Influence of Feature Dimensionality and Model Complexity on Speaker Verification Performance

Adam Dustor, Piotr Kłosowski, and Jacek Izydorczyk

Silesian University of Technology, Institute of Electronics
Akademicka 16, 44-100 Gliwice, Poland
{adam.dustor,piotr.klosowski,jacek.izydorczyk}@polsl.pl

Abstract. This paper provides description of a text dependent speaker recognition system based on vector quantization approach. The scope of this paper is to check influence of feature dimensionality and the complexity of the speaker model on verification process. Provided results show that MFCC features yield the lowest possible verification errors among all tested parameters. Although dimensionality of feature vectors is important, there is no need to increase it above some level as the improvement in verification performance is relatively low and computational complexity increases. Far more important than dimensionality is complexity of the speaker model.

Keywords: biometrics, security, speaker verification, voice identification, feature extraction.

1 Introduction

Division of Telecommunication, a part of the Institute of Electronics and Faculty of Automatic Control, Electronics and Computer Science of the Silesian University of Technology, for many years specializes in speech and speaker recognition [1–4]. One of the results of conducted research is presented in this paper which is devoted to speaker verification.

Speaker recognition is the process of automatically recognizing who is speaking by analysis speaker-specific information included in spoken utterances. This process encompasses identification and verification. The purpose of speaker identification is to determine the identity of an individual from a sample of his or her voice and it can be divided into two main categories, i.e. closed-set and open-set. In a closed-set identification there is an assumption that only registered speakers have an access to the system which makes a decision 1 from K, where K is the number of previously registered speakers. In an open-set identification there is no such an assumption so the identification system has to determine whether the testing utterance comes from a registered speaker or not and if yes it should determine his or her identity. The purpose of speaker verification is to decide whether a speaker is whom he claims to be. Most of the applications in which voice is used to confirm the identity claim of a speaker are classified as speaker verification. Speaker recognition systems can also be divided

A. Kwiecień, P. Gaj, and P. Stera (Eds.): CN 2014, CCIS 431, pp. 177–186, 2014.

into text-dependent and text-independent. In text-dependent mode the speaker has to provide the same utterance for training and testing, whereas in text-independent systems there are no such constraints. The text-dependent systems are usually based on template matching techniques in which the time axes of an input speech sample and each reference template are aligned and the similarity between them is accumulated from the beginning to the end of the utterance. Because these systems can directly exploit voice individuality associated with each phoneme or syllable, they usually achieve higher recognition performance than text-independent systems [5].

The paper is organized in the following way. At first fundamentals of speaker verification are discussed, next feature parameters and construction of speaker model are presented. At last achieved speaker verification results for the given dimensionality and complexity of the speaker model are shown.

2 Speaker Recognition

Basic structure of speaker verification system is shown in Fig. 1. Speech signal is cut into short fragments, which usually last for 20–30 ms known as speech frames. Feature extraction is responsible for extracting from each frame a set of parameters known as feature vectors. Extracted sequence of vectors is then compared to speaker model (in verification) or speaker models (in identification) by pattern matching. The purpose of pattern matching is to measure similarity between test utterance and speaker model. In identification an unknown speaker is identified as the speaker whose model best matches the test utterance. In verification the similarity between input test sequence and claimed model must be good enough to accept the speaker as whom he claims to be. As a result, verification requires choosing decision threshold. If computed distance is less than this threshold, a decision can be made that the speaker is whom he claims to be. How to find an optimum value of this threshold still remains a problem for scientist [6]. Another very desired property of this threshold is its independence of a speaker, which means that there is one threshold for all speakers. Since these problems are not solved satisfactorily, they still remain very important research issues, apart from problem of finding the best set of speech parameters, which must be studied further to make an improvement in speaker recognition technology.

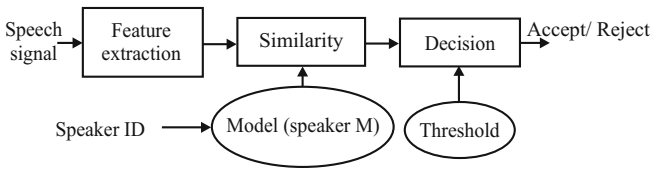

Fig. 1. Speaker verification scheme

3 Feature Parameters

One of the most important procedures in speaker recognition is feature extraction. The extracted parameters should have high speaker discrimination power, high interspeaker variability and low intraspeaker variability. Only such parameters guarantee very good speaker recognition results. Although there are a lot of techniques for extracting speaker specific information from the speech signal, probably the most important are features based on frequency spectrum of the speech as linear prediction coefficients LPC and parameters derived from them like LPC cepstrum known as LPCC fetaures.

3.1 LPC Parameters

Calculation of these parameters is based on the linear model for speech production shown in Fig. 2, where the glottal pulse, vocal tract and radiation are individually modeled as linear filters.

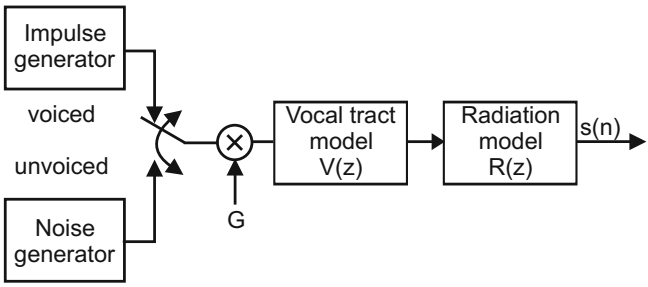

Fig. 2. The linear model of speech production

The source is either a random sequence for unvoiced sounds or a quasi-periodic impulse sequence for voiced sounds. The gain factor G controls the intensity of the excitation. The vocal tract is modeled by transfer function $V(z)$ whereas the radiation model $R(z)$ describes the air pressure at the lips. Combining these parts of the vocal tract yields an all-pole transfer function

$$H(z) = G(z)V(z)R(z) = \frac{G}{A(z)} \ , \tag{1}$$

$$H(z) = \frac{G}{A(z)} = \frac{G}{1 - \sum_{k=1}^{p} a_k z^{-k}} \ , \tag{2}$$

where p is the prediction order and a_k are predictor coefficients. The LPC model of the speech signal specifies that a speech sample $s(n)$ can be represented as a linear sum of the p previous samples plus an excitation term

$$s(n) = \sum_{k=1}^{p} a_k s(n-k) + Gu(n) \ . \tag{3}$$

As in speech applications the excitation term is usually unknown it is ignored and the LPC approximation of $s(n)$ depends only on the past output samples. Unfortunately some speaker specific information is included in the excitation term (e.g. fundamental frequency) which affects on the performance of the LPC based speaker recognition systems. Since vocal-tract changes its configuration over time, in order to model it, the predictor coefficients a_k must be computed adaptively over short intervals (10 ms to 30 ms) called frames during which time-invariance is assumed. There are two standard methods of solving for the predictor coefficients: autocorrelation and covariance method. Both of them are based on minimizing the mean-square value E of the prediction error $e(n)$ which is the difference between the actual and the predicted value of the speech sample

$$E = \sum_{n=0}^{N-1+p} e^2(n) = \sum_{n=0}^{N-1+p} \left[s(n) - \sum_{k=1}^{p} a_k s(n-k) \right]^2 . \tag{4}$$

The a_k parameters can be found after solving the linear equations resulting from

$$\frac{\partial E}{\partial a_i} = 0 \ , \quad i = 1, 2, \ldots, p \ . \tag{5}$$

Assuming that speech samples outside the frame of interest are zero and defining the autocorrelation function as

$$r(\tau) = \sum_{i=0}^{N-1-\tau} s(i)s(i+\tau) \ , \tag{6}$$

where N is the number of samples in a frame, the autocorrelation method yields the Yule-Walker equations given by [7]

$$\begin{bmatrix} r(0) & r(1) & r(2) & \ldots & r(p-1) \\ r(1) & r(0) & r(1) & \ldots & r(p-2) \\ r(2) & r(1) & r(0) & \ldots & r(p-3) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r(p-1) & r(p-2) & r(p-3) & \ldots & r(0) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_p \end{bmatrix} = \begin{bmatrix} r(1) \\ r(2) \\ r(3) \\ \vdots \\ r(p) \end{bmatrix} \ . \tag{7}$$

Since the matrix is Toeplitz, a computationally efficient algorithm known as the Levinson-Durbin recursion can be used to find the predictor coefficients. During this iterative procedure also other useful features known as reflection coefficients RC are found.

3.2 Cepstral Parameters

Cepstrum, defined as the inverse Fourier transform of the log of the signal spectrum, is an important spectral representation for speech and speaker recognition. It can be calculated from LPC coefficients or from the filter-bank spectrum. In the first case it is known as the LPC based cepstral coefficients LPCC. In the latter case as a mel frequency cepstral coefficients MFCC.

LPCC parameters can be calculated from the transfer function of the vocal tract $H(z)$ in (1) which requires calculating poles of the $H(z)$, or more computationally efficient recursion formula is used [7]

$$c_{\mathrm{lp}}(n) = \begin{cases} a_n + \sum\limits_{k=0}^{n-1} \frac{k}{n} a_{n-k} c_{\mathrm{lp}}(k), & 1 \leq n \leq p \ , \\ \sum\limits_{k=n-p}^{n-1} \frac{k}{n} a_{n-k} c_{\mathrm{lp}}(k), & n > p \ . \end{cases} \tag{8}$$

The LPC based cepstrum has many interesting properties. It is causal for the minimum phase $H(z)$ and of infinite duration. As the cepstrum represents the log of the signal spectrum, signals represented as the cascade of two effects which are products in the spectral domain are additive in the cepstral domain. This property of separability of pitch excitation and vocal tract is considered as one of the reasons that cepstral parameters are more effective for speaker recognition than other representations of speech signal. Another interesting property is the fact that $c_{\mathrm{lp}}(n)$ decays as fast as $1/n$ as n approaches $+\infty$ so the feature vector consists of the finite number, most significant components $c_{\mathrm{lp}}(1)$ to $c_{\mathrm{lp}}(x)$, where $x \approx 1.5p$.

MFCC parameters are based on the nonlinear human perception of the frequency of sounds. They can be computed as follows: window the signal, take the FFT, take the magnitude, take the log, warp the frequency according to the mel scale and finally take the inverse FFT. Mel warping transforms the frequency scale to place less emphasis on high frequencies [8].

4 Pattern Recognition

Since speaker recognition is based on similarity calculation between test utterance and the reference model, it is obvious that the problem of construction of the good model is crucial. The simplest approach to this problem but also the most computationally demanding during recognition process is to store in a memory all feature vectors extracted from the speech during training of the system. As speech is a very redundant signal, it can be easily seen that for text-independent recognition, which requires providing a lot of training speech to find a speaker model, it consists of thousands of multidimensional vectors. Computational efficiency of such model is very low. In order to compute distance between such model and vectors extracted from the test speech it is necessary to find for each test vector the most similar vector, known as a nearest neighbor NN, from the model.

Another method used for representing speaker in a speaker recognition system is based on vector quantization VQ. Speaker is represented as a set of several (less than 100) vectors that possibly in the best way represent speaker. This set of vectors is called a codebook. In this case during recognition each test vector is compared with its nearest neighbour from the codebook and the overall distance for the whole test utterance is computed. Calculation of normalized distance D for M frames of speech is given by

$$D = \frac{1}{M} \sum_{i=1}^{M} \min(d(x_i, c_q)) \quad 1 \leq q \leq L \ , \tag{9}$$

where x_i is a test vector and c_q a code vector from a codebook of size L. As it can be seen for M frames and L code vectors its necessary to calculate ML distances. The most often used measure of similarity is an Euclidean distance

$$d(x_i, c_q) = \sum_{k=1}^{p} (x_i(k) - c_q(k))^2 \tag{10}$$

where p is a dimension of a vector. VQ method is faster than NN technique, but unfortunately requires to find a codebook for each speaker, whereas in NN method a model consists just of all vectors. How to find the best codebook for speaker from a lot of training data? To solve this problem a kind of clustering technique is required, which can find a small set of the best representative vectors of a speaker. One of applied algorithms are k-means and Linde Buzo Gray procedure.

K-means algorithm is an iterative procedure and consists of four major steps. At first arbitrarily choose L vectors from the training data, next for each training vector find its NN from the current codebook, which corresponds to partitioning vector space into L distinct regions. The third step requires updating the code vectors using the centroid of the training vectors assigned to them and the last step – repeat steps 2 and 3 until some converge criterion is satisfied. The converge criterion is usually an average quantization error expressed in the same way as in (9) with an exception that x_i is a training vector.

Although k-means training method works well, it is even better to design a codebook in steps by using a splitting procedure, which leads to LBG algorithm. It starts with one cluster, which is the centroid of all training vectors, and then the code vector is split into two, $c_0 + \delta$ and $c_0 - \delta$, where δ is a small perturbation vector. With these two clusters k-means procedure is run. After the averaged distortion reaches steady level, the codebook is split again and the new codebook is trained with k-means method. This splitting is repeated until the desired codebook size is reached.

5 Speaker Verification in Matlab

All research was done on Polish database ROBOT [9]. This database consists of 2 CD with 1 GB of speech data. The speech utterances were collected from

30 speakers of both sex in a several time-separated sessions to catch intraspeaker variability. Main specifications of ROBOT are the following: sampling frequency 22 kHz, language – Polish, quantization 16 bit, file format ".wav", lack of files compression, recording environment – quiet, each file is preceded and followed by the silence. Recorded utterances consist of the words belonging to three dictionaries (L1, L2, and L3). Words in L1 and L3 are numbers from 0 to 9 and 10 to 99 respectively. Dictionary L2 consists only from commands used in robot control (start, stop, left, right, up, down, drop, catch, angle). These dictionaries were used to construct seven different sets of utterances Z1...Z7.

During training and testing of the speaker verification system the same signal processing procedure was used. Speech files, before feature extraction, were processed to remove silence. Voice activity detection was based on the energy of the signal. Next signal was preemphasized with a standard parameter of

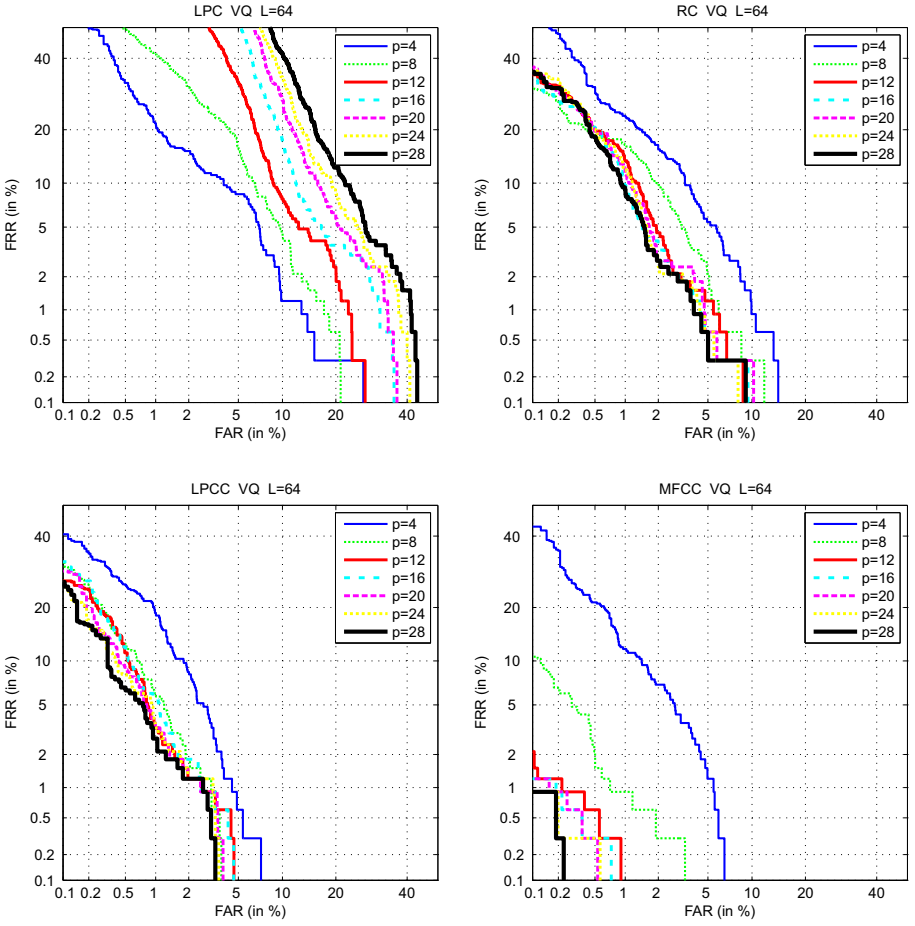

Fig. 3. Influence of feature dimensionality on speaker verification

$\alpha = 0.95$ and segmented into 10 ms frames every 5 ms. Hamming windowing was applied. For each frame LPC analysis was applied to obtain LPC and RC coefficients. LPC parameters were then transformed into LPCC coefficients using Equation (8). From each frame MFCC parameters were also computed. All utterances from Z3 set were used to obtain model of each speaker. Each model was constructed from approximately 90 s of speech after silence removing. Text dependent speaker verification was implemented. All test utterances were from Z4 (combination of numbers from Z3 set). Each speaker provided 11 test sequences of approximately 5 s each. As a result there were 9900 verification trials – 330 valid trails (30*11) and 9570 impostor trials (30*11*29) for each combination of dimensionality of the feature vector and the size of the speaker model.

In order to check the influence of dimensionality, testing was done for the prediction order p equal to 4, 8, 12, 16, 20, 24, and 28. Actual dimensionality

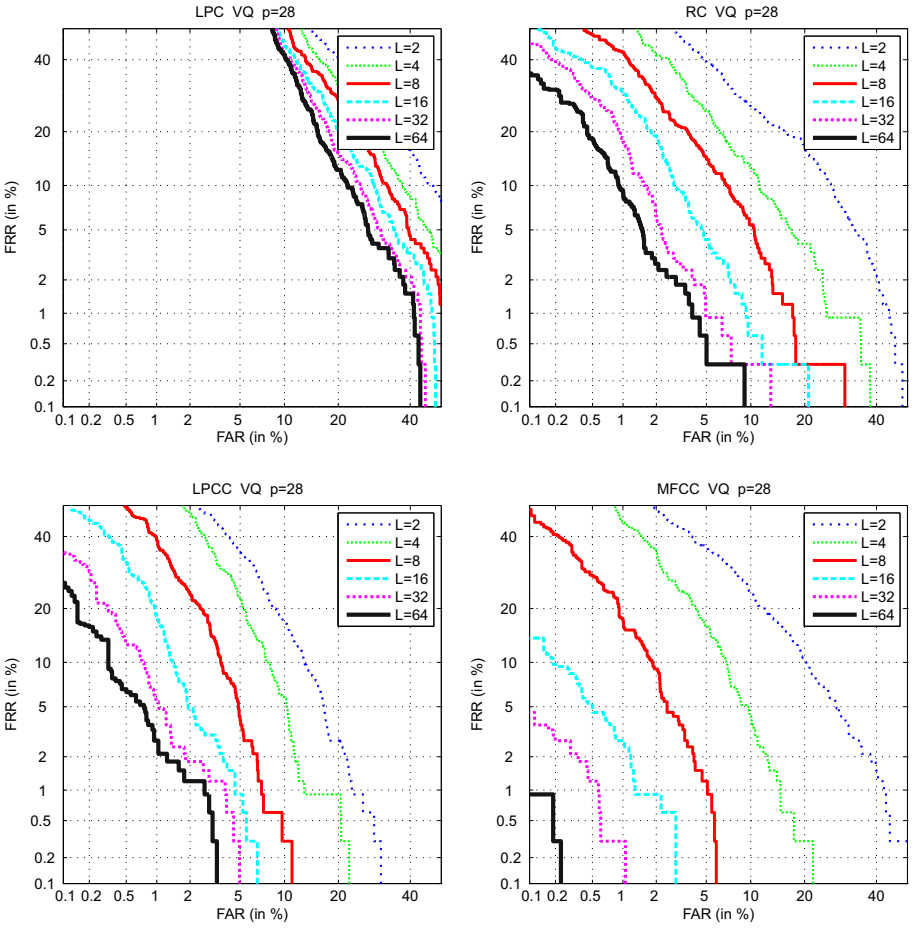

Fig. 4. Influence of model complexity on speaker verification

for the LPC and RC feature vectors were equal to p but for cepstral parameters (LPCC and MFCC) was $1.5p$ (6 to 42). LBG procedure was applied to obtain codebooks for each speaker. In order to check the influence of model size L, testing was done for the codebooks consisting of 2, 4, 8, 16, 32 and 64 code vectors.

Verification performance was characterized in terms of the two error measures, namely the false acceptance rate FAR and false rejection rate FRR. These measures correspond to the probability of acceptance an impostor as a valid user and the probability of rejection of a valid user. Changing the decision level, DET curves which show dependence between FRR and FAR can be plotted. Another very useful performance measure is an equal error rate EER which corresponds to error rate achieved for the decision threshold for which FRR=FAR. In other words EER is just given by the intersection point of the main diagonal of DET plot with DET curves.

Achieved results for the most complex model ($L = 64$ code vectors per speaker) as a function of dimensionality of the feature vectors p were shown in Fig. 3. Achieved results for the highest dimensionality ($p = 28$ parameters extracted from each segment of the utterance) as a function of model complexity L were shown in Fig. 4. The best achieved results of speaker verification were shown in Fig. 5.

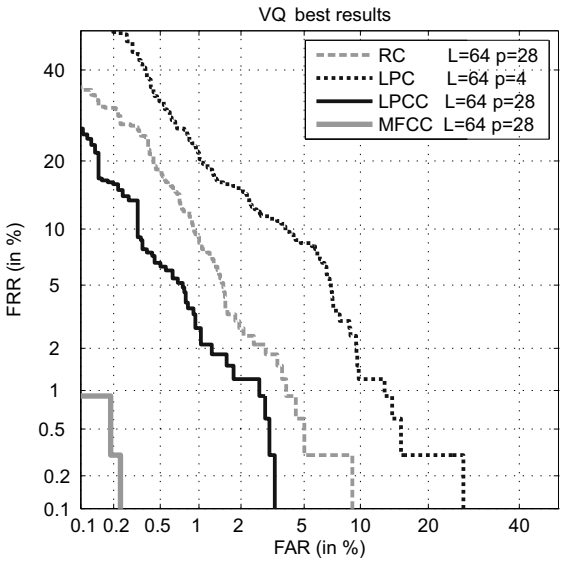

Fig. 5. The best achieved results for the speaker verification

6 Conclusion

The lowest FRR and FAR errors were achieved for the MFCC parameters (Fig. 5) For the best combination of dimensionality and complexity of the model ($p = 28$

and $L = 64$) achieved $EER = 0.27\,\%$ is definitely better than $EER = 1.55\,\%$ for LPCC, $EER = 2.42\,\%$ for RC and $EER = 6.43\,\%$ for LPC features. Such low error rate indicates that for limited number of speakers and high quality of speech, speaker verification may be implemented as an additional level in security systems or which is the final goal of this research may be implemented in a mobile phone. What is interesting is the fact that there is no need to increase dimensionality of feature vectors above some level. From the Fig. 3 optimum value of p may be estimated as between 12 and 20 for the MFCC and LPCC. Much more important than dimensionality is complexity of the model which was shown in the Fig. 4. Error rates are highly dependent on the number of code vectors per speaker model. Summarizing if enough learning data is available, the more code vectors per speaker the better.

Acknowledgment. This work was supported by The National Centre for Research and Development (www.ncbir.gov.pl) under Grant number POIG.01.03.01-24-107/12 *(Innovative speaker recognition methodology for communications network safety).*

References

1. Kłosowski, P.: Speech processing application based on phonetics and phonology of the polish language. In: Kwiecień, A., Gaj, P., Stera, P. (eds.) CN 2010. CCIS, vol. 79, pp. 236–244. Springer, Heidelberg (2010)
2. Dustor, A.: Voice verification based on nonlinear Ho-Kashyap classifier. In: International Conference on Computational Technologies in Electrical and Electronics Engineering, SIBIRCON 2008, Novosibirsk, pp. 296–300 (2008)
3. Dustor, A.: Speaker verification based on fuzzy classifier. In: Cyran, K.A., Kozielski, S., Peters, J.F., Stańczyk, U., Wakulicz-Deja, A. (eds.) Man-Machine Interactions. AISC, vol. 59, pp. 389–397. Springer, Heidelberg (2009)
4. Dustor, A., Kłosowski, P.: Biometric voice identification based on fuzzy kernel classifier. In: Kwiecień, A., Gaj, P., Stera, P. (eds.) CN 2013. CCIS, vol. 370, pp. 456–465. Springer, Heidelberg (2013)
5. Beigi, H.: Fundamentals of speaker recognition. Springer, New York (2011)
6. Togneri, R., Pullella, D.: An overview of speaker identification: Accuracy and robustness issues. IEEE Circuits and Systems Magazine 11(2), 23–61 (2011)
7. Rabiner, L.R., Juang, B.H.: Fundamentals of speech recognition. Prentice Hall (1993)
8. Fazel, A., Chakrabartty, S.: An overview of statistical pattern recognition techniques for speaker verification. IEEE Circuits and Systems Magazine 11(2), 62–81 (2011)
9. Adamczyk, B., Adamczyk, K., Trawiński, K.: Zasób mowy ROBOT. Biuletyn Instytutu Automatyki i Robotyki WAT 12, 179–192 (2000)