Pass-Phrase based Speaker Identification

Y.K.Viswanadham
Asst. Prof, I.T.Dept,

T.V.Subrahmanyam
Asoc.Prof, I.T.Dept.

I.Leela Priya
Asoc.Prof, I.T.Dept.

Department of I.T.,Gudlavalleru Engineering College, Gudlavalleru-521356,A.P,INDIA

ABASTRACT
The problem of speaker identification is an area with many different applications. The most practical use can be found in applications dealing with security, surveillance, and automatic transcription in a multi-speaker environment. Speaker identification is a difficult task and the task has several different approaches. The state of the art for speaker identification techniques include Dynamic Time Warped (DTW) template matching, Hidden Markov Modeling (HMM), and codebook schemes based on Vector Quantization (VQ). This paper emphasizes on text dependent speaker identification, which deals with detecting a particular speaker from a known population. The system reads the speech utterance. System identifies the user by comparing the codebook of speech utterance with those of the stored in the database and lists, which contain the most likely speakers, could have given that speech utterance. The vector quantization approach will be proposed, due to ease of implementation and high accuracy.

Keywords: Biometrics, Speaker Identification, LPC, Mel Cepstrum, HMM, VQ, Codebook.

1. INTRODUCTION
In order to implement the pass phrase based speaker ID system, one must go through several steps, including feature extraction, feature matching, and finally, identification of the speaker. Feature extraction is a method that takes a small amount of data from the voice signal which can later be used to generate a representation of each speaker. Feature matching involves the actual procedure of using vector quantization to identify the speaker according to the characteristics of the known speakers. Feature Extraction is the means by which speech data is reduced to much smaller amounts of data which represent the important characteristics of the speech. Many varieties of features can be used for speech processing, such as LPC coefficients, Mel Cepstrum, spectrograph, autoregressive models[1], hidden Markov models (HMM) [2], but the most widely used methods are the Gaussian mixture models (GMM) [3]. Multi layer perceptron, time delay neural networks, back propagation, radial basis function networks [4] and fuzzy neural networks [5] .

The basic building blocks of speaker identification system are shown in the Fig.1. The first step is the acquisition of speech utterances from speakers. To remove the background noises from the original speech. Then the start and end points detection algorithm has been used to detect the start and end points from each speech utterance. After which the unnecessary parts have been removed. Pre-emphasis filtering technique has been used as a noise reduction technique to increase the amplitude of the input signal at frequencies where signal-to-noise ratio (SNR) is low. The speech signal is segmented into overlapping frames. The purpose of the overlapping analysis is that each speech sound of the input sequence would be approximately centered at some frame. After segmentation, windowing technique has been used. Features were extracted from the segmented speech. The extracted features were then fed to the learning and classification.

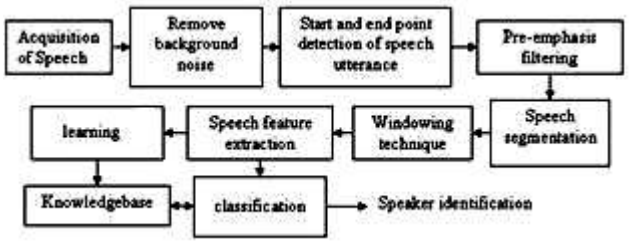

Fig 1: Block Diagram for Speaker Identification

When sound is emitted from the human mouth, it passes through two different systems before it takes its final form. The first system is the pitch generator, and the next system modulates the pitch harmonics created by the first system. Scientists call the first system the laryngeal tract and the second system the supralaryngeal/vocal tract [6]. The supralaryngeal tract consists of structures such as the oral cavity, nasal cavity, velum, epiglottis, tongue, etc. When air flows through the laryngeal tract, the air vibrates at the pitch frequency formed by the laryngeal tract as mentioned above. Then the air flows through the supralaryngeal tract, which begins to reverberate at particular frequencies determined by the diameter and length of the cavities in the supralaryngeal tract. These reverberations are called "resonances" or "formant frequencies". In speech, resonances are called formants. So, those harmonics of the pitch that are closest to the formant frequencies of the vocal tract will become amplified while the others are attenuated.

1.1 LPC Coefficients
The LPC (Linear Predictive Coding) calculates a logarithmic power spectrum of the signal. It is used for formant analysis. The waveform is used to estimate the settings of a filter. The filter is designed in a way to block certain frequencies out of white noise. With the correct settings, the result will match the original waveform.

1.2 Mel-Cepstral Coefficients
Mel-frequency cepstrum coefficients (MCC) are well known features used to describe speech signal. They are based on the known evidence that the information carried by low-frequency components of the speech signal is phonetically more important for humans than carried by high-frequency components [7]. Technique of computing MCC is based on the short-term analysis, and thus from each frame a MCC vector is computed. MCC extraction is similar to the cepstrum calculation except that

one special step is inserted, namely the frequency axis is warped according to the mel-scale.

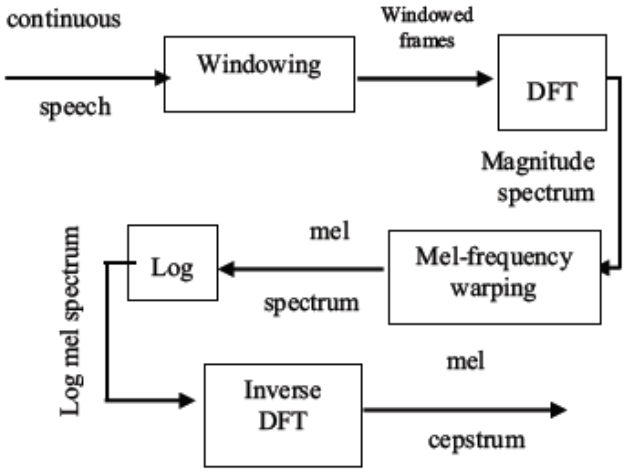

Fig 2: Computing Mel-Frequency Cepstral Coefficients

1.3 Vector Quantization

In our system, we are quantizing about 1200 feature vectors down to 128 codebook vectors. These vectors are individually known as centroid vectors. Ideally, a centroid vector should represent a cluster of feature vectors. The goal is to obtain 128 vectors such that the overall distortion (Euclidean distance) from each feature vector to its nearest centroid in the vector space is minimized. With minimal distortion, an accurate representation of the speaker can be obtained.

We must compare the features based on robustness with the following desirable feature characteristics

1. Cannot be mimicked or consciously controlled by the speaker.
2. Unaffected by health problems of the speaker.
3. Independent of speaking environment.
4. Distinguishable from noise caused by the recording process.

2. METHODOLOGY

2.1 Speaker Classification

In a speaker identification system, each speaker must be uniquely represented in an efficient manner. The means to do this is called vector quantization. Vector quantization is a process of mapping vectors from a large vector space to a finite number of regions in that space. Here 12 mel frequency cepstral coefficients are generated per frame and these are used as the feature vector. Before feeding this large number of features in the learning stage, Vector Quantization technique is applied on these features which compress the large number of data sets into a smaller number of data which acts as a representative of those data sets. The data is thus significantly compressed, yet still accurately represented. Without quantizing the feature vectors, the system would be too large and computationally complex. In a speaker recognition system, the vector space contains a speaker's characteristic vectors, which are obtained from the feature extraction described above.

2.2 Codebooks

After vector quantization takes place, only a few representative vectors remain, collectively known as that speaker's codebook. The codebook then serves as delineation for the speaker, and is used when training a speaker in the system. There are several different approaches to finding an optimal codebook for a speaker. The idea is to begin with a vector quantizer and a codebook and improve upon the initial codebook by iterating until the optimal one is found. The major problem was generating the initial codebook of 128 vectors.

2.2.1 Binary Splitting

The first method we tried is called binary splitting. One begins with one centroid vector, which is the centroid for the entire set of training vectors. From there the centroid is split into two by multiplying the vector by certain factors. The new large codebook is optimized according to the k-means algorithm described below. The process is repeated until the size desired is obtained. We used the mean of each vector dimension to come up with the first centroid. Then we multiplied that centroid by two factors, (1+*e)* and (1-*e)*, to get the two new centroids. *e* is usually in the range of 0.01 to 0.05. Then we used the k-means algorithm to get the best set of centroids for the split codebook. These steps were repeated until a 128-vector codebook was obtained. We found that this method returned a very poor representation of the speaker's feature vectors, even after optimization [8]. Our next attempt was simply to choose 128 random vectors from the feature set and optimize those iteratively. This method is called *random coding*, and was found to be much more effective than binary splitting.

2.3.2 Optimization with K-means

We selected the iterative improvement algorithm known as *k-means* (also known as the LBG or the generalized Lloyd algorithm). Given a set of *I* training feature vectors, {**a**1, **a**2... **a***I*} characterizing the variability of a speaker, we want to find a partitioning of the feature vector space, {S1, S2,..., S*M*}, for that particular speaker where S, the whole feature space, is represented as S = S_1 U S2 U...U S*M*. Each partition, S*i*, forms a non overlapping region and every vector inside S*i* is represented by the corresponding centroid vector, **b***i*, of S*i* [8]. Each iteration of k-means moves the centroid vectors such that the accumulated distortion between the feature vectors is lessened.

The more iteration you run, the less distortion you should have. The algorithm takes each feature vector and compares it to every codebook vector which is closest to each.

$$D_j = \sum_{i=1}^{I} (t_i - v_{j,i})^2$$

Where *v* are the vectors in the codebook and *t* is the training vector. The minimum distortion value is found among all measurements. Then the new centroid of each region is calculated. If *x* is in the training set, and *x* is closer to *vi* than to any other codebook vector, assign *x* to *Ci*. The new centroid is calculated as where *Ci* is the set of vectors in the training set that are closer to *vi* than to any other codebook vector. The next iteration will recomputed the regions according to the new

centroids. The total distortion will now be smaller. Iteration continues until a relatively small percent change in distortion is achieved.

3. IMPLEMENTATION

The voice data is sampled at 16000 Hz, and is split up into frames of 240 samples, which corresponds to 15ms. The frames overlap by 80 samples, meaning there is a frame every 10ms. Each frame is run through a simple filter with transfer function for the purpose of pre-emphasizing the high frequencies of the speech. The frame is then multiplied by a Hamming window, and 12th order LPC autocorrelation analysis is run on it.

3.1 Preprocessing

First, we wanted to remove all non-speech samples from the recorded, temporal signals. We implemented this using an energy detection algorithm developed in a heuristic manner from our data. Since none of our recordings contained speech in the first 100 ms of recording time, we analyzed this time frame and generated an estimate of the noise floor for the speaking environment. Then we analyzed each 20 ms frame and removed those frames with energy less than the noise floor.

3.2 Training

Each speaker records several training sentences, which are concatenated and from which features are extracted. The accumulated feature vectors are used to generate a codebook according to the algorithm described above. This codebook is used for identification.

3.3 Matching

The speaker identification system works by taking the feature vectors of an arbitrary input from one of the trained speakers and comparing them with all the codebooks in an *exhaustive search*. First, feature extraction is applied to the unknown speaker's input sample. Then, for each known speaker, each vector from the test utterance is quantized to that speaker's codebook, and the distortion involved in doing so is saved. The entire test utterance is evaluated this way, and the sum of the distortions of each frame represents the quality of the match. This process is repeated for each speaker, and the one with the least total distortion is chosen as the speaker [8]. Two types of errors can occur in a speaker verification system, namely, false rejection and false acceptance. A false rejection (or non detection) error happens when a valid identity claim is rejected. A false acceptance (or false alarm) error consists in accepting an identity claim from an impostor [9]. Both types of error depend on the threshold θ used in the decision making process. With a low threshold, the system tends to accept every identity claim thus making few false rejections and lots of false acceptances.

4. EXPERIMENTAL RESULTS

To test our system, we used MatLab. From our dataset, we divided it into two partitions. One partition is used to train the system and the other partition is used to test the system. The test samples included non registered speakers. The results of the system can be seen in Table 2. In addition, we came to the conclusion that feature extraction of the dataset was successful due to the smaller number of data points on the graphs generated by MatLab.

Table 2 : From 40 Test Samples

Accept	Reject	False Accept	True Reject
97%	3%	3%	1%

Accuracy is (97%-3%)=94%

5. ANALYSIS

The overall implementation of the speaker identification system was very challenging. The database consists of speech samples from 22 adults, 12 male and 10 female. Speakers were asked to read a pass phrase in normal speed, under normal laboratory conditions. Speech signals are sampled at 16 kHz with 16 bits A-D conversion. For each speaker, two files are recorded per phrase; one for training and one for testing the details are given in Table 1.

Table 1: Details of Samples Database

No.of Speakers (Male+Female)	22 (12 + 10)
No.of Phrases	3
No.of samples per phrase for each speaker	2
Total samples	22 x 3 x 2=132
Training Samples (20 Speakers)	90
Test Samples (22 speakers)	42

Training sample texts are read from the same phonetically rich text, which is about 1 minute long. On the other hand, testing samples are recorded from 10 seconds long random text. After recording, all samples are normalized. Our system is realized in the MATLAB 7.0 environment. The PC used in computations has single Pentium IV processor at 3.0 GHz and 2GB RAM. The Mat Lab implementation of feature extraction was very simple to implement. After the MatLab implementation was done, we attempted to model the speaker system.

6. CONCLUSION

The vector quantization approach to speaker identification is an efficient and accurate approach to the problem. The accuracy rate could be improved by some of more complex voice features mentioned above. Overall, the system meets the original goals.
As we obtained fairly reasonable results for a text dependent speaker recognition system. We plan to use this speaker identification scheme in a real-time virtual dialogue environment where the virtual characters communicate with real persons based on their recognized identifications and personal profile stored in the database.

However, the problem of automatic speaker recognition is very broad field with many problems yet to be solved. We decided to handle the challenging problem of text dependent speaker identification.

7. FUTER SCOPE

The *n*-speaker detection is similar to speaker verification [10]. It consists in determining whether a target speaker speaks in a conversation involving two speakers or more. The difference from speaker verification is that the test recording contains the whole conversation with utterances from various speakers.

7. REFERENCES

[1] F. Bimbot, L. Mathan, A. de Lima and G. Chollet, "Standard and target driven AR-vector models for speech analysis and speaker recognition," in IEEE ICASSP, vol. 2, 1992, pp. 5–8.

[2] J. de Veth and H. Bourlard, "Comparison of hidden Markov model techniques for automatic speaker verification in real-world conditions," Speech Communication, vol. 17, Mar. 1995, pp. 81–90.

[3] D. Reynolds and R. Rose, "Robust text-independent speaker identification using Gaussian mixture speaker models," IEEE Trans. Speech Audio Processing, vol. 3, Jan. 1995, pp. 72-83.

[4] M.W. Mak, W.G. Allen, and G.G. Sexton, "Speaker identification using multilayer perceptron and radial basis function networks," Neurocomputing, vol. 6, no. 1, 1994, pp. 99-117.

[5] Z.X. Yuan, B.L. Xu, and C.Z. Yu, "A kind of fuzzy Neural networks for text-independent speaker identification," in Proc. IEEE Int. Confe. Acoustics, Speech, Signal Processing, 1996, pp.657-660.

[6] Brian J,Jennifer Vining " Automatic Speaker Recognition Using Neural Networks" ,2004.

[7] M.W. Mak, W.G. Allen, and G.G. Sexton, "Speaker identification using multilayer perceptron and radial basis function networks," Neurocomputing, vol. 6, no. 1, 1994, pp. 99-117.

[8]A. Gersho, R. Gray, "Vector Quantization and Signal Compression", Kluwer Academic Publishers, Boston, 1992.

[9] Frederic Bimbot "A Tutorial on Text-Independent Speaker Verification", EURASIP Journal on Applied Signal Processing 2004:4, 430–451

[10] M. Przybocki and A. Martin, "The 1999 NIST speaker recognition evaluation, using summed two-channel telephone data for speaker detection and speaker tracking," in *Proc. European Conference on Speech Communication and Technology (Eurospeech '99)*, vol. 5, pp. 2215–2218, Budpest, Hungary, September 1999