SVM Based GMM Supervector Speaker Recognition Using LP Residual Signal

Dalila Yessad and Abderrahmane Amrouche

Speech Communication and Signal Processing Laboratory,
Faculty of Electronics and Computer Sciences, USTHB,
P.O. Box 32, El Alia, Bab Ezzouar, 16111, Algiers, Algeria
yessad.dalila@gmail.com, namrouche@usthb.dz

Abstract. Feature extraction is an important step for speaker recognition systems. In this paper, we generated MFCC (Mel Frequency Cepstral Coefficients) and LPCC (Linear Predictive Cepstral Coefficients) from LP residual of speech signal, instead their calculation directly from speech samples. These features represent complementary vocal cord information's. In this work, Universal Background Gaussian Mixture Models (GMM-UBM) and Gaussian Supervector (GMM-SVM) based speaker modeling have been used. Experimental results, using, ARADIG-ITS data-base, show the efficiency of the GMM-SVM based approach associated with feature vectors issued from LP residual signal.

Keywords: LPC, LPCC, LP residual, MFCC, GMM-UBM, GMM-SVM, Speaker Recognition.

1 Introduction

Voiced speech is usually used for speaker recognition. But in text-independent speaker recognition it would be better to use special voiced phonems which are present in all words. In the source-filter model of human speech production, the speech signal is modeled as the convolutional output of a vocal source excitation signal and the impulse response of a vocal tract filter system [1]. Cepstral features [2] such as Mel-frequency cepstral coefficients (MFCCs) and linear predictive cepstral coefficients (LPCCs) have been the dominant features for a long time in speaker recognition. These features are believed to provide pertinent cues for phonetic classification and have been successfully implemented in most existing speaker recognition systems [3]. This indicates that MFCC and LPCC features capture properties of vocal tract and contain important speaker-specific information. Since MFCCs capture a mixture of phonemic and speaker-related information, their use has resulted in good performance in speaker recognition. In [4], the standard procedures for extracting MFCC and LPCC features were applied to LP residual signals, resulting in a set of residual features for speaker recognition. In linear predictive (LP) modeling of speech signals, the vocal tract system is represented by an all-pole filter. The prediction error, which is named the LP residual signal, contains useful information about the source excitation.

A. Elmoataz et al. (Eds.): ICISP 2012, LNCS 7340, pp. 579–586, 2012.

In [5], the speaker information present in LP residual signals was captured using an auto-associative neural network model and in [6] features extracted from linear predictive (LP) analysis were used. Despite these investigations, state-of-art systems are mostly based on the Mel cepstral frequency coding (MFCC) or the linear predictive cepstral coding (LPCC). Indeed, these short-term features have proven their efficiency in terms of performances and are adapted for the Gaussian mixture models (GMMs).

Current state of the art systems for text-independent speaker recognition use cepstral coefficients as base features, and speaker modeling techniques, such as Universal Backgrounds Gaussian Mixture Models (GMM-UBM) and Gaussian Supervector (GMM-SVM). These later are two successful approaches recently used. The first approach uses a speaker model which is formed by MAP adaptation of the means of the UBM. In the second approach, the GMM supervector is formed by stacking all mean vectors of the adapted model and is classified using a Support Vector Machines (SVM)[7], [8], [9].

This paper deals with the MFCC and LPCC feature extraction techniques based on LP residual signal. Section 2 provides feature extraction technique. Then, Sections 3 elaborates speaker modeling principles. In Section 4, we discuss the evaluation of speaker recognition performance, followed by conclusion in Section 5.

2 Feature Extraction

2.1 Linear Prediction (LP) Residual

Linear prediction (LP) is the process of predicting future sample values of a digital signal from a linear system. It is therefore about predicting the signal $x(n)$ at instant n from p previous samples as in Eq. (1)

$$x(n) = \sum_{i=1}^{p} a_i x_{n-i} + Ge(n) \tag{1}$$

Where $a_1, a_2, ..., a_p$ are the Linear Prediction Coefficients (LPCs), p is the model order, G and $e(n)$ are the excitation gain and source, respectively. The LPCs are derived adaptively for each 20-30 ms speech frame by minimization of excitation mean square energy. For simplicity, we will assume that the order of LP model is uneven, $p = 2m - 1$. The LPC spectrum or the transfer function of the LP filtering is defined by:

$$H(z) = \frac{G}{A(z)} \tag{2}$$

Where

$$A(z) = 1 - \sum_{i=1}^{2m-1} a_i z^{-1} \tag{3}$$

2.2 Cepstral Linear Prediction Coding (LPCC)

The cepstrum coefficients $\{ceps_q\}_{q=0}^{Q}$ can be estimated from the LPC coefficients $\{a_q\}_{q=1}^{p}$ using a recursion procedure:

$$ceps_q = \begin{cases} ln(G), & q = 0 \\ a_q + \sum_{k=1}^{q-1} \frac{k-q}{q} a_k ceps_{q-k}, & 1 \leqslant q \leqslant p \\ \sum_{k=1}^{p} \frac{k-q}{q} a_k ceps_{q-k}, & p < q \leqslant Q \end{cases} \tag{4}$$

Where G is the gain term in the LPC model, p the LPC model order, and $Q+1$ the number of cepstrum coefficients.

2.3 Mel Frequency Cepstral Coefficients (MFCC)

The most commonly used feature vector in speech recognition is composed of Mel-Frequency Cepstral Coefficients (MFCC). The MFCC extraction is done in three steps:

1. Step 1-a: Cut up the signal in several overlapping windows;
2. Step 1-b: To decrease the spectral distortion, a Hamming windowing is applied to signal frames;

$$W(n) = 0.54 - 0.46 * cos(\frac{2\pi n}{N-1}) \tag{5}$$

   Where N is the window size.

3. Step 2-a: Apply the FFT ;
4. Step 2-b: The Mel-frequency scale is applied using the following transformation formula;

$$Mel(f) = 2595 * log10(1 + \frac{f}{700}) \tag{6}$$

5. Step 2-c: Apply the logarithm after the Mel scale;
6. Step 3: Finally, obtain the discrete cosine transform (DCT) of the output signal.

3 Classifiers

3.1 Gaussian Mixture Model Universal Background (GMM-UBM)

The speaker recognition system is a Gaussian mixture model-universal background. The GMM-UBM approach is the state of the art system in text- independent speaker recognition [10]. This approach is based on a statistical modeling paradigm, where a hypothesis is modeled by a GMM model:

$$p(x|\lambda) = \sum_{i=1}^{i<m} \alpha_i N(x|\mu_i, \sum_i) \tag{7}$$

Where α_i, μ_i and $\sum_i$ respectively, the weights, the mean vectors, and the covariance matrices (generally diagonal) of the mixture components. During a test, the system has to determine whether the recording Y was pronounced by a given speaker S. This question is modeled by the likelihood ratio;

$$\frac{p(x|\lambda_{hyp})}{p(x|\lambda_{\overline{hyp}})} \geq \tau \tag{8}$$

Where Y is the test speech recording, λ_{hyp} is the model of the hypothesis where S pronounced Y, $\lambda_{\overline{hyp}}$ corresponds to the model of the negated hypothesis (S did not pronounce Y), $p(y|m)$ is the GMM likelihood function, and τ is the decision threshold. The model $\lambda_{\overline{hyp}}$ is a generic background model, the so-called UBM, and is usually trained during the development phase using a large set of recordings coming from a large set of speakers. The model λ_{hyp} is trained using a speech record obtained from the speaker S. It is generally derived from the UBM by moving only the mean parameters of the UBM, using a Bayesian adaptation function.

In this study The GMM-UBM system is the LIA SpkDet system [11] based on the ALIZE platform3 and distributed under an open source license. This system produces speaker models using MAP adaptation by adapting only the means from a UBM. The UBM component was trained on a selection of 60 corpus. For all the experiments, the model size is 128 and the performances are assessed using DET plots and measured in terms of equal error rate (EER) and minimum of detection cost (minDCF).

3.2 Support Vector Machines (SVM)

The support vector machine (SVM) [8] is a two-class classier constructed from sums of a kernel function $k(\cdot, \cdot)$,

$$f(x) = \sum_{i=1}^{L} \alpha_i t_i k(x, x_i) + d \tag{9}$$

Where t_i are the ideal outputs, $\sum_{i=1}^{L} \alpha_i t_i = 0$, $i = 0$ and $\alpha > 0$. The vectors x_i are support vectors and obtained from the training set by an optimization process [11]. The ideal output are either 1 or -1, depending upon whether the corresponding support vector is in class 0 or class 1, respectively. For classification, a class decision is based upon whether the value, $f(x)$, is above or below a threshold. The kernel $k(\cdot, \cdot)$ is constrained to have certain properties, so that $k(\cdot, \cdot)$ can be expressed as :

$$k(x, y) = b(x)^t b(y) \tag{10}$$

Where $b(x)$ is a mapping from the input space (where x lives). For a separable data set, SVM optimization chooses a hyperplane in the expansion space with maximum margin [7], [8]. The data points from the training set lying on the boundaries are the support vectors in equation (1). The focus of the SVM training process is to model the boundary between classes in [7], [8].

3.3 GMM Supervector (GMM-SVM)

Gaussian mixture models with universal backgrounds is constructed by MAP adaptation of the means of the UBM. A GMM supervector is constructed by stacking the means of the adapted mixture components. We assume we are given a Gaussian mixture model universal background model (GMM-UBM):

$$p(x|\lambda) = \sum_{i=1}^{i<m} \alpha_i N(x|\mu_i, \sum_i) \tag{11}$$

Where α_i are the mixture weights, m indicates a Gaussian density and , μ_i and $\sum_i$ are the corresponding mean and covariance. From a speaker utterance, the GMM-UBM model is adapted by Maximum A Posteriori (MAP) adaptation to provide the speaker GMM model. Generally, only the means ${}^1i$ of Gaussian components are adapted. In this case, all GMMs have the same covariance matrices $\sum_i$ and differ only in means. As a consequence, for SVM classification, each model is represented only by the concatenation of all GMM Gaussians mean vectors, that is, a GMM supervector [12].

4 Results and Discussions

4.1 Speech Database and Features Extraction

Arabic digits, which are polysyllabic, can be considered as representative elements of language, because more than half of the phonemes of the Arabic language are included in the ten digits. The speech database used in this paper is a part of the database ARADIGITS [13]. It consists of a set of 10 digits of the Arabic language (zero to nine) spoken by 60 speakers of both genders with three repetitions for each digit. This database was recorded by Algerian speakers from different regions aged between 18 and 50 years in a quiet environment with an ambient noise level below 35 dB, in WAV format, with a sampling frequency equal to 16 kHz. In this work we used the "long training / short test" for speaker recognition on ARADIGITS. The features corresponding to the six digits (from zero to five, with concatenation of three repetitions) are used for training each speaker model. Only 60 speakers of the database are used in the speaker identification system for testing. Four digits(from six to nine) of every speaker is tested separately (60x4=240 test patterns of seconds each, in average). The experiments are totally text independent. Speaker utterances were represented by 19 coefficients LPCC or MFCC, with their first derivatives and the delta energy. Altogether, a 40 coefficients vector is extracted from each LP residual, based speech signal frame. Mean subtraction

and variance normalization were applied to all features. Figure 1 shows the speech waveforms and the corresponding LP residual signals, of the vowel /a/ from the sound /wahid/ uttered by two different female speakers. We can see the differences between the two segments of residual signals. In addition to the difference between their pitch periods, the residual signal of speaker A shows much stronger periodicity than that of speaker B.

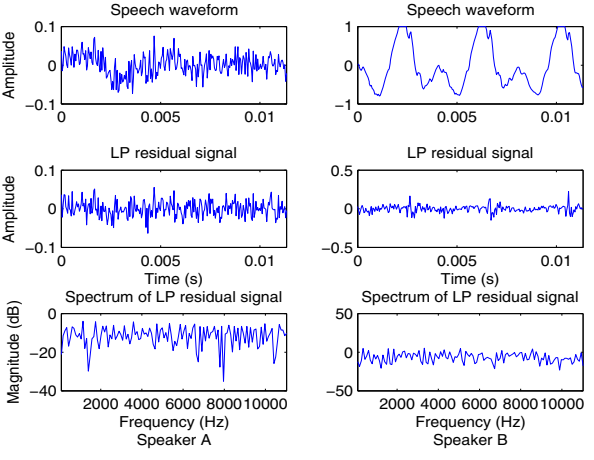

Fig. 1. Speech waveform, LP residual signals and Fourier spectra of LP residual signal of two female speakers; Speaker A in the left and speaker B in the right

4.2 Experimental Results

We evaluate the speaker recognition performances of MFCC and LPCC individually like baseline system, using both GMM-UBM and GMM-SVM classifiers. In addition, we evaluate these classifiers with MFCC or LPCC extracted from LP residual signal, using the same evaluation database. The EER performance of the baseline system are shown in Figure 2. The best performance are obtained with MFCC based GMM-SVM, 91% in average. Figure 3 shows the recognition performance of MFCC and LPCC features extracted from LP residual signal. The MFCC extracted from LP residual and based GMM-SVM achieved the best performance (it is found at 88% in average). Experimental results show that the GMM-SVM using MFCC features gives the best performances, and MFCC features outperform the MFCC extracted from LP residual, because in frequency domain, the useful temporal information, the amplitudes and the time locations of pitch pulses, are not represented in the Fourier spectra of LP residual. To characterize the time-frequency characteristics of the pitch pulses, others transformations like wavelet are more appropriate than the short-time Fourier transform.

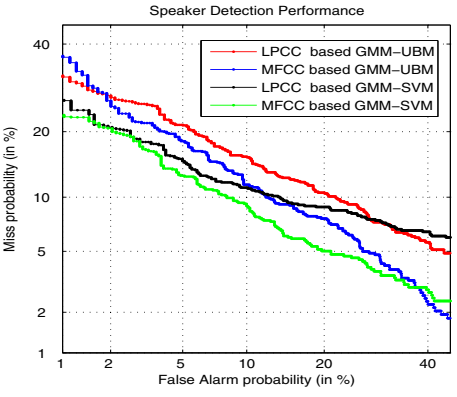

Fig. 2. The performance of GMM-UBM and GMM-SVM systems with MFCC and LPCC features extracted from ARADIGITS database

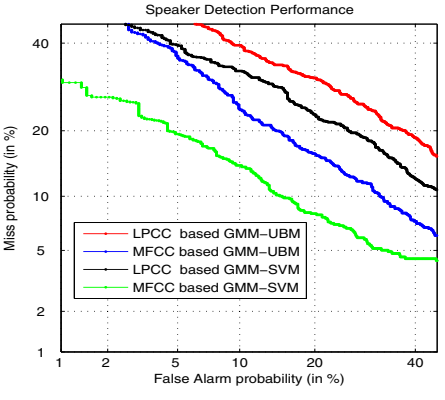

Fig. 3. The performance of GMM-UBM and GMM-SVM systems with MFCC and LPCC features extracted from LP residual signal

5 Conclusion

This paper investigates MFCC and LPCC features extraction from LP residual signal based on both GMM-UBM and GMM-SVM classifiers. We have shown that the MFCC and LPCC features based LP residual contain speaker-specific information for speaker recognition applications, and the MFCC features provide additional information in speaker recognition. This work shows the possibility of performing speaker recognition by extracting features directly from LP residual signal.

References

1. Rabiner, L.R., Schafer, R.W.: Digital Processing of Speech Signals. Prentice-Hall, Englewood Cliffs (1978)
2. Quatieri, T.F.: Discrete-Time Speech Signal Processing - Principles and Practice. Prentice-Hall (2002)
3. Reynolds, D.A.: An overview of automatic speaker recognition technology. In: Proc. Int. Conf. Acoust., Speech, and Signal Process. (ICASSP), pp. 4072–4075 (2002)
4. Chen, S.H., Wang, H.C.: Improvement of speaker recognition by combining residual and prosodic features with acoustic features. In: Proc. Int. Conf. Acoust., Speech, Signal Process. (ICASSP), pp. 93–96 (2004)
5. Yegnanarayana, B., Reddy, K.S., Kishore, S.P.: Source and system features for speaker recognition using AANN models. In: Proc. Int. Conf. Acoust.,Speech, Signal Process. (ICASSP), pp. 409–413 (2001)
6. Mary, L., Sri Rama Murty, K., Mahadeva Prasanna, S.R., Yegnanarayana, B.: Features for speaker and language identification. In: Proc. of the ISCA Tutorial and Research Workshop on Speaker and Language Recognition (Odyssey 2004), pp. 323–328 (2004)
7. Dong, X., Zhaohui, W.: Speaker Recognition using Continuous Density Support Vector Machines. Electronics Letters 37(17), 1099–1101 (2001)
8. Wan, V., Renals, S.: SVMSVM: Support Vector Machine Speaker Verification Methodology. In: Proc. Int. Conf. Acoust., Speech, Signal Process. (ICASSP), Hong Kong, vol. 2, pp. 221–224 (2003)
9. Karam, Z.N., Campbell, W.M.: A Multi-Class MLLR Kernel for SVM Speaker Recognition. In: Proc. Int. Conf. Acoust., Speech, Signal Process. (ICASSP), pp. 4117–4120 (April 2008)
10. Reynolds, D.A., Quatieri, T.F., Dunn, R.B.: Speaker Verification Using Adapted Gaussian Mixture Models. Digital Signal Processing 10(1-3), 19–41 (2000)
11. `http://www.lia.univ-avignon.fr/heberges/ALIZE/LIA/~RAL`
12. 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)
13. Amrouche, A., Debyeche, M., Taleb Ahmed, A., Rouvaen, J.M., Ygoub, M.C.E.: Efficient System for Speech Recognition in Adverse Conditions Using Nonparametric Regression. Engineering Applications on Artificial Intelligence 23(1), 85–94 (2010)