SPEAKER RECOGNITION VIA SPARSE REPRESENTATIONS USING ORTHOGONAL MATCHING PURSUIT

Vivek Boominathan and K Sri Rama Murty

Department of Electrical Engineering
Indian Institute of Technology Hyderabad, Hyderabad - 502205, India
vivek@iith.ac.in, ksrm@iith.ac.in

ABSTRACT

The objective of this paper is to demonstrate the effectiveness of sparse representation techniques for speaker recognition. In this approach, each feature vector from unknown utterance is expressed as linear weighted sum of a dictionary of feature vectors belonging to many speakers. The weights associated with feature vectors in the dictionary are evaluated using orthogonal matching pursuit algorithm, which is a greedy approximation to l_0 optimization. The weights thus obtained exhibit high level of sparsity, and only a few of them will have nonzero values. The feature vectors which belong to the correct speaker carry significant weights. The proposed method gives an equal error rate (EER) of 10.84% on NIST-2003 database, whereas the existing GMM-UBM system gives an EER of 9.67%. By combining evidence from both the systems an EER of 8.15% is achieved, indicating that both the systems carry complimentary information.

Index Terms— Sparse representation, orthogonal matching pursuit, l_0 optimization and Gaussian mixture modeling.

1. INTRODUCTION

Speaker recognition refers to recognizing persons from their voice [1]. A speaker recognition system can be operated in either identification mode or verification mode. In speaker identification, the goal is to identify the speaker of an utterance from a given population, whereas speaker verification involves validating the identity claim of a person. Speaker recognition systems can be categorized into: text-dependent systems, and text-independent systems. Text-dependent systems require recitation of a predetermined text, whereas text-independent systems accept speech utterances of unrestricted text. This work deals with text-independent speaker verification.

An automatic speaker recognition system typically comprises of three stages: feature extraction, feature matching and score computation. Depending on how the feature matching is done, speaker recognition systems can be classified into template-matching systems and probabilistic modeling systems [2], also known as nonparametric and parametric systems, respectively. In template-matching systems, the feature vectors from the training and testing utterances are compared directly, with the assumption that either of them is an imperfect replica of the other. Dynamic time warping is an example of template-matching method for text-dependent speaker recognition [3]. Dynamic time warping can not be applied to text-independent speaker recognition, due to the lack of temporal alignment between the *sequence* of feature vectors from training and testing utterances. Most of the text-independent speaker recognition systems rely on the probabilistic modeling of the *set* of feature vectors.

In probabilistic modeling, the set of feature vectors from each speaker is modeled with a fixed but unknown probability density function [4]. Matching is done by evaluating the likelihood of the testing utterance with respect to the speaker model. The unknown probability density function is usually approximated as a linear combination of Gaussian density functions, and is popular as Gaussian mixture modeling (GMM). The parameters, i.e mean vectors covariance matrices and weights, of the GMM are evaluated iteratively from the training data using maximum-likelihood estimation [5]. Since this method involves estimation of several parameters, it typically requires large amount of training data to capture the variability due to different environments, channels, speaking styles and so on. In order to reduce the data requirements during the training phase, Reynolds et. al., have introduced the concept of universal background modeling (UBM) [6]. UBM is essentially a very large GMM trained to represent the speaker independent distribution of the speech features gathered from a large number of speakers under different environments. When enrolling a new speaker to the system, the parameters of the background model are adapted to the feature distribution of the new speaker. The adapted model is used as the model of that speaker. In this approach, the model parameters are not estimated from scratch, instead prior knowledge about the distribution of the features in the acoustic space is being utilized. As a consequence, this method not only reduces the data requirements during training but also provides normalization across the speaker models as all of them are adapted from the same UBM.

In this work, we propose a template-matching approach for text-independent speaker recognition using sparse representations. Sparse representations have been used in image processing for face recognition [7], and iris recognition [8]. Sparse representation of GMM-supervectors was used for speaker identification in [9] and [10]. In both these approaches, sparse representation is used for classification of GMM-super vectors. Hence, these approaches require probabilistic modeling of the features before invoking the sparse representations. In this paper, we use sparse representations for directly matching feature vectors from the training and testing utterances. This approach does not require probabilistic modeling.

The main assumption behind the proposed approach is that the testing template lies approximately in the linear span of the training templates. Here the word template refers to a feature vector. That is, the testing template can be approximated as linear weighted sum of a few training templates. We have employed orthogonal matching pursuit (OMP) algorithm to identify which of the training templates contribute to the representation of a given testing template. The degree of contribution of different speakers in representing the testing template can be used to recognize the correct speaker.

The rest of this paper is organized as follows: The theory of

sparse representations and the application of OMP algorithm for speaker recognition are discussed in Section 2. Experimental evaluation of the proposed method on NIST-2003 database, and its comparison with GMM-UBM system is presented in Section 3. Finally in Section 4, we summarize the important contributions of this work, and indicate possible directions for future studies.

2. SPARSE REPRESENTATIONS USING OMP

Suppose that there are K speakers, and each speaker has a set of N frames extracted from his reference utterance[1]. Let a d-dimensional feature vector be extracted from each frame. Let

$$\mathbf{A}_k = [\mathbf{a}_{k1}\mathbf{a}_{k2}\dots\mathbf{a}_{kn}\dots\mathbf{a}_{kN}] \in R^{d\times N}$$

be a $d \times N$ matrix of feature vectors of the k^{th} speaker, where the column $\mathbf{a}_{kn} = [a_{kn_1}a_{kn_2}\dots a_{kn_d}]^T$ denotes the d-dimensional feature vector of the n^{th} frame belonging to the k^{th} speaker. A dictionary $\mathbf{A}$ can be defined as concatenation of feature vectors of all the K speakers, as follows:

$$\begin{aligned}\mathbf{A} &= [\mathbf{A}_1\mathbf{A}_2\dots\mathbf{A}_k\dots\mathbf{A}_K] \in R^{d\times KN} \\ &= [\mathbf{a}_{11}\dots\mathbf{a}_{1N}|\mathbf{a}_{21}\dots\mathbf{a}_{2N}|\dots|\mathbf{a}_{k1}\dots\mathbf{a}_{kN}|\dots|\mathbf{a}_{K1}\dots\mathbf{a}_{KN}]\end{aligned} \quad (1)$$

Let us consider that an observed feature vector $\mathbf{y} \in R^d$, extracted from a testing utterance, be expressed as a linear weighted sum of columns of dictionary $\mathbf{A}$ as

$$\mathbf{y} = \sum_{k=1}^{K}\sum_{n=1}^{N} x_{kn}\mathbf{a}_{kn}$$

where the scalar x_{kn} is the wight associated with the column $\mathbf{a}_{kn}$. The above equation can be compactly written in the matrix form as

$$\mathbf{y} = \mathbf{A}\mathbf{x}, \quad (2)$$

where

$$\mathbf{x} = [x_{11}\dots x_{1N}|x_{21}\dots x_{2N}|\dots|x_{k1}\dots x_{kN}|\dots|x_{K1}\dots x_{KN}]^T.$$

If the observation vector $\mathbf{y}$ belongs to the k^{th} speaker, then it lies approximately in the linear span of the training vectors of that speaker. This implies that the coefficients $\mathbf{x}$ that are not associated with the k^{th} speaker should ideally be close to zero. As a result, the weight vector $\mathbf{x}$ exhibits high level of sparsity with very few nonzero coefficients.

In order to represent $\mathbf{y}$ as a linear combination of columns $\mathbf{A}$, we need to solve the system of linear equations in (2). Since the dimensionality of the feature vector (d) is much smaller than the number of feature vectors in the dictionary ($K.N$), the system of linear equations in (2) is under-determined, and does not admit a unique solution. Out of the infinitely many solutions available in the solution space, we need to search for the sparsest solution. The sparsest solution can be obtained by solving the following optimization problem

$$\min_{\mathbf{x}} ||\mathbf{x}||_0 \qquad \text{subject to} \qquad \mathbf{y} = \mathbf{A}\mathbf{x} \quad (3)$$

where $||\mathbf{x}||_0$ refers to zero norm of $\mathbf{x}$ which denotes the number of nonzero coefficients in $\mathbf{x}$. This combinatorial optimization problem is NP-hard to solve, and several iterative algorithms like matching pursuit (MP), and orthogonal matching pursuit (OMP) were proposed to address it [11]. Attempts were also made to convexify the objective function in (3) by replacing the l_0 norm with the l_1 norm. The resulting optimization problem involves minimization of l_1 norm, and can be solved using linear programming techniques in polynomial complexity [11].

In this work, we have opted to the OMP algorithm to obtain an approximate sparse weight vector $\hat{\mathbf{x}}$ [12], because of its simplicity. In order to identify a sparse weight vector $\hat{\mathbf{x}}$, we need to determine which columns of $\mathbf{A}$ participate in the representation of $\mathbf{y}$. The basic idea behind the OMP algorithm is to pick those representative columns in a greedy fashion [13]. To start with $\hat{\mathbf{x}}$ is initialized to $\mathbf{0}$. At each iteration we choose one column of $\mathbf{A}$ that is most strongly correlated with the residual,

$$\mathbf{r}(\mathbf{y}) = \mathbf{y} - \mathbf{A}\hat{\mathbf{x}}.$$

Then the corresponding coefficient of $\hat{\mathbf{x}}$ is updated, and procedure is repeated on the new residual. This procedure is continued till the residual error

$$r(\mathbf{y}) = ||\mathbf{r}(\mathbf{y})||_2 = ||\mathbf{y} - \mathbf{A}\hat{\mathbf{x}}||_2$$

goes below a predefined error threshold θ. Since the residual error depends on $||\mathbf{y}||_2$, a fraction of $||\mathbf{y}||_2$ can be used as error threshold θ. That is

$$\theta = \lambda||\mathbf{y}||_2,$$

where $0 < \lambda < 1$. Notice that a high value of λ may result in only a gross representation of the observed vector, and may not capture the speaker-specific characteristics. On the other hand, a low value of λ may spoil the sparsity of the weight vector $\hat{\mathbf{x}}$ while trying to minimize the residual error, though the error might be due to noise in the measurements. A detailed description of OMP and a step-wise algorithm to implement it can be found in Chapter 3 of [11]

2.1. Evaluation of confidence score

Given an observation vector $\mathbf{y}$, extracted from a frame of the testing utterance, we find the sparse weight vector $\hat{\mathbf{x}}$ by solving (2) using OMP algorithm. Speaker recognition is performed based on the fact that most of the significant coefficients of the weight vector $\hat{\mathbf{x}}$ would belong to the genuine speaker. The contribution of individual speakers in the dictionary $\mathbf{A}$ in representing $\mathbf{y}$ can be quantified in terms of speaker-specific residual error. The speaker-specific residual error of the k^{th} speaker is computed by retaining the weights associated with that speaker, and setting the weights associated with other speakers to zero. This can be done by introducing a mask function $\Pi_k(\hat{\mathbf{x}})$ which selects the weights associated with k^{th} speaker as follows:

$$\Pi_k(\hat{\mathbf{x}}) = [0\dots0|0\dots0|\dots|\hat{x_{k1}}\dots\hat{x_{kN}}|\dots|0\dots0]^T$$

The speaker-specific residual error of the k^{th} speaker can be evaluated as

$$r_k(\mathbf{y}) = ||\mathbf{y} - \mathbf{A}\Pi_k(\hat{\mathbf{x}})||_2 \quad (4)$$

Since the residual error depends on $||\mathbf{y}||_2$, it is divided by $||\mathbf{y}||_2$ in order to obtain normalized residual error,

$$\tilde{r_k}(\mathbf{y}) = \frac{||\mathbf{y} - \mathbf{A}\Pi_k(\mathbf{x})||_2}{||\mathbf{y}||_2} \quad (5)$$

The normalized residual error always lies between 0 and 1, and is converted into a confidence score by exponentiating it as follows,

$$s_k(\mathbf{y}) = \exp(-\tilde{r_k}(\mathbf{y})), \quad (6)$$

[1] Note that the number of frames per speaker differs based on the duration of training utterance. Here, we have assumed same number (N) of frames for simplifying the presentation. This method can be applied even if the number of frames per speaker differs.

where $s_k(\mathbf{y})$ is the confidence score for a single frame in the testing utterance. If the testing utterance contains L frames ($\mathbf{y}_l, l = 1, 2, \ldots, L$), the above procedure is repeated for each of the L frames, and the mean confidence score

$$S_k = \frac{1}{L} \sum_{1}^{L} s_k(\mathbf{y}_l) \tag{7}$$

is used for making a decision with respect to k^{th} speaker. This procedure is repeated for all the K speakers in the dictionary $\mathbf{A}$ to evaluate their respective confidence scores.

3. EXPERIMENTAL EVALUATION

3.1. Database for the study

The speaker recognition system proposed in this paper is evaluated on NIST - 2003 speaker recognition evaluation corpus [14]. We have considered only male speaker trials for this evaluation. There are 149 male speakers, and the duration of the training utterance for each speaker is about 2 minutes. There are 1343 testing utterances, each having a duration of 15 - 45 seconds. Each testing utterance has 11 claimants, where the genuine speaker may or may not be present. All the speech signals are collected over telephone channels, and are sampled at 8 kHz.

3.2. Feature extraction

In this study, we have used Mel-frequency cepstral coefficients (MFCC) as features to represent the speaker-specific characteristics. The speech signal is preemphasized, and is divided into frames of 20 ms duration with an overlap of 10 ms. Each frame is multiplied by a hamming window, and a 19-dimensional MFCC vector is extracted using 26 Mel-scaled triangular filters in the telephone bandwidth ranging from 300 Hz to 3300 Hz. Delta-cepstral coefficients computed over a span of ± 2 frames are appended to the MFCCs producing a 38-dimensional feature vector. An energy based speech detector is applied to discard feature vectors from low-energy frames. Cepstral mean subtraction is performed to mitigate the channel effects.

3.3. Speaker recognition using OMP

In NIST-2003 speaker recognition evaluation, each test utterance has 11 claimants. The 38-dimensional MFCC vectors from all the 11 claimants are augmented to form a dictionary $\mathbf{A}$, as in (1) with $K = 11$. Each MFCC vector $\mathbf{y}_l$, $l = 1, 2, \ldots, L$, from test utterance is represented as a linear weighted sum of MFCC vectors in the dictionary $\mathbf{A}$ as in (2), and the associated weight vector $\hat{\mathbf{x}}_l$ is obtained using OMP algorithm described in Section 2. The OMP algorithm is iterated till the residual error $r(\mathbf{y}_l)$ goes below $0.1||\mathbf{y}_l||_2$. The weight vector thus obtained is used to compute the confidence score (per frame) for each of the 11 claimants as in (6). The mean confidence score of all the MFCC vectors extracted from the test utterance is used for verifying the speaker's claim. The performance of the proposed speaker recognition system is given as the detection error tradeoff (DET) curve [15] in Fig. 1 (dashed curve). From the DET curve, the EER is found to be 10.84%. In this evaluation, we have not performed test utterance normalization on the confidence scores. The performance could be improved by implementing test utterance normalization using background data [16].

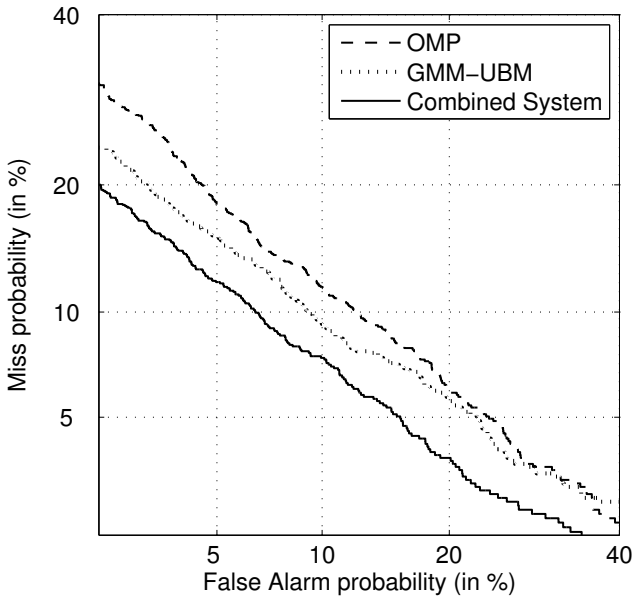

Fig. 1. DET curves for the proposed speaker recognition system, GMM-UBM system and combined system.

3.4. Speaker recognition using GMM-UBM

The performance proposed template-matching based method is compared with the state-of-the art probabilistic modeling method, i.e., GMM-based speaker recognition system. In this approach, speaker-specific GMM is obtained by adapting a UBM with 2048 mixtures. The UBM is trained using the male speaker trails of NIST-2002 development data. The parameters of the UBM are estimated using the maximum likelihood criterion, performing 20 iterations per mixture split. Speaker specific GMMs are obtained by adapting the UBM using classical maximum a-posteriori adaptation (MAP) with one iteration, and a relevance factor of 16 [6].

In the recognition mode, the MAP-adapted speaker model and the UBM model are coupled, and recognizer is commonly referred to as GMM-UBM. The confidence score is computed by subtracting the average log-likelihood of the test utterance with respect to the UBM from its average log-likelihood with respect to the speaker model. This subtraction helps in test utterance normalization, and makes the confidence scores across different test utterances comparable. The performance of the GMM-UBM system on NIST-2003 database is shown in Fig. 1(dotted curve). The EER for GMM-UBM system is found to be 9.67%, and is marginally better than the proposed system. Note that proposed system and GMM-UBM system are based on two different pattern matching paradigms, template-matching and probabilistic modeling, respectively. Hence, both these systems may carry complimentary information, and their combination may perform better.

3.5. Combination of speaker verification systems

The confidence scores C_{omp} and C_{gmm} obtained using OMP and GMM, respectively, are combined using the linear weighted sum

$$C = \alpha C_{omp} + (1 - \alpha) C_{gmm}.$$

The performance of the combined system is plotted as a DET curve in Fig. 1 (solid curve), for $\alpha = 0.5$. The EER of the combined sys-

tem is found to be 8.15%, which is better than either of the individual systems. This shows that the proposed template-matching approach is complimentary to the existing probabilistic-modeling approach.

4. SUMMARY AND CONCLUSION

In this paper, we have proposed a template-matching approach for text-independent speaker recognition. In text-independent speaker recognition, we can not directly compare the templates of training and testing utterances due to the lack of temporal alignment between them. In the proposed method, we have used sparse representations to identify a set of training templates, whose linear combination collectively represent a given testing template. The degree of contribution of training templates from different speakers in representing a given testing template can be used as a confidence measure for speaker recognition. The performance of the proposed template-matching system is comparable with the existing GMM-UBM method which is based on probabilistic modeling. However, the combined system performed better than either of the individual systems. This might be because these two systems are based on different pattern matching paradigms, and hence carry complementary information.

The proposed method does not involve any kind of probabilistic modeling, and hence does not involve estimation of parameters from the data. This method is based on providing a sparse representation for testing template in terms of training templates. This involves solving underdetermined system of linear equations for a sparse solution, or minimizing l_0 norm of the solution. We have used OMP algorithm for obtaining the sparse solution. It is observed that the performance of the OMP algorithm depends on the stopping criterion, i.e., the error threshold θ. We need to explore ways to adaptively set the error threshold to control the iterations of OMP algorithm. Though the performance of the OMP algorithm is satisfactory, it does not guarantee the most optimum solution. We need to explore methods like basis pursuit and lasso which are based on l_1 minimization for speaker recognition. Finally, the performance of the proposed method could be improved by implementing test utterance normalization. This can be done by augmenting the dictionary with known background data, and subtracting the background score from the confidence score of each speaker.

5. REFERENCES

[1] T. Kinnunen and Haizhou Li, "An overview of text-independent speaker recognition: From features to super vectors," *Speech Communication*, vol. 52, pp. 12–40, 2010.

[2] J. P. Campbell, "Speaker recognition: A tutorial," *Proc. IEEE*, vol. 85, no. 9, pp. 1437–1462, Sept. 1997.

[3] S. Furui, "Cepstral analysis technique for automatic speaker verification," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 29, pp. 254–272, Apr. 1981.

[4] D. A. Reynolds and R. C. Rose, "Robust text-independent speaker identification using gaussian mixture models," *IEEE Trans. Speech Audio Processing*, vol. 3, no. 1, pp. 72–83, Jan. 1995.

[5] D. A. Reynolds, "Speaker identification and verification using Gaussian mixture speaker models," *Speech Communication*, vol. 17, no. 1–2, pp. 91–108, 1995.

[6] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted Gaussian mixture models," *Digital Signal Processing*, vol. 10, no. 1–3, pp. 19–41, 2000.

[7] J. Wright, A. Y. Yang, A. Ganesh, and S. S. Sastry, "Robust face recognition via sparse representation," *IEEE Trans. Pattern Analysis, Machine Intelligence*, vol. 31, no. 2, pp. 210–227, 2009.

[8] J. K. Pillai, V. M. Patel, R. Chellappa, and N. K. Ratha, "Secure and robust iris recognition using random projections and sparse representations," *Accepted for publication in IEEE Trans. Pattern Analysis, Machine Intelligence*.

[9] Imran Naseem, Roberto Togneri, and Mohammed Bennamoun, "Sparse representation for speaker identification.," in *ICPR'10*, 2010, pp. 4460–4463.

[10] J. M. K. Kua, E. Ambikairajah, J. Epps, and R. Togneri, "Speaker verification using sparse representation classification," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Prague, Czech Republic, May 2011, pp. 4548–4551.

[11] Michael Elad, *Sparse and Redundant Representations*, Springer, New York, 2009.

[12] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition," in *Proceedings of the 27 th Annual Asilomar Conference on Signals, Systems, and Computers*, 1993, pp. 40–44.

[13] Joel A. Tropp and Anna C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Trans. Inform. Theory*, vol. 53, pp. 4655–4666, 2007.

[14] "NIST speaker recognition evaluation plan," in *Proc. NIST speaker recognition workshop*, University of Maryland, College Park, MD, USA, 2003.

[15] A. Martin, G. Doddington, T. Kamm, M. Ordowski, and M. Przybocki, "The DET curve in assessment of detection task performance," in *Eur. Conf. Speech Processing Technology*, Rhodes, Greece, Sept. 1997, pp. 1895–1898.

[16] R. Auckenthaler, M. Carey, and H. L. Thomas, "Score normalization for text-independent speaker verification systems," *Digital Signal Processing*, vol. 10, pp. 42–54, Jan. 2000.