Text-Independent Speech Recognition

Thomas Soong
Til Phan

Introduction

Speaker identification is an area with many different applications. The most practical uses can be found in areas such as security, surveillance, and automatic transcription in a multi-speaker environment. The goal of this project is to understand the development and implementation of a text-independent speaker recognition system. We will analyze the system by interpreting feature matching and extraction characteristics. After the text-independent speaker recognition system is understood, we will attempt to model it using Ptolemy. We hope to be able to model a system that will simulate a text-independent speaker system. As part of the literature review, we looked at various journals, IEEE proceedings, and other resources describing the applications of speaker recognition systems.

Procedure Overview

Speaker recognition is a complex problem that has several different approaches. Our group has focused on a specific approach involving *vector quantization*. First, a training set must be acquired. In general, the training set will consist of between 50-100 speakers. To begin the process of speaker identification, a random utterance is input into the PC via the sound card. The database speakers are sampled at 16kHz, and will be downsampled to 8kHz. The utterances will be acquired into the system by using a 32ms Hamming window,

which will input into frames of 10ms. The signal is the put through a pre-emphasis filter with a transfer function of $H(z) = 1 - 0.95z^{-1}$ which helps to minimize the prediction error [1]. The signal is now ready for *feature extraction*, which is performed by Linear Predictive Coding (LPC). The output is a set of coefficients that represent a combination of speaker information, such as vocal tract shape, glottal wave, and lip transfer function, and is a suitable to characterize a particular speaker extremely well. Next, *Cepstral Analysis* is performed, where cepstrum coefficients are determined from the LPC coefficients. LPC and Cepstral Analysis are applied to both the training set and the unknown utterance. For the training set, these coefficients are used to generate a codebook for each speaker. The unknown utterance is compared to the existing codebooks and an average distance is determined. The pair with the smallest distance is determined to be the correct speaker.

Feature Extraction

LPC Analysis

Feature extraction is the process by which a small amount of data is taken from the voice signal. This data should vary significantly from speaker to speaker and not be affected by noise or communication channel variation. The standard method for deriving this data is *linear predictive coding*, or LPC, which models the vocal tract with a finite-order all-pole transfer function. The coefficients of this model accurately indicate the instantaneous configuration of the vocal tract. These can then be used to identify the type of sound being produced, for speech recognition, or in this case, to identify the speaker based on small variations in the parameters.

A Hamming window is applied to the input voice signal to break it up into short (10-20ms) frames of data, over which the vocal tract remains more or less constant. The actual computation of the *predictor coefficients* is based on minimizing the estimation error *e(n)* given in [4] by $e(n) = s(n) - \sum_{i=1}^{p} a_i s(n-i)$, where s(n) is the signal and a_i are the predictor coefficients. One approach uses the autocorrelation $\mathbf{r}_s$ of the input and a Toeplitz matrix form of the autocorrelation coefficients, $\mathbf{R}_s$. Then the vector of LPC coefficients is given by

$$\mathbf{a} = \mathbf{R}_S^{-1}\mathbf{r}_s$$

There exists a computationally efficient algorithm to solve for **a** from the autocorrelation coefficients, known as the Levinson-Durbin recursion [4].

Cepstral Analysis

LPC coefficients can be converted to many different forms, such as cepstral coefficients, line spectral pairs, log area ratios, and others [1]. *Cepstral Analysis* has been shown to have the best rates for speaker identification [1], and will be used in this project.
The cepstrum is defined as the inverse Fourier Transform of the log of the Fourier transform of the input signal. In this case, the input is the impulse response of the linear predictive filter obtained by LPC analysis. The predictor coefficients are easily transformed to cepstral coefficients by the recursive relation from [4]:

$$c_{lp}(n) = a_n + \sum_{i=1}^{n-1} \left(\frac{i}{n}\right) c_{lp}(i) a_{n-i}$$

These cepstral coefficients are desirable because they are not very sensitive to variations in the communication channel or to noise. They are sometimes weighted to emphasize certain coefficients, based on the sensitivity to speaker variation. They can be transformed further to remove effects of noise and communication channel variations, by such algorithms as *mean subtraction* and *affine transformation*.

The final result is a *k* by *N* matrix, where *k* is the order of the LPC analysis, and *N* is the number of frames in the utterance.

Implementation

Source code to perform the LPC computation is freely available from the Texas Instruments WWW site. Cepstral transformation will be analyze and the resultant 12 or 14-point *feature vector* will be determined.

Feature Matching

Vector Quantization

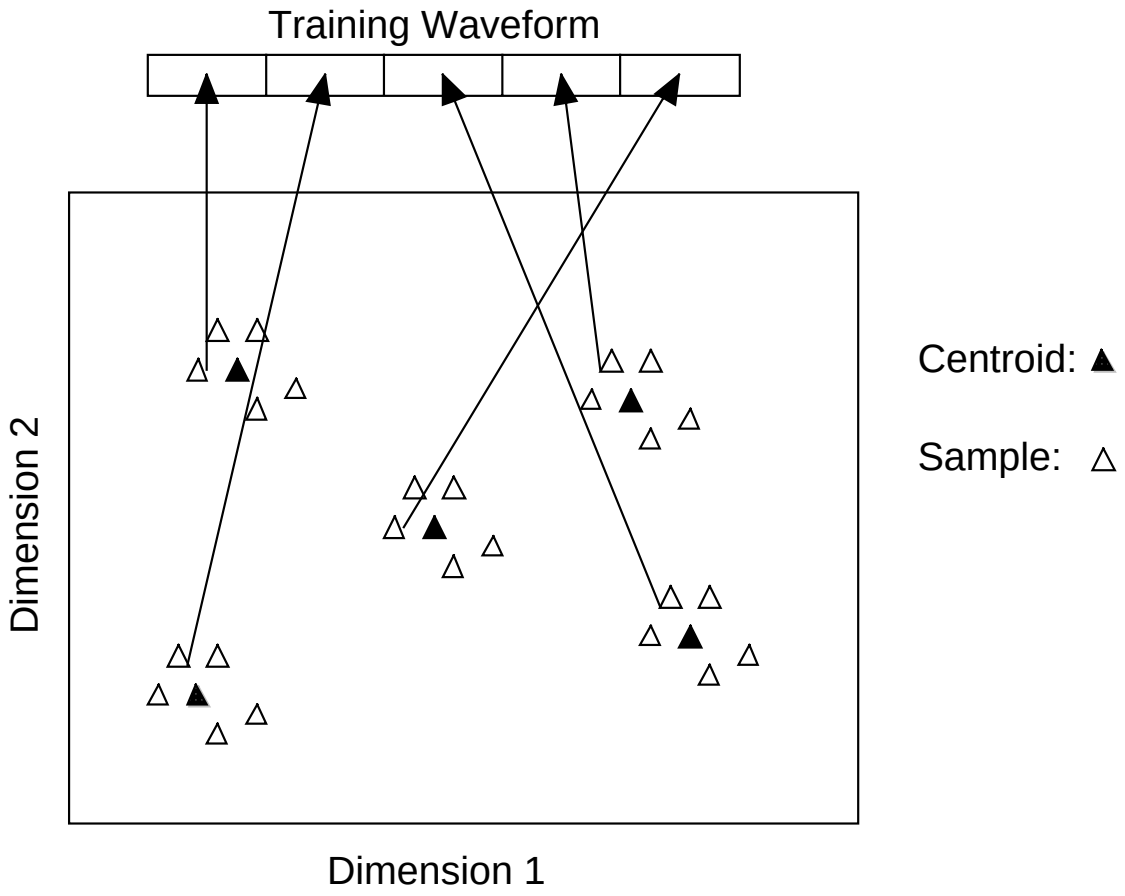
Recent studies have shown that we can implement speaker-independent speech recognition through a method called *vector quantization* (VQ). In general, VQ is the process of mapping a vector from a large vector space to a finite number of regions of that space[2]. In this way, the data is compressed at the cost of distortion. For this particular application, the regions correspond to specific spoken sounds, and the distortion caused by the quantization is a measure of how well the utterance matches the speaker.

Recognizing a speaker is a process of comparing the input vectors with training vectors to

find the best match. The reliability of the recognition depends on the quality of the dictionary.

A *codebook* is generated for each speaker. A codebook is composed of different vectors, which represent the essential characteristics of each speaker. Each codebook is generated as follows: Given a set of training feature vectors (a 12-point vector for each frame of the utterance) which characterize the speaker, find a partitioning of the feature vector space. Each region contains a cluster of vectors, which represent the same basic sound. This region is represented by the *centroid vector,* which is the vector, which causes the minimum distortion when vectors in the region are mapped to it [7]. Thus, each speaker has a codebook with a number of centroids. Input vectors are matched to the best region for each speaker, determined by the distance to the centroid of the region. The speaker with the lowest distortion is the speaker of the frame. Alternately, distances can be accumulated throughout the entire test utterance, with the speaker being determined after.

This is a diagram illustrating codebook generation [7]:



The distortion is defined as the distance to the centroid, which can be calculated either as the Euclidean distance in the vector space, or as a more robust form, such as the *Weighted Likelihood Ratio,* or WLR, given in [5] by

$$WLR = \sum_{i=1}^{12} (r_i - r_i^*)(C_i - C_i^*)$$

where r_i is the LPC coefficient i and C_i is the cepstral coefficient i. The $*$'s indicate reference vectors.

Implementation

We will analyze the implementation of codebook generation. The algorithm to do this is not yet determined, although we have some references. Actually matching the inputs to the codebooks is a straightforward, though computationally intensive, process. Typically there exist a library of 50-100 speakers, with 64 or 128 entries in each codebook.

Project Direction

We intend to look closely at feature matching and extraction of text-independent speaker systems. First we will research and understand both feature extraction and matching. Then we will attempt to model a simple speaker system with Ptolemy.

References

1. B. Atal "Automatic Recognition of Speakers from Their Voices". *Proceedings of the IEEE*, vol. 64, April 1976, pp. 460-475.

2. A. Gersho, R. Gray, *Vector Quantization and Signal Compression*, Kluwer Academic Publishers, Boston, 1992.

3. Linguistics Data Consortium: http://www.ldc.upenn.edu

4. R. Mammone, X. Zhang, R. Ramachandran, "Robust Speaker Recognition", *IEEE Signal Processing Magazine*, September 1996.

5. F. Soong, E. Rosenberg, B. Juang, and L. Rabiner. "A Vector Quantization Approach to Speaker Recognition". AT&T Technical Journal, vol. 66, March/April 1987, pp. 14-26.