available, as well. In the experiments, the two-session-full condition was preferred, where there were five training files, each consisted 1 minute of speech taken from phone conversations.

Two handset types are available in this database: electret and carbon-button. So, a same-handset condition means that the training and test segments for a speaker are both collected by using the electret, or both carbon-button. A different-handset condition indicates that the training segments for a speaker are collected via the electret type, and test segments for the same speaker are collected with the carbon-button type, or vice-versa. Therefore, the performance of the proposed method under channel mismatch conditions, which is one of the main sources for performance degradation, was also tested.

For the tests, speech segments with 3, 10, and 30 seconds durations were used. For each of these test durations, there were 1308 speech files collected from the same-handset type, and 1192 speech files collected from a different-handset type. For each test file, there was one trial for the target speaker, and nine trials for the non-target speakers. The total number of trials in each test data duration was 13080 for the same-handset, and 11920 for the different-handset conditions.

The main metric used for the performance comparison is the Equal Error Rate (EER), which is widespreadly used in the speaker recognition literature. EER value defines a threshold where the false acceptance rate and the false rejection rate of a system is equal. Also, a detection cost function (DCF) is defined as given in Eq. (6)

$$\mathrm{DCF} = C_{FA} P_{FP|N} P_N + C_{FR} P_{FN|T} P_T, \qquad (6)$$

where $P_{FP|N}$ is the false positive rate (FPR), $P_{FN|T}$ is the false negative rate (FNR), the cost of the false acceptance is $C_{FA} = 10$, the cost of the false rejection is $C_{FR} = 1$, the *a priori* probability of target tests is $P_T = 0.1$, and the *a priori* probability of nontarget tests is $P_N = 0.9$. The minimum of the DCF is also calculated as another performance metric. The detection error tradeoff (DET) curves are given for the baseline systems, and the best performing clusters of the proposed method. Note that the other clusters' curves are not shown to avoid confusing illustrations, since the curves highly interfere with each other.

The HTK Toolkit (Young *et al.*, 2000) was used to extract Mel-Frequency Cepstral Coefficients (MFCCs) from the training, and test data. A Hamming window with a 25 ms length, and a 10 ms shift was employed. 26 triangular bandpass filters were used in the filter bank. First twelve of the MFCCs, excluding the zeroth coefficient, were selected, and the normalized log-energy was appended. The Cepstral mean subtraction was applied to reduce the convolutive channel effects. Adding the delta features, the final 26 features were obtained. The other processes (Training a UBM model, adapting speaker models, model clustering, and scoring) were implemented using the C++ programming language.

A UBM with 1024 Gaussian components was trained by pooling the available training data. Then, the speaker models were adapted with the speaker-dependent training data, with a relevance factor of 16. Top scoring 5 mixtures were selected in the UBM for each test feature vector. Resulted EER (minDCF) values for this baseline GMM-UBM method are given in the third row of Table 1. The EER (minDCF) value increased as the duration of test data decreased, as expected. In addition, the channel mismatch dramatically decreased the performance.

2, 3, 4, and 5 clusters were considered to check the validity of the proposed method. The speaker models adapted from the baseline UBM were used. Their results are given in the last four rows of Table 1. The best improvements were obtained by three-cluster except the 3-seconds cases. 10.14%, and 10.64% relative EER reductions were achieved for the 10-seconds, and 30-seconds durations under same-handset condition, respectively. For the different-handset tests, 8.67% and 13.02% relative EER reductions were achieved for the 10-seconds, and 30-seconds durations, respectively. The last column of Table 1 shows the average EER (minDCF) reductions for each cluster, compared to the baseline UBM method. The highest performance improvement was achieved by three-cluster case, as shown in Table 1. Also, it should be noted that the

Table 1. EER values obtained for conventional and proposed methods. minDCF values are given in parenthesis.

Duration	Same-handset condition			Different-handset condition			Average Reduction
	3-s	10-s	30-s	3-s	10-s	30-s	
UBM	11.2385 (0.2054)	5.2752 (0.0989)	3.5933 (0.0621)	25.5872 (0.4766)	20.3020 (0.3827)	16.1074 (0.3041)	
2-Cluster	**10.2446** (0.1904)	4.893 (0.0902)	3.211 (0.06)	25.4195 (0.4804)	19.9664 (0.3782)	16.1074 (0.3045)	4.84% (3.29%)
3-Cluster	10.5505 (*0.1903*)	**4.7401** (*0.0885*)	**3.211** (*0.057*)	24.1611 (*0.4443*)	**18.5403** (*0.3494*)	**14.0101** (*0.2627*)	**9.03%** (*9.03%*)
4-Cluster	10.3976 (0.1928)	4.9694 (0.09)	3.2875 (0.0615)	24.0772 (0.4465)	18.7919 (0.3498)	14.1779 (0.2676)	7.85% (7.17%)
5-Cluster	11.0092 (0.2001)	4.9694 (0.0925)	3.3639 (0.0611)	**23.9933** (0.4522)	19.1275 (0.3576)	14.1779 (0.2678)	6.37% (5.71%)

three-clusters always yielded the best minDCF performance. The results proved that the imposter models created by clustering the speaker models were more suitable than the UBM model. Figures 3 and 4 shows the DET curves for the same-handset, and different-handset conditions, respectively.

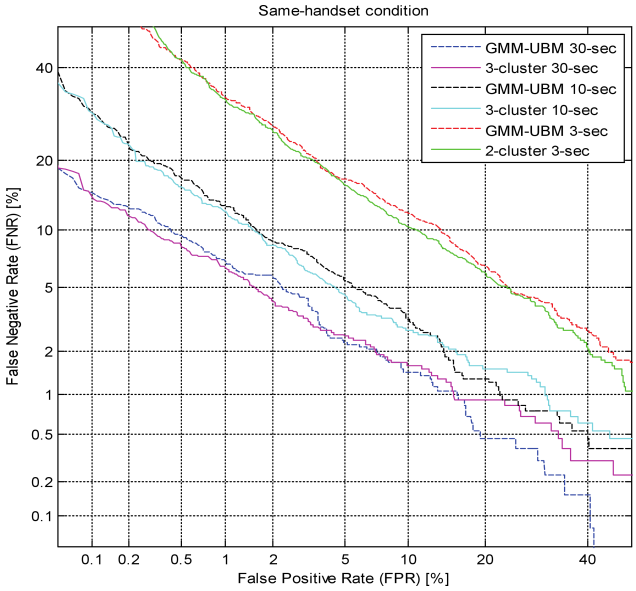

Fig. 3. DET curves of the baseline GMM-UBM, and the best performing clusters for the same-handset condition, and for each data duration.

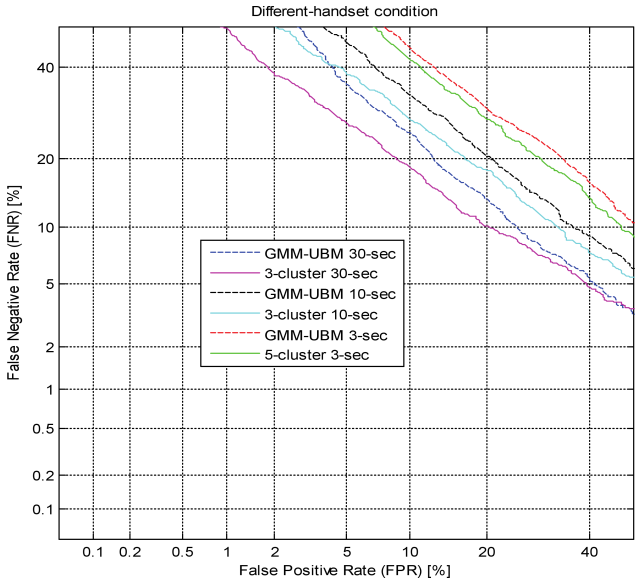

Fig. 4. DET curves of the baseline GMM-UBM, and the best performing clusters for the different-handset condition, and for each data duration.

Re-adapting the speaker models from the cluster imposture models were also examined. This method can be considered as the subsets of the baseline GMM-UBM method. The results for this approach is given in Table 2. As in the previous case, three-cluster gave the best overall performance improvement in terms

Table 2. EER values for speaker models adapted from their respective cluster imposter models.

Duration	Same-handset condition			Different-handset condition			Average Reduction
	3-s	10-s	30-s	3-s	10-s	30-s	
2-Cluster	10.3976 (0.1901)	4.8930 (0.089)	**3.1346** (0.0581)	25.3356 (0.4773)	20.1342 (0.3799)	16.1074 (0.3036)	4.88% (4.11%)
3-Cluster	**10.2446** (*0.1882*)	4.8930 (*0.084*)	3.211 (*0.0568*)	**24.245** (*0.4567*)	19.4631 (*0.3635*)	15.2685 (0.2855)	6.88% (*7.88%*)
4-Cluster	10.3211 (0.1901)	4.8930 (0.0869)	**3.1346** (0.0578)	24.4128 (0.459)	19.7148 (0.3681)	15.1007 (0.2834)	**6.98%** (6.80%)
5-Cluster	10.6269 (0.1953)	4.8930 (0.0869)	3.2875 (0.0584)	24.8322 (0.4648)	**19.2953** (0.3659)	**14.9329** (*0.2824*)	6.06% (6.17%)

of minDCF. Improvements over the baseline can be still observed, but this method is less effective than the traditionally adapted speaker models under mismatched channel condition.

Recently, *i*-vector approach (Dehak *et al.*, 2011) has become the state of the art method for speaker recognition. In this method, a total variability matrix is trained with the aid of a UBM model. Then, by using this matrix, an utterance can be represented with a fixed low dimensional vector (*i*-vector). This representation also gives the opportunity to use various algorithms for reducing the channel effects such as linear discriminant analysis (LDA), nuisance attribute projection, and within class covariance normalization, etc. (Dehak *et al.*, 2011). To compare the proposed method with the *i*-vector technique, the baseline UBM and the pooled training data were used to train the total variability matrix in twenty iterations, then 100 dimensional *i*-vectors were extracted from each utterance. LDA was used to reduce the channel mismatch effects, and probabilistic LDA was employed for scoring the *i*-vectors. MSR Identity Toolbox (Sadjadi *et al.*, 2013) was used in the *i*-vector extraction, and scoring processes. As the *i*-vector approach includes the LDA for channel mismatch compensation, in order to make a fair comparison, handset normalization (Reynolds, 1997) was added to the proposed method. Handset normalization is a score normalization technique to reduce the channel mismatch effects, and applying this method to the clusters may be beneficial.

In Table 3, test results with handset normalization are given for the proposed method. The relative improvements compared to the *i*-vector with the LDA approach are given in the last column of the table. The results showed that the prosed method achieved a superior performance than the *i*-vector approach in 3-seconds same-handset condition tests. A relative improvement of 20.8% was achieved by using two-clusters. The proposed method consistently gave better performances, yielding average reductions higher than 10% in terms of EER, as seen in the last column. For the 30-seconds same-handset condition tests, modest improvements can be seen in EER values. Hence, the results indicated that comparable or better performances can be achieved by the proposed clustering method, without the excessive training procedure that the *i*-vector approach demands. It is also important to emphasize that, without the handset normalization, the proposed method still shows comparable results with the *i*-vector approach, which can be examined by comparing the results from Table 1, or Table 2, with the results of the *i*-vector in Table 3. Figures 5 and 6 show the DET curves for the same-handset, and the different-handset conditions, respectively.

The statistical significance of the results is also examined with the McNemar's test, which is also used in speech recognition area (Pallet *et al.*, 1990; Gillick, Sox, 1989). Consider two classifiers (named A, and B) are tested with a test data, and the following variables are counted.

N00: number of examples misclassified by both A, and B,

N01: number of examples misclassified by A, but not B,

N10: number of examples misclassified by B, but not A,

N11: number of examples misclassified by neither A, nor B.

The null hypothesis expects that the two algorithms have the same error rate (N01 = N10). Continuity corrected McNemar test is given in Eq. (7).

$$X^2 = \frac{(|N01 - N10| - 1)^2}{N01 + N10}. \tag{7}$$

Under the null hypothesis, X^2 has a chi-square distribution with 1 degree of freedom. The value of test at 5% significance level for 1 degree of freedom is 3.84. Hence, if the test is greater than this value, the null hypothesis is rejected, which indicates the two classifiers have different performances. In Table 4, the proposed method is compared with the *i*-vector, based on the EER values given in Table 3. The results indicate that

Table 3. EER values for *i*-vector approach and the proposed method with handset normalization.

Duration	Same-handset condition			Different-handset condition			Average Reduction
	3-s	10-s	30-s	3-s	10-s	30-s	
i-vector	10.7034 (0.1957)	4.893 (0.0868)	3.211 (*0.053*)	24.3289 (0.458)	18.1208 (0.343)	14.7651 (0.2729)	
2-Cluster	**8.4098** (*0.1544*)	4.3578 (0.0817)	2.9817 (0.0562)	22.0638 (*0.4039*)	16.443 (0.3083)	13.5906 (0.2564)	11% (8.15%)
3-Cluster	8.7920 (0.1672)	**4.1284** (*0.0765*)	**2.9052** (0.0547)	21.896 (0.4138)	**16.1074** (0.305)	13.255 (0.2463)	**12.4%** (*8.95%*)
4-Cluster	8.8685 (0.1654)	4.2813 (0.0794)	2.9817 (0.0541)	21.9799 (0.4128)	16.1913 (0.3046)	13.0872 (0.2454)	11.4% (8.45%)
5-Cluster	8.945 (0.1665)	4.2049 (0.0781)	3.211 (0.0561)	**21.7282** (0.4086)	**16.1074** (*0.3013*)	**12.7517** (*0.2411*)	10.98% (8.94%)

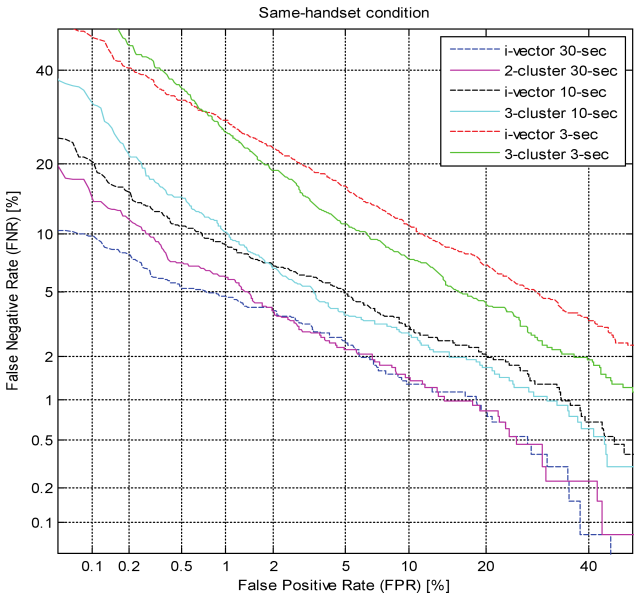

Fig. 5. DET curves of the baseline i-vector, and the best performing clusters for the same-handset condition, and for each data duration.

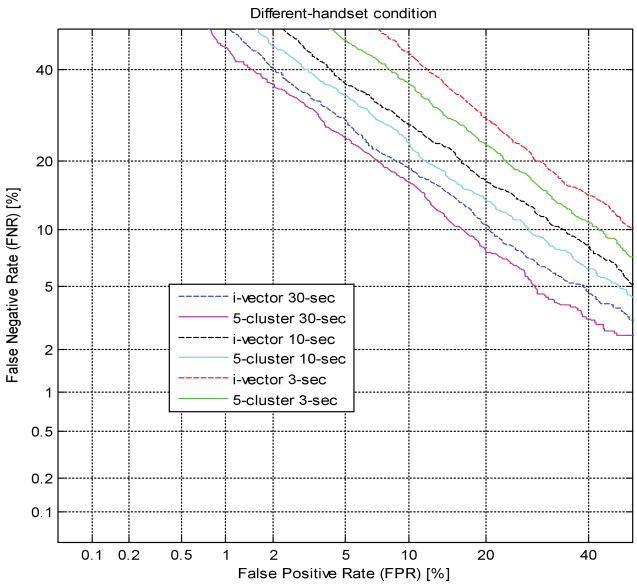

Fig. 6. DET curves of the baseline i-vector, and the best performing clusters for the different-handset condition, and for each data duration.

Table 4. X^2 values obtained by using the proposed classifiers and the i-vector classifier.

Duration	Same-handset condition			Different-handset condition		
	3-s	10-s	30-s	3-s	10-s	30-s
2-Cluster	54.98	7.42	1.5	24.44	17.73	9.07
3-Cluster	36.16	14.6	2.18	26.78	25.74	16.2
4-Cluster	34.41	10	1.12	25.41	23.9	20.33
5-Cluster	31.33	11.84	0.0017	30.82	25.58	28.83

there are significant performance differences between methods, except the 30-second case. The highest differences are found in the 3-second tests, which strongly supports that the proposed method is more suitable for short duration utterances.

4. Discussion

The experiments proved that the proposed clustering method showed improvements over the conventional GMM-UBM, and the i-vector methods. This should be due to a better estimation of the imposter models, as expected from the proposed algorithm. The DET curves supports this idea, since for a given FNR, the proposed methods produce lower FPRs, especially in the short utterance tests (3-seconds, 10-seconds), and the different-handset condition. This property makes the proposed method much more suitable for practical applications. As an example, consider a system which verifies the speakers over phone calls. A speaker enrolled in the system may use different phones at different times, which results in a handset mismatch. Also, the speakers probably want to be verified with a few words, or phrases, so a short verification time is favorable.

The statistical significance test results given in Table 4 implies that the results found in the experiments are not by coincidence. The only similarity occurred in the 30-seconds same-handset condition tests. The reason behind this situation is that better i-vector representations are acquired as the utterance duration increases, hence the classifier performance also increases. The duration mismatch in i-vectors is another research problem, which is out of the scope of this paper.

The three-cluster showed the best performance in general (based on average reductions given in Table 1, and Table 3). On the other hand, performances of the clusters are close to each other. It is a kind of expected situation, because only the speaker model means are considered for classification. As discussed before, if there is no speaker data related to a mixture, the speaker's model use the mean of the respective mixture of the UBM. This similarity effects the cluster performance. Another option to be tested in the future is to adapt both means, variances, and weights to obtain speaker models, and including them in the clustering algorithm. Also, it should be noted that the clustering for the proposed method is made by the k-means algorithm. A different clustering method may yield more accurate imposter models, therefore, increase the performance, and the statistical significance of the system.

5. Conclusion

In this work, speaker models are clustered to improve the speaker verification performance. Conventional methods use a UBM model, and speaker models are derived by adapting the mixture means according to available speaker-dependent data. In the proposed

algorithm, the adapted speaker models are clustered by using the k-means algorithm with the Euclidean distance criteria to create cluster dependent imposter models. It is shown that the imposter models constructed by this approach produced superior results than the traditional GMM-UBM method consistently, for different test data durations, and under channel match, or mismatch conditions.

The i-vector approach, which has become the state-of-the-art method for speaker verification, is also considered in the experiments. LDA was applied to i-vectors for compensating channel mismatch effects. To make a fair comparison, a handset score normalization was applied to the proposed clustering method to reduce mismatch degradations. On average, the three-cluster yielded a 12.14% relative EER reduction, which is the best. The DET curves showed that the proposed method produces lower FPRs, especially in the different-handset condition, and short test utterance durations.

The experiments indicated that the proposed method showed better verification performances than the conventional UBM and i-vector approaches. Future research directions are exploring the effects of different distance measures, different clustering methods, and extract sufficient statistics for each cluster from their respective imposter models (instead of the UBM model) for i-vectors.

References

1. Apsingekar V.R., De Leon P.L. (2009), *Speaker Model Clustering for Efficient Speaker Identification in Large Population Applications*, IEEE Trans. Audio. Speech. Lang. Processing, **17**, 848–853.
2. Auckenthaler R., Mason J.S. (2001), *Gaussian selection applied to text-independent speaker verification*, Proc. Speaker Odyssey: The Speaker Recognition Workshop, 83–88, Greece.
3. Beigi H.S.M., Maes S.H., Chaudhari U.V., Sorensen S. (1999), *A hierarchical approach to large-scale speaker recognition*, European Conference on Speech Communication and Technology, 2203–2206, Hungary.
4. Bimbot F., Bonastre J.-F., Fredouille C., Gravier G., Magrin-Chagnolleau I., Meignier S., Merlin T., Ortega-Garcia J., Petrovska-Delacretaz D., Reynolds D.A. (2004), *A Tutorial on Text-Independent Speaker Verification*, EURASIP J. Adv. Signal Process., **2004**, 430–451.
5. Brew A., Cunningham P. (2009), *Combining Cohort and UBM Models in Open Set Speaker Identification*, Seventh International Workshop on Content-Based Multimedia Indexing, 62–67, Crete.
6. Brew A., Cunningham P. (2010), *Combining cohort and UBM models in open set speaker detection*, Multimed. Tools Appl., **48**, 141–159.
7. Campbell J.P. (1997), *Speaker recognition: a tutorial*, Proc. IEEE, **85**, 1437–1462.
8. Campbell W.M., Sturim D.E., Reynolds D.A., Solomonoff A. (2006), *SVM Based Speaker Verification using a GMM Supervector Kernel and NAP Variability Compensation*, IEEE International Conference on Acoustics Speed and Signal Processing Proceedings, I-97-100, France.
9. De Leon P.L., Apsingekar V. (2007), *Reducing Speaker Model Search Space in Speaker Identification*, Biometrics Symposium, 1–6, USA.
10. Dehak N., Kenny P.J., Dehak R., Dumouchel P., Ouellet P. (2011), *Front-End Factor Analysis for Speaker Verification*, IEEE Trans. Audio. Speech. Lang. Processing, **19**, 788–798.
11. Doddington G., Przybocki M., Martin A., Reynolds D. (2000), *The NIST speaker recognition evaluation – Overview, methodology, systems, results, perspective*, Speech Communication, **31**, 225–254.
12. Gillick L., Cox S. (1989), *Some statistical issues in the comparison of speech recognition algorithms,* International Conference on Acoustics, Speech, and Signal Processing, 532–535.
13. Hossa R., Makowski R. (2016), *An Effective Speaker Clustering Method using UBM and Ultra-Short Training Utterances*, Archives of Acoustics, **41**, 107–118.
14. Kenny P. (2005), *Joint factor analysis of speaker and session variability: Theory and algorithms*, CRIM, Montr. CRIM-06/08-13, 1–17.
15. Kenny P., Boulianne G., Ouellet P., Dumouchel P. (2007), *Joint Factor Analysis Versus Eigenchannels in Speaker Recognition*, IEEE Trans. Audio, Speech Lang. Process., **15**, 1435–1447.
16. Kinnunen T., Li H. (2010), *An overview of text-independent speaker recognition: From features to supervectors*, Speech Communication, **52**, 12–40.
17. McClanahan R.D., De Leon P.L. (2012), *Mixture Component Clustering for Efficient Speaker Verification*, Interspeech, 1086-1090, USA.
18. McClanahan R.D., De Leon P.L. (2015), *Reducing computation in an i-vector speaker recognition system using a tree-structured universal background model*, Speech Communication, **66**, 36–46.
19. McLaren M., Vogt R., Baker B., Sridharan S. (2010), *Data-Driven Background Dataset Selection for SVM-Based Speaker Verification*, IEEE Trans. Audio. Speech. Lang. Processing, **18**, 1496–1506.
20. Pallet D., Fisher W., Fiscus J. (1990), *Tools for the analysis of benchmark speech recognition*, International Conference on Acoustics, Speech, and Signal Processing, 97–100.
21. Reynolds D.A. (1995), *Speaker Identification and Verification using Gaussian mixture speaker odels*, Speech Communication, **17**, 91–108.
22. Reynolds D.A. (1997), *Comparison of Background Normalization Methods for Text-Independent Speaker*

Verification, European Conference on Speech Communication and Technology, Greece.

23. Reynolds D.A., Quatieri T.F., Dunn R.B. (2000), *Speaker Verification Using Adapted Gaussian Mixture Models*, Digital Signal Processing, **10**, 19–41.

24. Reynolds D.A., Rose R.C. (1995), *Robust text-independent speaker identification using Gaussian mixture speaker models*, IEEE Trans. Speech Audio Process., **3** 72–83.

25. Richardson F., Reynolds D., Dehak N. (2015), *Deep Neural Network Approaches to Speaker and Language Recognition*, IEEE Signal Processing Letters, **22**, 1671–1675.

26. Sadjadi S.O., Slaney M., Heck L. (2013), *MSR Identity Toolbox v1.0: A MATLAB Toolbox for Speaker Recognition Research*, Speech and Language Processing Technical Committee Newsletter, IEEE, 1–4.

27. Saeidi R., Kinnunen T., Mohammadi H.R.S., Rodman R., Franti P. (2010), *Joint frame and Gaussian selection for text independent speaker verification*, IEEE International Conference on Acoustics, Speech and Signal Processing, 4530–4533, USA.

28. Xiang B., Berger T. (2003), *Efficient text-independent speaker verification with structural gaussian mixture models and neural network*, IEEE Trans. Speech Audio Process., **11**, 447–456.

29. Xiong Z., Zheng T.F., Song Z., Soong F., Wu W. (2006), *A tree-based kernel selection approach to efficient Gaussian mixture model–universal background model based speaker identification*, Speech Communication, **48**, 1273–1282.

30. Zhu D., Ma B., Li H. (2011), *Speaker Verification With Feature-Space MAPLR Parameters*, IEEE Trans. Audio. Speech. Lang. Processing, **19**, 505–515.