Circuits, Systems, and Signal Processing manuscript No.
(will be inserted by the editor)

Improved i-vector representation for speaker diarization

Yan Xu · Ian McLoughlin · Yan Song · Kui Wu

Received: Feb.2015 / Accepted: Sept. 2015

Abstract This paper proposes using a previously well-trained deep neural network (DNN) to enhance the i-vector representation used for speaker diarization. In effect, we replace the Gaussian Mixture Model (GMM) typically used to train a Universal Background Model (UBM), with a DNN that has been trained using a different large scale dataset. To train the T-matrix we use a supervised UBM obtained from the DNN using filterbank input features to calculate the posterior information, and then MFCC features to train the UBM instead of a traditional unsupervised UBM derived from single features. Next we jointly use DNN and MFCC features to calculate the zeroth and first order Baum-Welch statistics for training an extractor from which we obtain the i-vector. The system will be shown to achieve a significant improvement on the NIST 2008 speaker recognition evaluation (SRE) telephone data task compared to state-of-the-art approaches.

Keywords Speaker diarization · DNN · i-vector

1 Introduction

Speaker diarization is a technology used to solve the problem of "who spoke what and when did they speak" in a multi-party conversation. Speaker segmentation and clustering are two important components of a speaker diarization system. Segmentation detects change points in a recording and then cuts the speech into many smaller segments at these divisions. Ideally each small segment contains speech from just one speaker. Next, speaker clustering gathers together neighbouring segments uttered by

Yan Xu · Yan Song · Kui Wu
National Engineering Laboratory of Speech and Language Information Processing,
The University of Science and Technology of China, Hefei, PRC. E-mail: xuyan12@email.ustc.edu.cn, songy@ustc.edu.cn, wukui@email.ustc.edu.cn
Ian McLoughlin
School of Computing, The University of Kent, Medway Campus, Chatham, UK. E-mail: ivm@ustc.edu.cn

the same speaker. The most popular method for clustering is currently a bottom-up approach known as Agglomerative Hierarchical Clustering (AHC)[17].

Diarization can be applied to several speech areas [7]. The main applications include the transcription of telephone and broadcast meetings, dominant speaker detection and auxiliary video segmentation. With such technology, we can envisage effective management of audio streams, leading to the realisation of structured content at a higher semantic level. Segmentation itself has application in other areas such as speaker verification. Meanwhile, with the advent of reliable speaker diarization methods, we can achieve real-time detection of the number of speakers as well as being able to attribute what each speaker says in a meeting or in a news broadcast.

Although state-of-the-art speaker diarization systems can achieve good results on telephone data, there are still problems with current systems. Previous approaches model each segment with a single GMM model or i-vector extracted from a Universal Background Model (UBM), for example in [18]. This has been shown capable of representing some segments in speaker diarization quite well, but the complexity and hence capability of the model is relatively low and thus it is not always able to represent all of the underlying speech. This is one area that will be addressed in this paper.

In recent years, many alternative diarization algorithms have been proposed, often inspired from related speech research. For example, factor analysis was first applied to speaker diarization by Kenny et. al. [9]. The technique was used with a simple eigenchannel (EC) algorithm for speaker verification [6]. Subsequently, other researchers [18] contributed a two-step clustering method based on Cross Likelihood Ratio (CLR). This was developed to measure the similarity between segments. Used in the second pass, this is an effective solution to the problem of a single Gaussian model describing the complex distribution of the features. It also helps to solve a common problem associated with the Bayesian Information Criterion (BIC) in that the distance metric between clusters is data size dependent [1].

An open research topic is the derivation of a suitable model that can represent short segments, as well as enable a measure of the similarity and difference between neighbouring segments for clustering. State-of-the-art systems represent segments by making use of a Gaussian Mixture Model (GMM) adapted from the Universal Background Model (UBM) to form an i-vector. Such representations, which we denote UBM/i-vector, generally report good results. However we note that deep neural networks (DNN) have been found to perform well in related fields (including speech recognition, language recognition and speaker verification). They appear to be capable of constructing accurate models of speech, even for shorter segments. We thus propose utilising a well-trained DNN to construct a UBM and T-matrix with the aim of the extracted i-vectors being better models of the underlying segments. Additional motivation comes from some promising recent work which combines convolutional neural networks (CNN) with an i-vector representation for language [10] and speaker [12] identification tasks, as well as the use of transfer learning for DNN/i-vector in language identification [16].

In this paper, the Switchboard database will be used to train a DNN [8], using phonetic ground truth data. While the resulting DNN can be very well trained due to the quality and quantity of the training database, it's output conveys phonetic informa-

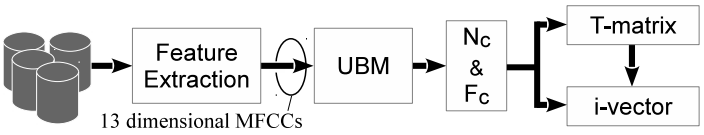

Fig. 1 *The UBM/i-vector baseline system*

tion rather than the speaker-dependent information required for diarization. Thus we will specifically consider the DNN to be a UBM which encodes phone information. Next we model the variance of all outputs in a similar way to a total variability (TV) system [6] and subsequently combine the DNN and TV information into a new representation that we denote DNN/i-vector. The performance of this proposed approach will be evaluated with various system-level parameters against current state-of-the-art UBM/i-vector methods.

The remainder of this paper is organised as follows. In Section 2, we briefly describe the baseline UBM/i-vector technique followed by the proposed DNN/i-vector technique. Section 3 reports results from a number of experiments for different features and dimensions. Finally, Section 4 will conclude the paper.

2 Diarization overview

The baseline diarization system is constructed based on Wu et. al. [18]. The main difference being that we propose replacing the UBM/i-vector extractor with a well-trained DNN/i-vector extractor that has been trained on a phonetic basis using a much larger database. We will describe the method in terms of both structure and training below, after first reviewing the traditional i-vector extractor.

2.1 Traditional UBM/i-vector systems

Fig. 1 shows the structure of a traditional diarization system which trains a UBM, usually based on 13 dimensional MFCC features. Next a T-matrix and hence i-vector are extracted using zero and first order statistics from the UBM, from the same input features [18, 15]. In general, the first step is to use the Linde-Buzo-Gray (LBG) algorithm [17] to extract initial model parameters in a GMM representation. However the Gaussian model parameters derived from the LBG algorithm use hard decisions which can easy fall into local minima, meaning that the final Gaussian model will not be a good match. Therefore the Expectation Maximisation Algorithm (i.e. allowing soft decisions) is applied to adjust the parameters of the model. Given this, the same MFCC features are then used to train the UBM,

$$p(X) = \sum_{i=1}^{c} \lambda_i N(X; M_i, \Sigma_i) \tag{1}$$

where λ_i is the weight of each Gaussian, $N(X; M_i, \Sigma_i)$ represents the Gaussian function and the mean and covariance matrices of the Gaussian function are M_i and

Σ_i. For each mixture component c, we denote the extracted centred zero and first order Baum-Welch statistics as N_c and F_c,

$$N_c = \sum_t \gamma_t(c) \tag{2}$$

$$F_c = \sum_t \gamma_t(c)(X_t - m_c) \tag{3}$$

where m_c is a subvector corresponding to mixture component c and $\gamma_t(c)$ is the posterior probability that the observation at time t is generated by mixture component c. This information is used to train the UBM. Following that, the T matrix and i-vector are extracted. Dehak et. al. [6] were the first to make the further simplification and refinement in speaker verification of using Joint Factor Analysis (JFA), which combines the speaker space and channel space together. Named the Total Variability (TV) space, this captures the difference between speakers and across different channels. The speaker and channel dependent GMM mean supervector M for a given utterance can then be modelled as follows,

$$M = M_0 + Tw \tag{4}$$

where M_0 is the UBM supervector. T is the total variability vector, w is a random low dimension matrix with normal distribution $N(0,I)$. However in speaker diarization, unlike in speaker verification, the additional issue of intra-speaker variability must be considered. Thus we extend the basic TV model to explicitly compensate for the intra-conversation intra-speaker variability. In this extended model, each short speech segment in the conversation is represented as follows:

$$M_s = M_0 + Tw + U_1 x_s \tag{5}$$

where the definitions of M_0, T and w are the same as in Eqn. 4 for total variability. M_s is the GMM mean supervector of a speech segment s. Intra-conversation intra-speaker variability is modelled by $U_1 x_s$. More detail on the method of this representation can be found in [18]. Because of the nature of speaker diarization, segments can be very short and sometimes TV can not model these short segments well. However the intra-conversation intra-speaker variability can be explicitly compensated for to yield a more accurate representation. The intra-conversation intra-speaker variability subspace is trained according to Eqn. 5. To achieve this in practice, the output of the voice activity detector (VAD) already present in the front end of the diarization system is scanned to identify short segments. These are then extracted and used to explicitly model U_1.

2.2 DNN/i-vector system

The proposed DNN/i-vector system structure and sequence are shown in Fig. 2. This was inspired partly from recent work by Yun Lei et. al. [11] who reported that i-vectors derived from a well-trained DNN performed well for speaker identification (SID) tasks, compared to existing i-vector extraction methods. Further details of

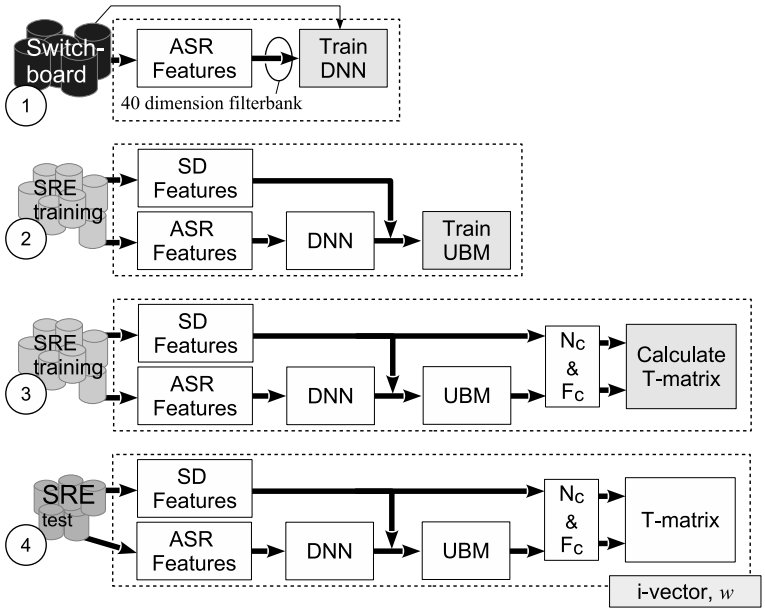

Fig. 2 *The proposed DNN/i-vector system, showing training step 1 using Swichtboard, training steps 2 to 3 using SRE training material, and i-vector extraction from SRE test data in step 4.*

DNNs can be found explained in a number of references, such as in [13]. Since the i-vector extraction step used in state-of-the-art diarizaton systems is similar to that for the SID task, we also attempt to create some well-trained DNNs using different input features for i-vector extraction. In other words, inspired by this SID approach, we adapt it into a diarization framework. Subsequently we evaluate whether the use of a DNN to train a UBM for i-vector extraction, yields benefit for the diarization task.

In traditional speaker verification systems employing i-vector models, the t-th speech frame $x_i^{(i)}$ is derived from a generative model using the i-th speech segment Gaussian distribution as follows,

$$x_t^{(i)} \sim \sum_k \gamma_k t^{(i)} N(\mu_k + T_k w^{(i)}, \Sigma_k) \tag{6}$$

$$\gamma_k t^{(i)} = p(k|x_t^{(i)}) \tag{7}$$

where μ_k and Σ_k is the mean and convariance of the k-th Gaussian and $\gamma_k t^{(i)}$ are the alignments of $x_t^{(i)}$. In general, the posterior of the k-th Gaussian is used to represent the alignments. By contrast to the traditional method, we first train a DNN using a large-scale development dataset (Fig. 2, step 1). Then use this DNN as a feature extractor from training data to train the UBM (Fig. 2, step 2). The means μ_k and the convariance Σ_k are now as follows,

$$\mu_k = \frac{\sum_{i,t} \gamma_{kt}^{(i)} x_t^{(i)}}{\sum_{i,t} \gamma_{kt}^{(i)}} \tag{8}$$

$$\Sigma_k = \frac{\sum_{i,t} \gamma_{kt}^{(i)} x_t^{(i)} x_t^{(i)^T}}{\sum_{i,t} \gamma_{kt}^{(i)}} - \mu_k \mu_k^T \tag{9}$$

The posteriors $p(k|x_t^{(i)})$ are computed from the ASR system for each class k for each frame. $x_t^{(i)}$ are the acoustic features which can differ from the features used by the DNN. ASR features (e.g., log-Mel filterbanks) act as the inputs to the DNN for generating posteriors for each senone, for each frame. These posterior probabilities, along with the *SD features*, are used to train the UBM (Fig. 2, step 2). In addition the posterior acts as the Gaussian distribution in a traditional UBM. The zeroth and first order statistics, as well as means μ_k and covariances Σ_k, are then obtained from the UBM. Following this, the zeroth and first order statistics from the UBM, operating on training data, are used to form a T-matrix as in UBM (Fig. 2, step 3), which then enables i-vector extraction from test data, and performance evaluation, to proceed as usual (described in Section 2.1 and shown in Fig. 2, step 4).

2.3 Selection of input features

The resulting DNN/i-vector system uses posterior probability information from the well-trained DNN in addition to the traditional UBM features to perform diarization. It would be reasonable to expect the UBM input features and the DNN input features to be the same (typically MFCCs), however the arrangement allows for an interesting possibility of using mismatched feature types. Note that a similar exploration may also be possible in other systems such as in the CNN/i-vector language identification approach of Lei et al. [10]. One set of input features, which we will term *ASR features* (since they are effectively performing an automatic speech recognition front end task), is used to train the DNN and subsequently to calculate the posterior probabilities. The second set of input features, which we term *SD features* (since they are those used typically in state-of-the-art speaker diarization systems), is used to train the UBM and for the following stages, alongside the posterior probabilities from the previously trained DNN. We will explore the effect of several choices for each feature input.

An important observation is that both sets of features should be properly aligned (i.e. the audio sample ranges forming the analysis frames of both features should be identical), otherwise substantial performance degradation occurs. Once the UBM is trained, we determine the zero and first order statistics to train the T-matrix and hence extract i-vectors as usual. The backend processing is also unchanged from existing systems.

3 Experiments and results

For baseline comparison, we use a state-of-the-art speaker diarization system [18] comprising voice activation detection (VAD), speaker change detection (SCD), segmentation, clustering, re-segmentation and refinements. After VAD and SCD, the

Table 1 UBM/i-vector performance of different size MFCCs and Gaussian mixture numbers.

Mix num	13-MFCC	26-MFCC	39-MFCC
128	1.36%	2.34%	3.42%
256	1.18%	2.28%	3.26%
512	1.06%	2.17%	3.14%
1024	0.91%	2.05%	3.10%

speech is chopped into small segments using the method in [4]. The proposed DNN/i-vector and UBM/i-vector approaches are then compared by using them to form i-vectors of the segments. The input segment list, and all subsequent processing and classification steps are common. Namely, the following step is to apply Principal Component Analysis (PCA) to reduce the dimensionality of the i-vector and obtain the directions of the maximum variability in the i-vector space. When we perform clustering, we apply k-means to the PCA-projected and reduced-dimension i-vectors based on their cosine distance. An HMM model is used to do the Viterbi decoding during the re-segmentation procedure. Meanwhile the cluster models are re-estimated through soft-clustering, as described in [3]. During the second pass, the segmentation results are further refined by iteratively extracting a single i-vector for each respective speaker from the re-segmented features, and reassigning the entire segment i-vector to its closest speaker i-vector, in terms of cosine similarity.

3.1 UBM/i-vector

The dataset used for training the baseline system is the SRE 05 and SRE 06 telephone data from NIST. SRE 08 data is then used for testing. In former experiments by other researchers using the same datasets, such as Shum et. al. [15], Wu et. al. [18] and Kenny et. al. [9] it was found that MFCC features tended to yield better results for speaker diarization than other common ASR features. Therefore we also adopt MFCC features for the UBM/i-vector baseline, and will additionally evaluate the performance of the basic 13-dimension MFCCs, as well as 26-dimension MFCC+Δ, and 39-dimension MFCC+Δ+Δ^2 features, which are commonly used in ASR and related domains. Several UBMs with 128, 256, 512 and 1024 diagonal components are also evaluated using these features. Meanwhile, the intra-conversation intra-speaker variability U matrix is formed from the same training data as used to determine the T matrix. The same prior work [15,18,9] reported better results for i-vectors of dimension 100 along and a rank of 100 for U compared with dimension 50 vectors (we will also evaluate both). Note that the traditional AHC, re-segmentation and refinements are performed identically on all of the compared systems.

Results are reported in Table 1, in terms of the composite Diarization Error Rate (DER) as defined by NIST for the SRE competitions. It can be seen that the best performing system has 1024 Gaussian mixtures and employs only the 13-dimensional MFCCs, yielding a performance score of 0.91%. This is comparable with the best performance reported by other authors on the same dataset, namely 0.91% in [18] and 0.90% in [15]. Generally speaking it might be expected that when the dimension of features grows, results will improve to some extent, because higher dimension

Table 2 UBM/i-vector performance of different of various U-matrix ranks and Gaussian mixture numbers.

	Long sentences		Short sentences	
Mix no.	$rank(U) = 50$	$rank(U) = 100$	$rank(U) = 50$	$rank(U) = 100$
128	1.39%	1.36%	1.59%	1.54%
256	1.17%	1.18%	1.41%	1.38%
512	1.08%	1.06%	1.35%	1.33%
1024	0.94%	0.91%	1.16%	1.12%

Table 3 DNN/i-vector performance comparison for different U-matrix ranks for for different DNN input (ASR) features.

	Long sentences		Short sentences	
U-matrix	39-dim PLP	40-dim FBK	39-dim PLP	40-dim FBK
$rank(U) = 50$	1.24%	0.89%	1.38%	1.02%
$rank(U) = 100$	1.18 %	0.72%	1.29%	0.87%

features can convey more information. However we see that the simplest features perform best here, which may be due to the fact that many segments are too short to reliably capture higher order statistics. Thus we maintain the 13-dimension MFCC input features throughout. As mentioned, we also explore the effect of the U-matrix rank. Systems were constructed with both rank 100 and rank 50, using 13 MFCC input features and a rank 100 T-matrix.

Results are reported separately in Table 2 for long (5 minute) sentences and short (1 minute) sentences. These figures confirm that the best performing U matrix rank for almost every tested condition is 100. There is a performance degradation of around 15% between results for the longer and shorter sentences.

In summary, this section has constructed a baseline UBM/i-vector system and explored the effects of several system parameters. Performance is shown to be on par with the best previously published state-of-the-art system performance on the SRE08 diarization test. We will now evaluate the DNN/i-vector system similarly, and compare against these results.

3.2 DNN/i-vector

The DNN configuration we adopt is similar to that used for ASR [2,5,14]. The system is first well trained using the large (300 hours) Switchboard dataset. The input layer of the DNN encompasses 15 frames of features (i.e. features from the current frame concatenated with features from a context of 7 neighbouring frames). The output layer matches the phonetic content of the dataset, comprising 3349 senones. Thus the structure of the DNN is one input layer, five hidden layers of size 1200 and one output layer (i.e. $600-\{1200 \times 5\}-3349$). In operation, each input frame corresponds to 40 log mel-filterbank coefficients and the DNN is used to yield the posterior probabilities from each frame plus context, on a frame-by-frame basis.

For consistency, the same features from the UBM/i-vector baseline system (i.e. 13-dimensional MFCCs) were used to compute sufficient statistics from the frame alignment given by the DNN, and the system hyper-parameters were also matched to the baseline system.

Table 4 DNN/i-vector performance comparison of different ASR features being input to the DNN along with different SD features used by the UBM.

SD	ASR features	
Features	39-dim PLP	40-dim FBK
13-MFCC	1.18%	0.72%
26-MFCC	2.49%	1.93%
39-MFCC	4.52%	2.74%
13-PLP	2.04%	0.96%
26-PLP	3.52%	2.47%
39-PLP	5.03%	3.75%

We repeated the U-matrix rank experiment of the previous section, but this time we also tried two different types of ASR features for training/operating the DNN. One system used 39-dimension perceptual linear prediction (PLP) features, which have shown promise in related speech fields. The second system used 40-dimensional mel filterbank (FBK) features. The structure of the remainder of the DNN for each system was identical. Results, shown in Table 3 are again presented separately for long and short sentences. These confirm the findings from the previous section that better performance is achieved with a U-matrix rank of 100, and additionally show that FBK outperforms PLP for ASR features. In this case it is noticeable that the performance degradation between long and short sentences is significantly reduced compared to the results in Table 2, indicating that the DNN/i-vector system is less sensitive to the source sentence length than the UBM/i-vector system.

For further comparison, setting the U- and T-matrix ranks at 100, we evaluated SD features using different orders of PLP and MFCC, allied with posterior probabilities from DNNs trained using both types of ASR feature (PLP and FBK). The results are shown in Table 4. Common sense would suggest that, since PLP features can carry more speaker-relevant information than the FBK features, they should perform better, whereas in fact they exhibit higher error rates for all tested conditions. In fact it may be that the ability of the DNN to learn its own discriminative features outweighs the well-known advantages of choosing perceptually-relevant features. In fact this ability of DNNs to infer relevant information from less structured data has been noted in related audio fields [13][5].

In operation, we effectively treat each output from the DNN as a UBM that is just concerned with phone information, without speaker-dependency. In practice we need to model the variance of all these outputs in a way similar to that for the TV method. So when we combine a DNN with TV it therefore follows that we should probably not use features for the DNN which emphasise speaker information. To put this another way, the ASR features should be those that perform better for speaker independent tasks, while the SD features should be those that perform better for speaker dependent tasks. Thus it is no surprise that the best result is the $600 - \{1200 \times 5\} - 3349$ DNN whose ASR features are 40 dimension FBK, allied with 13 dimension MFCC SD features.

Table 5 Performance comparison of state-of-the-art UBM/i-vector baseline, the Shum et. al [15] system, a phonetically-aligned GMM and the proposed DNN/i-vector method.

UBM/i-vector baseline	UBM/i-vector [15]	phonetically-aligned GMM	DNN/i-vector
0.91 %	0.90%	1.41%	**0.72**%

3.2.1 Summary

The overall performance of the baseline UBM/i-vector system, and that of the best published SRE08 result that the authors are aware of, is given in Table 5. Since the proposed DNN-based method benefits from the phonetic alignment from an underlying ASR system, a fair comparison would be against a phonetically-aligned GMM. This was then implemented (with matching 3349 mixtures) and evaluated, with results also presented in Table 5. Finally, the proposed DNN/i-vector system performance is also given. Clearly a significant improvement is achieved using the proposed method over the baseline, prior work and over the phonetically-aligned GMM. The latter result indicates that a major benefit is obtained through the discriminative learning of the DNN rather than from the underlying alignment. We can thus conclude that the DNN extractor that was first proposed for speaker verification, appears to work well for diarization, possibly because it is better able to represent shorter segments, since results above indicate that it is less sensitive to utterance length. In summary, the combination of the DNN-derived ASR features with the more traditional SD features with the DNN/i-vector approach proposed in this paper enables a 20% step improvement in performance over existing state-of-the-art UBM/i-vector approaches for SRE08 diarization evaluation. In effect, since the SRE08 evaluation consists of telephone speech, the diarization performance is being aided by a relevant feature extractor which has been better trained using the much larger training dataset of Switchboard telephone speech.

4 Conclusion

This paper has proposed a novel diarization method that makes use of a well-trained DNN to enhance the representation of speech segments through an accurate phonetic classification. This is inspired by recent work in the related speaker verification domain, extended here to cater for intra-speaker, intra-conversational variability in a speaker diarization context. The relatively speaker-independent DNN-derived UBM features, allied with more traditional speaker diarization features which capture speaker dependent information, are shown to yield a more representative i-vector representation of individual speech segments. In operation, the DNN is well trained using filterbank features from a very large database of telephone speech, and performs a roughly similar task to the GMM in a typical UBM/i-vector system. One advantage of utilising two separate models is that different representations can be chosen for each. In fact, evaluations tested several of the more common feature types, including zero, first and second order MFCCs, PLPs and filterbanks, with the best performance being obtained from the DNN trained using 40-dimensional filterbank data combined

with 13-dimensional MFCC features. Other evaluations investigated the effect of various parameters such as matrix rank and number of Gaussian mixtures for the more traditional UBM/i-vector approach. The overall performance of the system for the SRE08 task is significantly better than that of the baseline method, as well as the currently published state-of-the art UBM/i-vector system performance.

5 Acknowledgements

For part of this research project, Ian McLoughlin was supported by the Fundamental Research Funds for the Central Universities, China, under grant no. WK2100000002. Yan Song is supported by the Natural Science Foundation of China (NSFC, Grant No. 61172158).

References

1. Ajmera, J., Wooters, C.: A robust speaker clustering algorithm. In: Automatic Speech Recognition and Understanding, 2003. ASRU'03. 2003 IEEE Workshop on, pp. 411–416. IEEE (2003)
2. Bao, Y., Jiang, H., Liu, C., Hu, Y., Dai, L.: Investigation on dimensionality reduction of concatenated features with deep neural network for lvcsr systems. In: Signal Processing (ICSP), 2012 IEEE 11th International Conference on, vol. 1, pp. 562–566 (2012). DOI 10.1109/ICoSP.2012.6491550
3. Castaldo, F., Colibro, D., Dalmasso, E., Laface, P., Vair, C.: Stream-based speaker segmentation using speaker factors and eigenvoices. In: Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on, pp. 4133–4136 (2008). DOI 10.1109/ICASSP.2008.4518564
4. Chen, S., Gopalakrishnan, P.: Speaker, environment and channel change detection and clustering via the bayesian information criterion. In: Proc. DARPA Broadcast News Transcription and Understanding Workshop, vol. 8, pp. 127–132. Virginia, USA (1998)
5. Dahl, G., Yu, D., Deng, L., Acero, A.: Context-dependent pre-trained deep neural networks for large vocabulary speech recognition. IEEE Transactions on Audio, Speech, and Language Processing (receiving 2013 IEEE SPS Best Paper Award) **20**(1), 30–42 (2012). URL http://research.microsoft.com/apps/pubs/default.aspx?id=144412
6. Dehak, N., Kenny, P., Dehak, R., Dumouchel, P., Ouellet, P.: Front-end factor analysis for speaker verification. Audio, Speech, and Language Processing, IEEE Transactions on **19**(4), 788–798 (2011). DOI 10.1109/TASL.2010.2064307
7. Gauvain, J.L., Lamel, L., Adda, G.: Partitioning and transcription of broadcast news data. In: ICSLP, vol. 98-5, pp. 1335–1338 (1998)
8. Hinton, G., Deng, L., Yu, D., Dahl, G., Mohamed, A., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T., Kingsbury, B.: Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. Signal Processing Magazine, IEEE **29**(6), 82–97 (2012). DOI 10.1109/MSP.2012.2205597
9. Kenny, P., Reynolds, D., Castaldo, F.: Diarization of telephone conversations using factor analysis. Selected Topics in Signal Processing, IEEE Journal of **4**(6), 1059–1070 (2010). DOI 10.1109/JSTSP.2010.2081790
10. Lei, Y., Ferrer, L., Lawson, A., McLaren, M., Scheffer, N.: Application of convolutional neural networks to language identification in noisy conditions. Proc. Odyssey-14, Joensuu, Finland (2014)
11. Lei, Y., Ferrer, L., McLaren, M., Scheffer, N.: A deep neural network speaker verification system targeting microphone speech. In: Proc. Interspeech (2014)
12. McLaren, M., Lei, Y., Scheffer, N., Ferrer, L.: Application of convolutional neural networks to speaker recognition in noisy conditions. In: Fifteenth Annual Conference of the International Speech Communication Association (2014)
13. McLoughlin, I., Zhang, H.M., Xie, Z., Song, Y., Xiao, W.: Robust sound event classification using deep neural networks. Audio, Speech, and Language Processing, IEEE Transactions on **23**, 540–552 (2015)

14. Seide, F., Li, G., Chen, X., Yu, D.: Feature engineering in context-dependent deep neural networks for conversational speech transcription. In: ASRU 2011. IEEE (2011). URL http://research.microsoft.com/apps/pubs/default.aspx?id=157341
15. Shum, S., Dehak, N., Chuangsuwanich, E., Reynolds, D.A., Glass, J.R.: Exploiting intra-conversation variability for speaker diarization. In: INTERSPEECH'11, pp. 945–948 (2011)
16. Song, Y., Jiang, B., Bao, Y., Wei, S., Dai, L.R.: I-vector representation based on bottleneck features for language identification. Electronics Letters **49**(24), 1569–1570 (2013)
17. Tranter, S., Reynolds, D.: An overview of automatic speaker diarization systems. Audio, Speech, and Language Processing, IEEE Transactions on **14**(5), 1557–1565 (2006). DOI 10.1109/TASL.2006.878256
18. Wu, K., Song, Y., Guo, W., Dai, L.: Intra-conversation intra-speaker variability compensation for speaker clustering. In: Chinese Spoken Language Processing (ISCSLP), 2012 8th International Symposium on, pp. 330–334 (2012). DOI 10.1109/ISCSLP.2012.6423465