Closed-Set Text-Independent Automatic Speaker Recognition System Using VQ/GMM

Bidhan Barai, Debayan Das, Nibaran Das, Subhadip Basu and Mita Nasipuri

Abstract Automatic speaker recognition (ASR) is one type of biometric recognition of human, known as voice biometric recognition. Among plenty of acoustic features, Mel-Frequency Cepstral Coefficients (MFCCs) and Gammatone Frequency Cepstral Coefficients (GFCCs) are used popularly in ASR. The state-of-the-art techniques for modeling/classification(s) are Vector Quantization (VQ), Gaussian Mixture Models (GMMs), Hidden Markov Model (HMM), Artificial Neural Network (ANN), Deep Neural Network (DNN). In this paper, we cite our experimental results upon three databases, namely Hyke-2011, ELSDSR, and IITG-MV SR Phase-I, based on MFCCs and VQ/GMM where maximum log-likelihood (MLL) scoring technique is used for the recognition of speakers and analyzed the effect of Gaussian components as well as Mel-scale filter bank's minimum frequency. By adjusting proper Gaussian components and minimum frequency, the accuracies have been increased by 10–20% in noisy environment.

Keywords ASR · Acoustic Feature · MFCC · GFCC · VQ · GMM · MLL Score

1 Introduction

Automatic speaker recognition (ASR) system was first introduced by Pruzansky et al. [7]. There are two primary tasks within the *speaker recognition* (SR), namely *speaker identification* (SI) and *speaker verification* (SV). This paper concerns with

B. Barai (✉) · D. Das · N. Das · S. Basu · M. Nasipuri
Jadavpur University, Kolkata 700032, India
e-mail: bidhanb@research.jdvu.ac.in

D. Das
e-mail: debayan.157@gmail.com

N. Das
e-mail: nibaran@cse.jdvu.ac.in

S. Basu
e-mail: subhadip@cse.jdvu.ac.in

M. Nasipuri
e-mail: mnasipuri@cse.jdvu.ac.in

V. Bhateja et al. (eds.), *Intelligent Engineering Informatics*, Advances in Intelligent Systems and Computing 695, https://doi.org/10.1007/978-981-10-7566-7_33

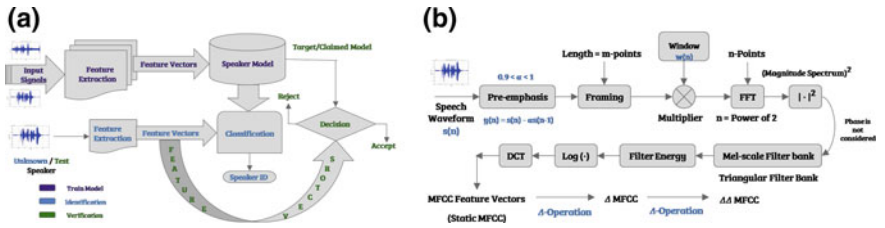

Fig. 1 Block diagram **a** for SR and **b** for MFCC feature extraction

SI and we have used the terms SI and SR synonymously. A general block diagram for SI and SV is shown in Fig. 1a. The SR can also be classified as *text-dependent* and *text-independent* recognition and further divided into *open-set* and *closed-set* identification.

An immense number of features are invented, but at present day the features that are popularly used for robust SR are *Linear Predictive Cepstral Coefficient* (LPCC) and *Perceptual Linear Predictive Cepstral Coefficient* (PLPCC) [9], *Gammatone Frequency Cepstral Coefficient* (GFCC) [11], *Mel-Frequency Cepstral Coefficient* (MFCC), combination of *MFCC and phase information* [6], *Modified Group Delay Feature* (MODGDF) [5], *Mel Filter Bank Energy-Based Slope Feature* [4], *i-Vector* [3], *Bottleneck Feature of DNN* (BF-DNN). In some cases to increase robustness, combined features are developed by fusion of some of these robust features. Some of combined features are LPCC+MFCC, MFCC+GFCC, PLPCC+MFCC+GFCC. The state-of-the-art methods for modeling/classification are *Vector Quantization* (VQ) [10], *Hidden Markov Model* (HMM), *Gaussian Mixture Model* (GMM) [8], *GMM-Universal Background Model* (GMM-UBM) [1], *Support Vector Machine* (SVM), *Deep Neural Network* (DNN) and hybrid models like *VQ/GMM*, *SVM/GMM*, *HMM/GMM*. Among these, the hybrid modelbreak HMM/GMM is very useful for SR in noisy environment because HMM isolates the speech feature vectors from the noisy feature vectors and then estimates the multivariate probability density function using GMM in the feature space.

2 Feature Extraction

The first step of SR is feature extraction, also known as *front-end processing*. It *transforms*(*maps*) the raw speech data into the *featurespace*. The features like MFCC and GFCC are computed using frequency domain analysis and *Spectrogram*. In our experiment, the MFCC feature is used for SR. The block diagram for extracting MFCC feature is shown in Fig. 1b. The computation of MFCC is discussed briefly as follows:

Pre-emphasis: The speech signal is passed through a HPF to increase the amplitude of high frequency. If $s(n)$ is the speech signal, then it is implemented as $\tilde{s}(n) = s(n) - \alpha s(n)$, where $0.9 < \alpha < 1$. Generally, the typical value of α is 0.97.

Framing: To compute MFCC, short time processing of speech signal is required. The whole speech signal is broken into overlapping frames. Typically, 25–60 ms frame is chosen with the overlap of 15–45 ms.

Window: For *Short Time Fourier Transform* (STFT) for $x(n)$, where $x(n)$ be a short time frame, we must choose a window function $h(n)$. A typical window function $h(n)$ is given by

$$h(n) = \beta - (1 - \beta)\, cos\left(\frac{2\pi n}{N-1}\right) \tag{1}$$

where N is the window length. Here, $\beta = 0.54$ for Hamming window and $\beta = 0.5$ for Hanning window.

DFT and FFT: The *Discrete Fourier Transform* (DFT) for the windowed signal is computed as $X(\omega, n) = \sum_{m=-\infty}^{\infty} x(m)h(n-m)e^{-j\omega n}$, *where* $0 \le n \le N$. For discrete STFT, continuous $X(\omega, n)$ is sampled with N (length of windowed signal) equal points in frequency (ω) as $X(k, n) = X(k) = X(\omega, n)|_{\omega = \frac{2\pi}{N}k}$, *where* $0 \le k \le N$. The graphical display of $|X(k, n)|$ as color intensity is known as *Spectrogram*. Fortunately, two previous equations can be simplified with the help of *Fast Fourier Transform* (FFT) as $X(k) = \mathcal{FFT}\{x(n)h(n)\}$, *where* $0 \le k \le N$. To facilitate *FFT*, we must make N as power of 2. To do so, it is required to pad zeros with the frame to make frame length a nearest power of 2 if N is not a power of 2, otherwise zero padding is not required.

Magnitude Spectrum: The squared magnitude spectrum is computed as $S(k) = |X(k)|^2$, where $0 \le k \le N$

Mel-Scale Filter Bank: In Mel scale, n_B number of overlapping triangular filters are set between $M(f_{min})$ and $M(f_{max})$ to form a filter bank. The relation between Mel scale (mel) and Linear scale (Hz) is given by

$$M(f) = 1127 log_e\left(1 + \frac{f}{700}\right) \tag{2}$$

where f in Hz and $M(f)$ in *mel*. A filter in filter bank is characterized by start, center, and end frequencies, i.e., $M(f_s)$, $M(f_c)$, and end $M(f_e)$, respectively. Using inverse operation of (2), we can compute f_s, f_c, and f_e using the following equation:

$$f = 700(e^{\frac{M(f)}{1127}} - 1) \tag{3}$$

where f in Hz and $M(f)$ in *mel*. Next, we map the frequencies f_s, f_c, and f_e to the corresponding nearest FFT index numbers given by f^s_{bin}, f^c_{bin}, and f^e_{bin}, respectively, which are called FFT bins by using the following equation:

$$f_{bin} = \lfloor \frac{(N+1).f}{F_s} \rfloor, \qquad f = f_s, f_c, f_e \tag{4}$$

Here, F_s is the sampling frequency of the speech signal. The filter weight is maximum at center bin f^c_{bin} which is 1, and zero weight is assumed at start and end bins, f^s_{bin} and f^e_{bin}. The weights are calculated as follows:

$$H_m(k) = \begin{cases} 0 & \text{if } k < f^s_{bin} \\ \frac{k - f^s_{bin}}{f^c_{bin} - f^s_{bin}} & \text{if } f^s_{bin} \le k \le f^c_{bin} \\ \frac{f^e_{bin} - k}{f^s_{bin} - f^c_{bin}} & \text{if } f^c_{bin} \le k \le f^e_{bin} \\ 0 & \text{if } k > f^e_{bin} \end{cases} \tag{5}$$

Filter Energy: The filter bank is set over the squared magnitude spectrum $S(k)$. For each filter in the filter bank, the filter weight is multiplied with the corresponding $S(k)$ and summed up all the products to get the *filter energy*, denoted by $\{\tilde{S}(k)\}_{k=1}^{k=n_B}$. Taking *logarithm*, we get *log energies*, $\{log(\tilde{S}(k)\}_{k=1}^{k=n_B}$.

DCT: To perform *Discrete Cosine Transform* (DCT), the following operation is carried out.

$$C_n = \sum_{k=1}^{D} (\log \tilde{S}(k)) cos(n(k - \frac{1}{2})\frac{\pi}{D}), \qquad n = 1, 2, ..., D \tag{6}$$

Here, $D = n_B$ is the dimension of the vector C_n which is called MFCC vector.

3 Speaker Model

The models that are used frequently in SR are Linear Discriminative Analysis (LDA), Probabilistic LDA (PLDA), Gaussian Mixture Model (GMM), GMM-Universal Background Model (GMM-UBM), Hidden Markov Model (HMM), Artificial Neural Network (ANN), Deep Neural Network (DNN), Vector Quantization, Dynamic Time Warping (DTW), Support Vector Machine (SVM). GMM is the most popular model used in SR. These models are used to build speaker templates. Score domain compensation aims to remove handset-dependent biases from the likelihood ratio scores. The most prevalent methods include H-norm, Z-norm, and T-norm.

3.1 Vector Quantization (VQ)

It is used as a preliminary method for clustering data, so that the process of Vector Quantization(VQ) can be applied more suitably. The grouping is done by minimizing *Euclidean* distance between vectors. If we get V number of vectors after the feature

extraction phase, then after VQ we will get K vectors where $K < V$. This set of K vectors is called *codebook* which represents the set of centroids of the individual clusters. In the modeling section, the GMM model is built upon these K vectors.

3.2 Gaussian Mixture Model (GMM)

Let for jth speaker there are K number of quantized feature vectors of dimension D, viz. $\mathcal{X} = \{\boldsymbol{x_t} \in \mathbb{R}^D : 1 \leq t \leq K\}$. The GMM for jth speaker, λ_j, is the weighted sum of M component D-variate Gaussian densities where mixture *weights* w_i $\{i = 1\ to\ M\}$ must satisfy $\sum_{i=1}^{M} w_i = 1$. Hence, the GMM model λ_j is given by $p(\boldsymbol{x_t}|\lambda_j) = \sum_{i=1}^{M} w_i \mathcal{N}(\boldsymbol{x}_t; \boldsymbol{\mu_i}, \boldsymbol{\Sigma}_i)$ where $\mathcal{N}$ $(\boldsymbol{x}_t; \boldsymbol{\mu_i}, \boldsymbol{\Sigma}_i)$ $\{i = 1\ to\ M\}$ are D-variate Gaussian density functions given by

$$\mathcal{N}(\boldsymbol{x}_t; \boldsymbol{\mu_i}, \boldsymbol{\Sigma}_i) = \frac{1}{(2\pi)^{D/2}|\boldsymbol{\Sigma}_i|^{1/2}} e^{-\frac{1}{2}(\boldsymbol{x}_t - \boldsymbol{\mu}_i)' \boldsymbol{\Sigma}_i^{-1} (\boldsymbol{x}_t - \boldsymbol{\mu}_i)} \tag{7}$$

with mean vector $\boldsymbol{\mu}_i \in \mathbb{R}^D$ and covariance matrix $\boldsymbol{\Sigma}_i \in \mathbb{R}^{D \times D}$. $(\boldsymbol{x}_t - \boldsymbol{\mu}_i)'$ represents the transpose of vector $(\boldsymbol{x}_t - \boldsymbol{\mu}_i)$. The GMM model for jth speaker λ_j is parameterized by weight w_i, mean vector $\boldsymbol{\mu}_i$, and covariance matrix $\boldsymbol{\Sigma}_i$. Hence, $\lambda_j = \{w_i, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\}$.

These three parameters are computed with the help of EM algorithm. In the beginning of the EM iteration, the three parameters are required to initialize per Gaussian component. Initialization could be absolutely random, but in order to converge faster one can use *k-means* clustering algorithm also. A block diagram for GMM is shown in Fig. 2.

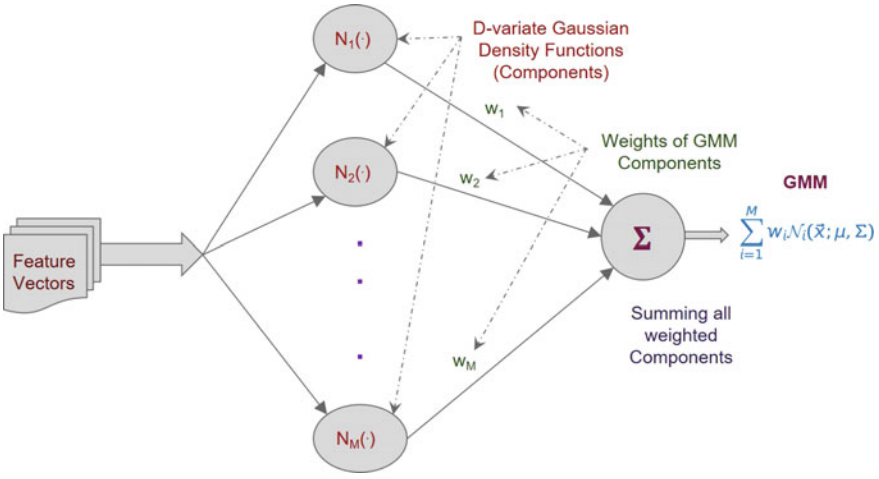

Fig. 2 A block diagram of GMM

3.2.1 Maximum Likelihood (ML) Parameter Estimation (MLE):

The aim of the EM algorithm is to re-estimate the parameters after initialization, which give the maximum likelihood (ML) value given by

$$p(\mathcal{X}|\lambda_j) = \prod_{t=1}^{K} p(\boldsymbol{x}_t|\lambda_j) \tag{8}$$

The EM algorithm begins with an initial model λ_0 and re-estimate a new model λ in such a way that it always provides a new λ for which $p(\mathcal{X}|\lambda) \geq p(\mathcal{X}|\lambda_0)$.

To estimate the model parameters, mean vector $\boldsymbol{\mu}_i$ is initialized using k-means clustering algorithm and this mean vector is used to initialize covariance matrix $\boldsymbol{\Sigma}_i$. w_i is assumed to $1/M$ as its initial value. In each EM iteration, three parameters are re-estimated according to the following three equations to get the new model λ_{new}.

$$w_i = \frac{1}{K}\sum_{t=1}^{K} \mathcal{P}(i|\boldsymbol{x}_t, \lambda_j) \tag{9}$$

$$\boldsymbol{\mu}_i = \frac{\sum_{t=1}^{K} \mathcal{P}(i|\boldsymbol{x}_t, \lambda_j)\boldsymbol{x}_t}{\sum_{t=1}^{K} \mathcal{P}(i|\boldsymbol{x}_t, \lambda_j)} \tag{10}$$

$$\boldsymbol{\Sigma}_i = \frac{\sum_{t=1}^{K} \mathcal{P}(i|\boldsymbol{x}_t, \lambda_j)(\boldsymbol{x}_i - \boldsymbol{\mu}_i)(\boldsymbol{x}_i - \boldsymbol{\mu}_i)'}{\sum_{t=1}^{K} \mathcal{P}(i|\boldsymbol{x}_t, \lambda_j)} \tag{11}$$

The iteration continues until a suitable convergence criteria holds. For the covariance matrix $\boldsymbol{\Sigma}_i$, only diagonal elements are taken and all off-diagonal elements are set to zero. The probability $\mathcal{P}(i|\boldsymbol{x}_t, \lambda_j)$ is given by

$$\mathcal{P}(i|\boldsymbol{x}_t, \lambda_i) = \frac{w_i \mathcal{N}(\boldsymbol{x}_t; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)}{\sum_{j=1}^{M} w_j \mathcal{N}(\boldsymbol{x}_t; \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} \tag{12}$$

4 Speaker Identification with MLL Score

Let there are S speakers $\mathcal{S} = \{1, 2, 3, ..., S\}$ and they are represented by the GMM's $\lambda_1, \lambda_2, \lambda_3, ..., \lambda_S$. Now, the task is to find the speaker model with the maximum *posteriori* probability for the set of feature vectors $\mathcal{X}$ of test speaker. Using minimum error Bayes' decision rule, the identified speaker is given by

$$\hat{S} = \arg\max_{k \in S} \left(\sum_{t=1}^{K} log(p(\boldsymbol{x}_t | \lambda_k)) \right) \tag{13}$$

Here, $\hat{S}$ is the identified speaker and kth speaker's log-likelihood (LL) score is given by $\sum_{t=1}^{K} log(p(\boldsymbol{x}_t | \lambda_k))$ [8]. The identified speaker $\hat{S}$ has the maximum log-likelihood (MLL) score.

5 Experimental Results and Discussion

We conducted the SR experiment extensively over the three databases, namely IITG Multi-Variability Speaker Recognition Database (IITG-MV SR), ELSDSR, and Hyke-2011. The IITG-MV SR database contains recorded speech from five recording devices, namely digital recorder (D01), Headset (H01), Tablet PC (T01), Nokia 5130c mobile (M01), and Sony Ericsson W350i mobile (M02), in noisy environment. However, ELSDSR and Hyke-2011 contain clean speech; i.e., noise level is very low and the speeches are recorded with a microphone. The sampling frequency for D01, H01, T01 is 16 kHz, for M01, M02 is 8 kHz, and for ELSDSR, Hyke-2011 is 8 kHz. We chose frame size about 25 ms and overlap about 17 ms, i.e., frameshift is $(25 - 17) = 8$ ms for 16 kHz speech signal and 50 ms frame size and about 34 ms overlap; i.e., frameshift is $(50 - 34) = 16$ ms for 8 kHz speech signal. The pre-emphasis factor α is set to 0.97. To compute *FFT* 512-point, FFT algorithm is used. For *mel*-scale frequency conversion, maximum and minimum linear frequencies are $f_{min} = 0, 300$ Hz and $f_{max} = 5000$ Hz. The frequency, f_{min}, has significant effect on the accuracy of ASR. Number of triangular filters in filter bank is $n_B = 26$ which produces 26 MFC coefficients, and among them first 13 MFCC are chosen to create MFCC feature vector of dimension $D = 13$. The accuracy rate for the mentioned databases is shown in Table 1. In VQ, we consider 512 clusters, to reduce large number of vectors, upon which GMM is built using 5 EM iteration.

It is clearly shown that the accuracy rate is low for noisy speech as compared to the clean speech. This is because the noise level distorts the frequency spectrum of the signal considerably and vectors in the feature space are shifted and distorted from the original vectors. All the databases show the highest accuracy with vector dimension equal to 13, no. of Gaussian components equal to 32, and accuracy degrades beyond this limits. Another observation is that the bandwidth of the filters in the filter bank in linear scale (Hz) also influences the accuracy rate. The SR under mismatch and reverberant conditions are more challenging tasks, because in these cases, performance of SR system degrades drastically. Other important issues for SR are language dependency and device mismatch. It has been seen that the accuracy rate degrades if there is a mismatch of language between training and testing data. Specially, for the device mismatch between training and testing data, the accuracy rate degrades drastically. Though GMM shows satisfactory accuracy rate, HMM is more robust than GMM and provides better result in environmental mismatch condition. Hybrid

Table 1 Recognition accuracy for databases IITG-MV SR, Hyke-2011 and ELSDSR for 5 EM iteration and 512 VQ clusters

Database name and Recording device		Type of speech	No. of speakers	No. of Gaussian components	Starting frequency (f_{min})	Vector dimension (D)	Testing time (sec)	Accuracy (%)
IITG-MV SR	DVR (D01)	**Noisy**	100	16	0	13	3638	96
				16	300	13	3287	95
				32	0	13	5119	96
				32	300	13	5336	**96**
	Headset (H01)	**Noisy**	100	16	0	13	4010	70
				16	300	13	4239	89
				32	0	13	5871	76
				32	300	13	5801	**93**
	Tablet PC (T01)	**Noisy**	100	16	0	13	3922	90
				16	300	13	3513	89
				32	0	13	5433	**91**
				32	300	13	5129	89
	Nokia 5130c (M01)	**Noisy**	100	16	0	13	2570	94
				16	300	13	2741	**95**
				32	0	13	4434	92
				32	300	13	4806	94
	Sony Ericsson W350i (M02)	**Noisy**	100	16	0	13	2824	86
				16	300	13	2641	87
				32	0	13	4456	86
				32	300	13	4645	**90**
Hyke-2011	Mic	**Clean**	83	16	0	13	3660	**100**
				16	300	13	3830	100
				32	0	13	6780	100
				32	300	13	6812	100
ELSDSR	Mic	**Clean**	22	16	0	13	656	**100**
				16	300	13	714	100
				32	0	13	1292	100
				32	300	13	1367	100

HMM/GMM-based SR in noisy environment performs better than only GMM-based SR. In noisy environment, the accuracy of GMM-based SR degrades more rapidly than the HMM/GMM-based SR.

6 Conclusion

SR has a very close relation with the speech recognition. Emotion extraction from speech data using corpus-based feature and sentiment orientation technique could be thought of as an extension of SR experiment [2]. In this paper, we cite SR experiment and analyze feature extraction and modeling/classification steps. It is very important to mention that number of GMM components and Mel filter bank's minimum frequency f_{min} have significant influence on the recognition accuracy. Since there are sufficient differences in accuracies between clean speech data and noisy speech data, we can infer that noise level shifts the data from its true orientation. Various normalization techniques in feature domain and modeling/classification domain could be applied to combat with the unwanted shift of data in feature space. Indeed, before transforming data into feature space various filtering techniques to reduce the effect of noise are also available.

Acknowledgements This project is partially supported by the CMATER laboratory of the Computer Science and Engineering Department, Jadavpur University, India, TEQIP-II, PURSE-II, and UPE-II projects of Government of India. Subhadip Basu is partially supported by the Research Award (F.30-31/2016(SA-II)) from UGC, Government of India. Bidhan Barai is partially supported by the RGNF Research Award (F1-17.1/2014-15/RGNF-2014-15-SC-WES-67459/(SA-III)) from UGC, Government of India.

References

1. Campbell, W.M., Sturim, D.E., Reynolds, D.A.: Support vector machines using gmm supervectors for speaker verification. IEEE Signal Process. Lett. **13**(5), 308–311 (2006)
2. Jain, V.K., Kumar, S., Fernandes, S.L.: Extraction of emotions from multilingual text using intelligent text processing and computational linguistics. J. Comput. Sci. (2017)
3. Kanagasundaram, A., Vogt, R., Dean, D.B., Sridharan, S., Mason, M.W.: I-vector based speaker recognition on short utterances. In: Proceedings of the 12th Annual Conference of the International Speech Communication Association, pp. 2341–2344. International Speech Communication Association (ISCA) (2011)
4. Madikeri, S.R., Murthy, H.A.: Mel filter bank energy-based slope feature and its application to speaker recognition. In: Communications (NCC), 2011 National Conference on, pp. 1–4. IEEE (2011)
5. Murthy, H.A., Yegnanarayana, B.: Group delay functions and its applications in speech technology. Sadhana **36**(5), 745–782 (2011)

6. Nakagawa, S., Wang, L., Ohtsuka, S.: Speaker identification and verification by combining MFCC and phase information. IEEE Trans. Audio Speech Lang. Process. **20**(4), 1085–1095 (2012)
7. Pruzansky, S.: Pattern-matching procedure for automatic talker recognition. J. Acoust. Soc. Am. **35**(3), 354–358 (1963)
8. Reynolds, D.A., Rose, R.C.: Robust text-independent speaker identification using gaussian mixture speaker models. IEEE Trans. Speech Audio Process. **3**(1), 72–83 (1995)
9. Sapijaszko, G.I., Mikhael, W.B.: An overview of recent window based feature extraction algorithms for speaker recognition. In: Circuits and Systems (MWSCAS), 2012 IEEE 55th International Midwest Symposium on, pp. 880–883. IEEE (2012)
10. Soong, F.K., Rosenberg, A.E., Juang, B.H., Rabiner, L.R.: Report: a vector quantization approach to speaker recognition. AT&T Techn. J. **66**(2), 14–26 (1987)
11. Zhao, X., Wang, D.: Analyzing noise robustness of MFCC and GFCC features in speaker identification. In: Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on, pp. 7204–7208. IEEE (2013)