World Academy of Science, Engineering and Technology
International Journal of Electrical and Computer Engineering
Vol:1, No:11, 2007

Improved Weighted Matching for Speaker Recognition

Ozan Mut, Mehmet Göktürk

***Abstract*—**Matching algorithms have significant importance in speaker recognition. Feature vectors of the unknown utterance are compared to feature vectors of the modeled speakers as a last step in speaker recognition. A similarity score is found for every model in the speaker database. Depending on the type of speaker recognition, these scores are used to determine the author of unknown speech samples. For speaker verification, similarity score is tested against a predefined threshold and either acceptance or rejection result is obtained. In the case of speaker identification, the result depends on whether the identification is open set or closed set. In closed set identification, the model that yields the best similarity score is accepted. In open set identification, the best score is tested against a threshold, so there is one more possible output satisfying the condition that the speaker is not one of the registered speakers in existing database. This paper focuses on closed set speaker identification using a modified version of a well known matching algorithm. The results of new matching algorithm indicated better performance on YOHO international speaker recognition database.

***Keywords*—** Automatic Speaker Recognition, Voice Recognition, Pattern Recognition, Digital Audio Signal Processing.

I. Introduction

SPEAKER recognition can be classified into two categories; Speaker Verification (SV) and Speaker Identification (SI) [4]. Speaker verification is the task of accepting or rejecting the identity of a speaker claimed to be someone. Speaker Identification is the task of finding the identity of an unknown speaker among a stored database of speakers. Speaker Identification can be done in closed-set or open-set forms. In closed-set form, the unknown speaker is definitely one of the speakers in the database. In open-set form on the other hand, the speaker may not belong to one of the registered speakers in the database, therefore an open-set identification system has one more possible output for rejection. Yet, there is another classification method for speaker recognition; that is, text-dependence. In Text-Dependent systems, the uttered word or phrase is known apriori to the system whereas in Text-Independent systems, as the name implies, utterance is not necessarily known to the system.

This work was supported in part by the Gebze Institute of Technology.

Ozan Mut is a Master of Science student at Gebze Institute of Technology (email: ozan.mut@hititbt.com).

Mehmet Göktürk is a faculty member at Gebze Institute of Technology (email: gokturk@gyte.edu.tr).

Speaker Identification process consists of two main phases; namely, Enrollment (Training) and Identification (Matching). In enrollment phase, all samples from the speakers are trained and stored in a database. The goal of training is to create a reference model for each speaker to be used in classification of unknown utterances in recognition phase.

In this paper, a closed-set Text-Independent Speaker Identification System is reviewed and a new modified algorithm for the matching part is introduced.

II. Feature Extraction

This stage is often referred as speech processing front end. The primary goal of feature extraction is to simplify recognition by summarizing the vast amount of speech data and obtaining the acoustic properties that define speaker individuality. MFCC (Mel Frequency Cepstral Coefficients) is one of the most widely used feature extraction techniques [2]. Since speech signal varies over time, it is more appropriate to analyze the signal in short time intervals where the signal is more stationary. To find the MFCC, the signal is split into short frames and a windowing function is applied for each frame to eliminate the effect of discontinuities at edges of the frames. Then the windowed signal is converted to frequency domain by taking the FFT (Fast Fourier Transform) and Mel scale filter bank is applied to the resulting frames. Average human ear has nonlinear frequency response. Previous research indicates that scaling is linear up to 1 kHz and logarithmic above that frequency. The Mel-Scale (Melody Scale) filter bank characterizing the frequency response of human ear is shown in Fig. 2.1. It is used as a band pass filter during first phase of identification.

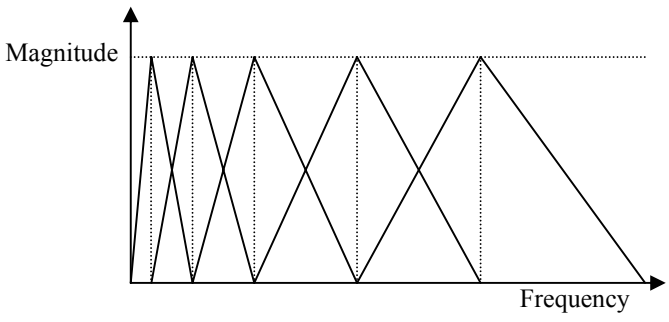

Figure 2.1 Mel Scale Filterbank

After Mel frequency warping the frames, logarithm of the signal is passed to the inverse DFT function converting the signal back to time domain. As a result of the final step, 13 coefficients named MFCC for each frame are obtained. The 0th coefficient is not used because it represents the average energy in the signal frame and contains little or no usable information. The feature extraction steps are depicted in Fig. 2.2.

As the output of feature extraction phase, vectors in 12 dimensions are obtained for each frame. Each vector is called codeword and all codewords of a speaker model is called codebook. At the end of feature extraction, all speakers' codebooks are determined and stored for later use in identification phase.

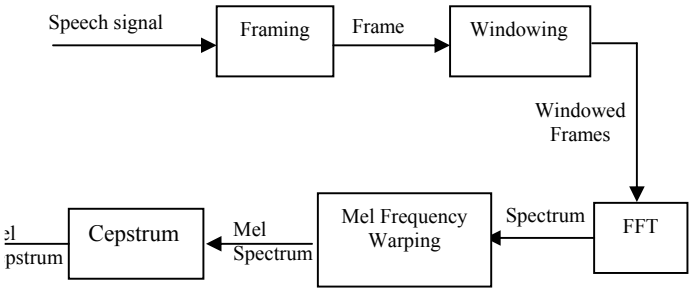

Figure 2.2 Feature Extraction Steps

III. Vector Quantization

For a speech that is sampled at 8 KHz, and a 32 ms frame length with 16 ms overlapping, there will be 499 frames resulting 499 feature vectors for every seconds of speech. The amount of this data is significantly large for efficient processing. Therefore, these vectors need to be compressed using a clustering algorithm. Vector Quantization is one of the preferred methods to map vast amount of vectors from a space to a predefined number of clusters each of which is defined by its central vectors or centroids. Fig.3.1 shows the VQ approach. A speaker is represented by its n feature vectors before the VQ is applied. There will be predefined number of centroids representing the speaker in a more compact way form after clustering operation.

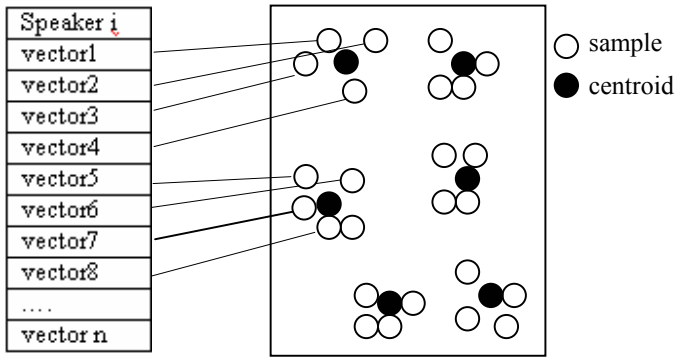

Figure 3.1 Vector Quantization of a speaker

IV. Feature Matching

The last phase of speaker identification process is the feature matching phase, which itself is a general pattern matching (classification) problem. In this phase, similarity scores are calculated for an unknown speech sample and stored models in database. One of the well known matching algorithms is based on computing the Euclidean Distance between the codebooks of the unknown utterance and the reference models [3]. Equation (1) shows the similarity score calculation between the unknown speech sample and a model.

$$S = \sum_{i=1}^{T} \frac{1}{d_{\min}(v_i, C_x)} \tag{1}$$

In the equation, $\boldsymbol{S}$ stands for the similarity measure between two codebooks, C_x (Codebook for the unknown utterance) and C_k (k^{th} codebook in the speaker database). $\boldsymbol{dmin(v_i, C_x)}$ indicates the distance from code vector v_i to the nearest code vector in the codebook C_x. $\boldsymbol{T}$ indicates the number of code vectors. (It is the same for all codebooks.)

An improvement to this method is proposed by [1] uses the weighted matching method, taking correlations between the known models in database into account. This model states that code vectors that are not similar to other codebooks have more power to identify a speaker. Therefore those code vectors that are much discriminative than the others are emphasized by assigning higher weight scores. To implement that, weights for each codebook is calculated and stored in the enrollment step. In the weighted matching method as shown in (2), the local similarity score is multiplied by the weight score associated with the nearest code vector.

$$S_1 = \sum_{i=1}^{T} \frac{1}{d_{\min}(v_i, C_x)} * w_i \tag{2}$$

Fig. 4.1 represents this method more intuitively. For the first code vector v_1 in the Codebook k (C_k) (k^{th} codebook in the models), the distances to the code vectors in the Codebook X (unknown model) are calculated. Then local similarity score is calculated as the reciprocal of minimum distance multiplied by the weight associated to the code vector that yields the minimum distance. The local similarity scores found for each code vector in the Codebook k are summed together to form the final similarity score.

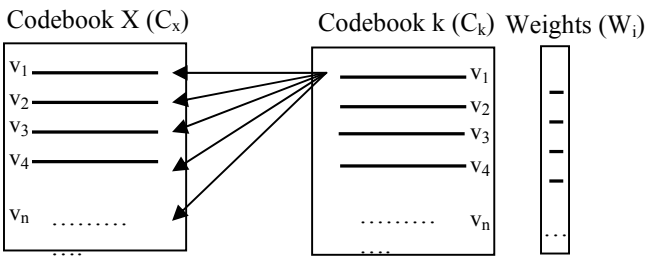

Figure 4.1 Weighted matching method

The new method we proposed to the weighted matching method uses the distances from the Codebook X to the Codebook k as opposed to the first measure. The second similarity measure is computed as in (3).

$$S_2 = \sum_{j=1}^{T} \frac{1}{d_{\min}(v_j, C_k)} * w \qquad (3)$$

S_2 is the similarity measure between the two codebooks, C_x and C_k. $dmin(v_j, C_k)$ denotes the distance from code vector v_j of Codebook C_x to the nearest code vector in the codebook C_k. w indicates the weight value corresponding to the code vector that yields the minimum distance in the codebook C_k. T denotes the number of code vectors.

The Fig.4.2 depicts the new method. For the first code vector v_1 in the Codebook X (C_x) (unknown model), the distances to the code vectors in the Codebook k are calculated. Then the local similarity score is calculated as the reciprocal of minimum distance multiplied by the weight associated to the code vector that yields minimum distance. The local similarity scores found for each code vector in the Codebook X are added together forming the final similarity score.

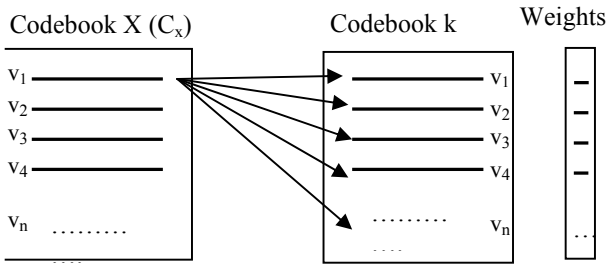

Figure 4.2 The new matching method we proposed

V. Experiments & Results

The new modified method is tested using the speech samples collected from the YOHO speaker database. Four sets of each having 40 speakers are collected from enrollment sessions and another four sets are collected from authentication sessions. The number of recognitions out of 40 speakers for each set is tabulated in Table I. Out of 10 tests, 5 tests for the weighted matching method, and 5 tests for the proposed method are performed. The average recognition rates are 89% for the weighted method and 89.5% for the proposed method.

To be able compare how confident one method recognizes the correct speaker relative to the other, a confidence score is calculated between the similarity score of the recognized model and the second best score. If S_c indicates the confidence score then it is calculated as shown in (4), where S_{b1} indicates the best score and S_{b2} indicates the second best score.

$$S_c = (S_{b1} - S_{b2}) / S_{b1} \qquad (4)$$

TABLE I
EXPERIMENT RESULTS

Speaker Sets	Recognition Rates S1	Recognition Rates S2
Set1 (40)	35 (87.5%)	36 (90 %)
Set2 (40)	34 (85 %)	34 (85 %)
Set3 (40)	38 (95 %)	38 (95 %)
Set4 (40)	38 (95 %)	36 (90 %)
Set5 (40)	33 (82.5 %)	35 (87.5 %)
Average	35.6 (89%)	35.8 (89.5%)

The confidence scores for the two methods are depicted in Fig. 5.1. The results of the experiments suggest that the new modified method is able to find the correct speaker with about 2.35% more confidence in average.

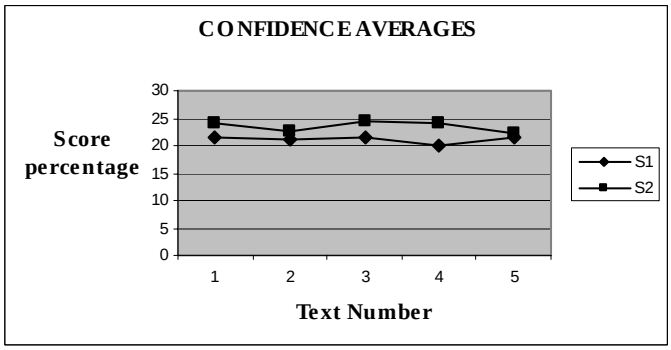

Figure 5.1 Relative confidence scores

Acknowledgment

The author would like to thank to Mr. Volkan Tunalı for his contributions in conducting the experiments.

References

[1] T. Kinnunen and P. Fränti, "Speaker Discriminative Weighting Method for VQ-Based Speaker Identification", Proc. 3rd International Conference on audio and video-based biometric person authentication (AVBPA)), Halmstad, Sweden, 2001.

[2] T. Kinnunen and P. Fränti, " Spectral Features for Automatic Text-Independent Speaker Recognition" Licentiate's Thesis. http://www.cs.joensuu.fi/pages/pums/public_results/2004_PhLic_Kinnunen_Tomi.pdf

[3] Campbell, J. JR. Senior Member, IEEE Speaker recognition: a tutorial. Invited Paper

[4] Evgeny Karpov,.Real-Time Speaker Identification. 15.01.2003. University of Joensuu. Department of Computer Science. Master's Thesis