Speaker Differentiation with Speech Segmentation Under Noiseless Setting

Abstract

Speaker identification has seen extensive usage in both academic and commercial applications. Common approaches employ various machine learning algorithms on extracted features, such as pitch, formant frequencies, and Mel-Frequency Cepstrum. This paper's approaches the problem by performing vowel segmentation and vowel-to-vowel comparison. The paper proposes a stable method to extract vowels based on k-means clustering. Afterward, extracted features from each vowel are fed into a classifier. Various combinations of extracted features and learning algorithms are evaluated for accuracy. The best performance was achieved by extracting three isolated vowels and using support vector machine with an error rate of 3.86 %

Index Terms: speech recognition, speaker recognition

1. Introduction

Speaker Recognition is the process of extracting a speaker's identify through the speaker's voice characteristics. Such systems are commonly used in voice biometrics and generating voice footprints for security.

The main objective of this project is to build a speaker identification system that mimics the way humans process the differentiating unique speakers by combining speech and speaker recognition techniques. Typically, humans use a combination of pitch, timbre, and pronunciation of individual words to distinguish different speakers. Following this idea, we propose a speaker recognition system that first isolates the words being said with speech recognition techniques, and then compare how the corresponding words in other recordings differ.

In this paper, speech is separated into two basic categories: vowels and constants. Vowels are usually modeled as periodic signals with harmonics generated by vibrating vocal folds modified by the vocal tract's transfer function. The fundamental frequency of the harmonic is also called the pitch frequency. Usually, vowels can be characterized by its formant frequencies– the specific resonant frequencies of the vocal tract. In contrast, consonant are usually aperiodic with no clear harmonics and no clear resonance. Sometimes a consonant has only smeared energy at the high frequency spectrum correspond to noise generated by flowing air passing out of the mouth. This is often the case in a class of consonants known as the fricatives, which includes sounds such as [s]. Sometimes the vocal tract transfer function of a consonant contains zeros that absorb most of the speech signal's energy. This usually happens with nasal consonants such as [n] and [m]. See Figure 1 and 2 for sample spectrograms of these consonants. As can be seen from the spectrograms, the consonants tend to be either noisy or have very little energy presented on the spectrogram. Thus they tend to have fewer distinct features for speech and speaker recognition. [1] As a result, our proposed system will ignore the consonants in favor of the vowels for speaker recognition.

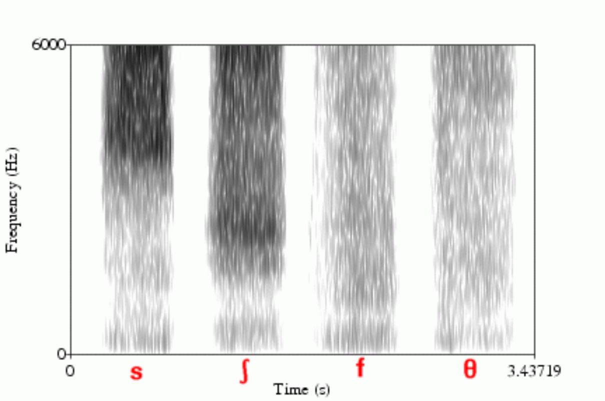

Figure 1: *A spectrogram of voiceless fricatives.*

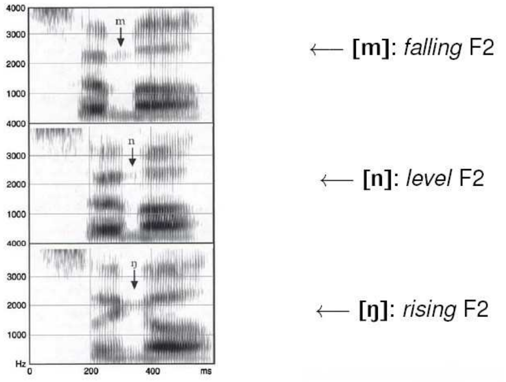

Figure 2: *A spectrogram of nasals.*

In this paper, we will discuss a two-step speaker recognition system. In the first step, the system performs rudimentary speech recognition by extracting the vowels from a speech signal using a modified version of k-means clustering. In the second step, the system extracts a set of features from each vowel obtain in step 1 and feeds them into a classifier for speaker recognition. These features include pitch frequency, formant frequencies, Mel-Frequency Cepstrum Coefficients (MFCC), and higher-order MFCCs. Only the corresponding vowels in each recording will be compared. The system model is shown in Figure 3.

The paper is organized as follows: Section 2 discusses the assumptions on the speech signals and the experimental data. Section 3 goes over the algorithm used for vowel isolation. Section 4 discusses feature extraction. Section 5 talks about the various machine learning algorithms used to classify each recording pair. Section 6 reports on the accuracy of the classifier and the effects of supplemental techniques on error rate. Section 7

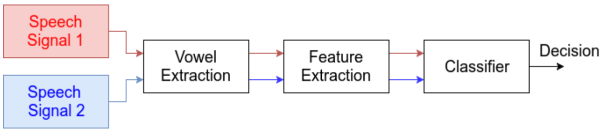

Figure 3: *A model of the speaker recognition system.*

provides a summary of this report as well as some future directions.

2. Problem Formulation

2.1. Assumptions

This project considers the problem of same-text speaker recognition under noiseless setting. More precisely, given a pair of speech signals, we propose an algorithm to determine whether the two utterances are from the same speaker under the following assumptions:

1. All the speech signals contain the same phrase.
2. The speech signals are noiseless or have high signal-noise ratio.
3. The speech signals have at least one vowel.
4. The speakers are flat (there is no yelling or singing during the production of speech signals) so the signal could be consider approximately stationary over short time segment.

2.2. Data Set

The experimental set used is a set of recordings from 10 male speakers and 10 female speakers provided by UCLA Medical School. All recordings contain the same utterance, “A pot of tea helps to pass the evening”. Each utterance is repeated 5 times for a total of 100 recordings. The speech signals are all relatively flat and clean. Thus the assumptions made in the previous sections are satisfied.

The test the system, 450 pairs of recordings are chosen from the data set along with an intra-speaker indicator indicating whether the speakers are the same. The data is spit into 5 parts for five-fold cross validation. That is, five trials are performed, and for each trial, 80% of the recording is used for training and the remaining 20% is used for testing. The performance of this approach is evaluated by averaging the testing error from the 5 trials are averaged.

3. Vowel Extraction Algorithm

3.1. Mel-Frequency Cepstral Coefficients

The Mel-Frequency Cepstral Coefficients is a type of cepstrum derived based on Mel-Frequency, which is a type of frequency representation that closely approximates human auditory system. More formally, while there are many variations, a common variant based on discrete cosine transform is defined in the following steps [2]:

1. From a windowed speech signal $x(t)$, obtain the Discrete Fourier Transform $X(k)$.
2. Convert to Fourier Transform to Mel-Scale to get $\tilde{X}(k)$.
3. Perform Discrete Cosine Transform on $\log\left(\tilde{X}(k)\right)$ to obtain the MFCC.

MFCCs are known to be very effective in speaker and speech recognition in the absence of noise. While the exact reason for its effective is not well understood, there are some conjectures. First, the derivation of MFCC is based on a model of human auditory system. Thus it seems the performance of MFCC is not unwarranted. Second, it can be shown that the amplitudes of MFCC decreases exponentially. Thus, by taking the first couple coefficients, we are in effect performing a form of lossy data compression that preserves most of the structures in the original speech signal. And this data compression process helps reducing the dimension of the data, thus simplifying the problem complexity [1].

Furthermore, in the context of speech recognition, many incorporates the dynamics of the speech signal into their model through the use of Delta and Delta-Delta, which are the first order and second order finite difference of the MFCCs. While it is usually not done in speaker recognition, this project also considers the use of them for speaker recognition.

In spite of the benefits MFCCs provide, it is known that they are not robust against noise. For example, a signal-noise ratio of 1 can completely overwhelms a system using MFCCs. While there are attempts made to address this issue, in general the sensitivity of MFCCs to noise is an issue plaguing speech/speaker recognition systems [3] . However, as stated in Section 2 the speech signals are assumed to be noiseless. As such, we will not delve on this issue.

For this project, MFCCs will be used for both speech recognition and speaker recognition.

3.2. K-means Clustering

K-means clustering is an unsupervised learning algorithm used to classify groups of data. The main goal of k-means clustering is to categorize a set of datapoints into k clusters. The algorithm first assigns every point to a mean. The means are then recalculated from the cluster that was assigned to them and the dataset is reassigned again. The process is repeated until the means converge onto the global or local maxima. This algorithm is used in part to identify vowels in the recording.

In this study, the k-means clustering algorithm is trained to group MFCC with 26 coefficients into 3 clusters. One cluster was reversed for silence. Through experimenting, it was determined that classifying the vowels into 2 clusters yield the best results. The algorithm is ran as to minimize the aggregate

$$\arg\min \sum_{k=1}^{3} \sum_{i \subseteq S_k} ||x_i - \mu_k||^2$$

where S_k refers to the k’th cluster, x_i refers to the i’th MFCC from sample x, μ_k refers to the cluster centers. In this paper, the three cluster centers is provided beforehand for the greatest likelihood of algorithm to converge onto the global maximum. The cluster centers were computed from the aggregate average of cluster centers of 100 recordings in the problem set. The variance of cluster centers from recording to recording was found to be low, so the aggregate average can be assumed be to true cluster center of each recording in all cases. Thus, for efficiency, the assignment operation of the k-means clustering is run only.

3.3. Vowel Classification

The Vowel extraction algorithm is performed in three stages. In the first stage, the recording is low-pass filtered with a moving average with a cutoff frequency of about 4400 Hz. A moving hamming window is applied to the recording to generated 36 ms-long frames with a 12 ms shift. The MFCC of each windowed frame is calculated to form the MFCC-time representation of the recording.

The second stage then performs k-means clustering on the MFCC-time representation. The MFCC-time representation is first normalized to prevent biasing the vowel classifier. Each frame in the recording is assigned to one of the two vowel clusters or to the silence cluster. Figure 4 showcases the distance to each of the three cluster centers for a given frame for a particular recording. The red and blue lines signify the frame by frame MFCC distance to the vowel cluster centers. The yellow line signify the frame by frame MFCC distance to silence cluster center.

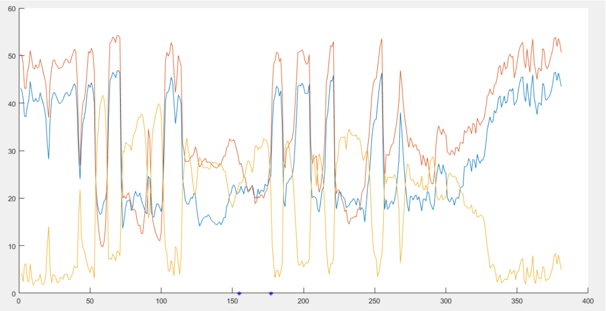

Figure 4: *Frame by Frame MFCC distance to the three centroids*

The third stage classifies each frame as belonging to one of three cluster centroids. At each frame, it can hold three values: it is either considered silence, or one of the two vowel centroids. A series of peak detection, thresholding, and filtering follows on the clustering result to determine vowel location in the recording. A 9-frame mode filter is first used on the classified data to remove any aberrant peaks cause by noise. The mode filtering works by setting the value of each frame to be equal to the most frequent value of this neighbors. For this approach, the mode filter will only change the frame's value if more than 80 % of the frame's neighbors have the same value. Figure 2 demonstrate the result of vowel classification and subsequent filtering. A value of 0 corresponds to silence and a value of 1 or 2 corresponds to one of the vowel centroids.

The third stage performs silence removal and peak detection. The recording is truncated such that the silence leading up to the initial vowel and the silence trailing after the last vowel are removed. The truncated vowel classifier is run through peak detection to isolate specific vowels. Peaks with prominence and width below certain threshold are filtered out. The vowels first targeted for isolation are 'tea', 'helps', and 'pass'. Peaks are matched with the signature of the targeted vowels and corresponding time stamps are fed into feature extraction.

4. Subsequent Feature Extraction

4.1. System Process

The feature extraction process begins with the vowels extracted by the vowel extraction algorithm mentioned in Section 3. Each vowel is first partitioned into N intervals, from each interval a hand-selected set of features are calculated. Next, for each vowel, we use the average of the features over all the intervals as its features. This process is summarized in Figure 6.

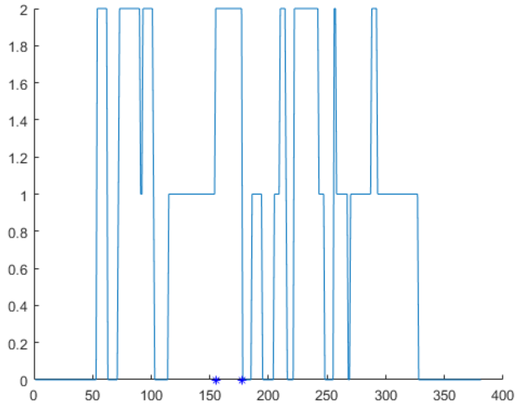

Figure 5: *Frame by Frame MFCC distance to the three centroids*

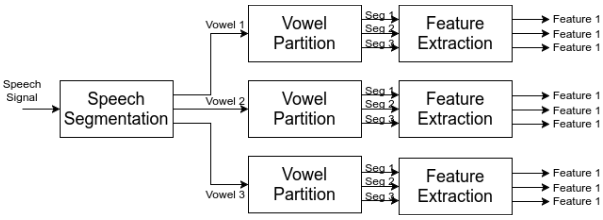

Figure 6: *System flow of the feature extraction process.*

4.2. Features

For each of the vowels, we extract the following features:

- Pitch: As human ears presumably use pitch as one of its features, pitch is a natural feature used by many speaker recognition systems. While the pitch is mostly time-invariant in English, there are still some minor changes in pitch due to stress and intonation. Thus, comparing the pitch on a vowel by vowel basis in a sentence should improve the accuracy of our prediction system.
- Formants and Formant Amplitudes: Conventional wisdom states that while the formants are powerful tools in speech recognition, they are less ideal in speaker recognition. This statement is based on the assumption that the formants depend on the word be spoken, so it's hard to compare the formants directly. But since our algorithm first extracts the vowel segments from the speech signal, we can assume the vocal tract is approximately time invariant for each the vowel segment and then compare the formants for the segments.
- Mel-Frequency Cepstral Coeffcients (MFCC): The MFCCs are obtained by first mapping the frequency content of a speech signal onto the Mel scale (which is empirically modeled on human hearing), and then transform back to get the cepstral coefficients. The MFCCs have repeatedly been shown to achieve very good results on both speaker and speech recognition under noiseless setting, and thus they are one of the key features chosen for our algorithm. Note, however, that the performance achieved through MFCC degrades considerably in the presence of noise. As is often the case, we use the first thirteen coefficients.

- Higher Order MFCCs: The higher order MFCCs are the deltas (the first derivatives of the MFCCs) and the delta-deltas (the second derivatives of the MFCCs). The higher order MFCCs are usually used to model the dynamical nature of speech signal. While they are not usually used in speaker recognition, experimentally, we find that the use of these features improve our performance by about 2%.

After features are extracted from the speech signals, we perform an extra preprocessing step by centering and normalizing it. Mathematically, if there are N samples with D features, and $\tilde{h}_{i,j}$ represents the sample i's feature j, we create $h_{i,j}$ defined as

$$h_{i,j} = \frac{\tilde{h}_{i,j} - \text{mean}(\tilde{h}_{:,j})}{\left(\text{var}(\tilde{h}_{:,j})\right)^{1/2}}$$

where

$$\text{mean}(\tilde{h}_{:,j}) = \frac{1}{N} \sum_{i=1}^{N} \tilde{h}_{i,j}$$

is the sample mean of feature j and

$$\text{var}(\tilde{h}_{:,j}) = \frac{1}{N-1} \sum_{i=1}^{N} \left(\tilde{h}_{i,j} - \text{mean}(\tilde{h}_{:,j})\right)^2$$

is the sample variance of feature j. This is done to ensure that all the feature inputs has zero-mean and unit variance, so that the statistical model would treat the features equally.

5. Classification Algorithm

5.1. System Process

The classification algorithm uses supervised learning to train the model. More specifically, we are given a label vector $y \in \mathbb{R}^K$, where each entry y_k denotes whether two samples i_k and j_k belong to the same speaker. A value of 0 denotes different speakers, and a value of 1 denotes same speakers. We feed the absolute difference of features along with the label to the statistical model. i.e., our k'th sample would consist of the feature vector $x_k \in \mathbb{R}^N$ and label y_k, where x_k is defined by

$$x_{k,l} = |h_{i_k,l} - h_{j_k,l}|$$

where $l = 1, 2, \ldots, N$. A illustration of this is shown in Figure 7.

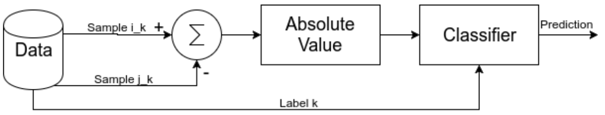

Figure 7: *The system model of the classifier.*

We also perform five-fold cross-validation to approximate the generalization error.

One thing to note from this speaker recognition approach via speech segmentation is that the number of features grow linearly as the number of extracted vowels. As a result, in a setting where training samples available to the statistical model is rare, the dimension of the feature space can easily exceed the number of training samples, and thus we run into the issue of overfitting. To combat this issue, we examine two possible solutions: the ensemble method and the dimensionality reduction method.

5.2. Ensemble Method

In the first method, we use an ensemble method to minimize the generalization error. This is done by first training many weak classifiers that take only a couple available features as the input, and then assign a score to each classifier based on its performance. Then, for the actual classification, our classifier make the final decision by making an weighted average of the individual weak classifiers based on their scores. An illustration is shown in Figure 8.

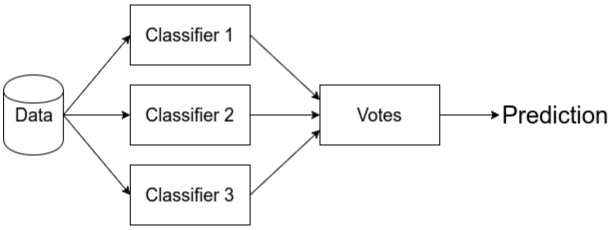

Figure 8: *An illustration of the ensemble method.*

The effect of this method is two fold: first, since each individual classifier only uses a couple features, they are unlikely to over fit the data; second, while the individual classifiers are weak, it is unlikely that all the classifiers would all make the same mistake, so by taking the average of their prediction, the final output would be more likely to be right. Furthermore, if the output of the individual classifier has a good interpretation, the whole ensemble can also be interpreted similarly by treating it as taking average from a sample of classifiers.

There are however some performance issues with the ensemble method. For example, even though the ensemble is easy and fast to train (training weak classifiers is usually a straightforward process with low time complexity), actually making a prediction might take a longer time because the ensemble's prediction requires the outputs from all the individual classifiers. Thus, support for parallel computing or classifier pruning might be needed to provide real time speaker classification. This consideration, however, is not within the scope of this project.

In our experiment, we tried ADAptive BOOSTed (ADABOOST) Random Forest and Bagged Random Forest with 500 trees.

5.3. Dimensionality Reduction Method

In the second method, we first reduce the feature vector's dimension using techniques such as Principal Component Analysis (PCA) to extract the most significant variations in the features, and then perform classification with a more power classifier. This approach is similar to a technique called eigenfaces in facial recognition [4]. An illustration of this method is shown in Figure 9.

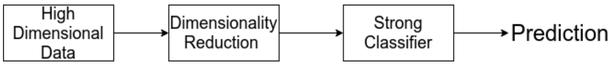

Figure 9: *An illustration of the dimensionality reduction method.*

The main issue with this approach is that Principle Component Analysis replies unsupervised learning. i.e. it makes no use of the labels available. As a result, the variations detects by PCA might not be useful to our classification problem.

For example, PCA might detect the largest variation among the speech signals to be difference in male and female speech. But if we are only comparing male speech to male speech and female speech to female speech, this variation would be useless. Thus, under this problem setting, Fisher Discriminant Analysis (FDA), which takes the data labels into account, might be more appropriate. Unfortunately, due to time constraint, this project does not implement FDA and merely uses LDA.

As for the actual classifier, we employed Soft-Margin Support Vector Machine (SVM) with Gaussian Radial Basis Kernel. SVM is chosen to deal with potential non-linearity of the separating region at the possible cost of generalization error, which would be dealt with by the dimensionality reduction step and cross validation. Note that other choice of kernel could potentially lead to better performance than the generic kernel we have chosen, but due to time constraint, this project implements only Gaussian Kernel.

6. Experimental Results

The vowel segmentation algorithm's peak detection stage has its parameters, such as thresholding and search range, adjusted so that all 100 recordings successfully. Initially, only two extractable vowels were used. The initial configuration of the approach is extracting pitch, the first three formants, the amplitudes of the three formants, and the MFCC on two extracted vowels for an initial set of 38 features per recording. In later trials, additional vowels, higher derivative MFCCs, and modification to the speaker identification classifier increase the number of features and push the error rate lower.

For the classifier, we experimented with Adaboosted Random Forest and Bagged Random Forest, each with 500 tries. A Support Vector Machine with Gaussian Kernel was also experimented with.

Furthermore, it is found that the use of PCA did not help to reduce the generalization error. It merely degrades the performance. As discussed in Section 5, this is like due to the fact that PCA is an unsupervised learning algorithm did not take the data label into account. Thus, the largest variation in the features might not correspond to the variations in data labels. Therefore, PCA is removed from the speaker recognition system and a simple SVM is used directly. Surprisingly, this yields fairly decent result as shown in Table 1. The error rate, however, should be treated cautiously as its rapid as the number of feature size increases hints at overfitting the data.

As discussed in Section 2, 450 pairs of recording provided by UCLA Medical School are used to evaluate the system. Table 1 shows the average error rate for each feature and for each classifier from five-fold cross validation.

Features	Adaboost Tree	Bagged Tree	SVM
2 Vowels, F0-F3, A1-3, MFCC	16.68%	11.56 %	16.44 %
2 Vowels, F0-F3, A1-3, MFCC, delta, delta^2	14.34%	8.33%	9.56 %
3 Vowels, F0-F3, A1-3, MFCC, delta, delta^2	9.33 %	7.11%	3.78 %

Table 1: *The error rate with various features and various classifiers.*

7. Conclusions

This paper proposes a two-stage same-text speaker recognition system under noiseless setting. The first stage of the system performs vowel extraction with a rudimentary speech recognition algorithm. The second stage of the system performs vowel-to-vowel feature comparison to estimate whether the speakers in the speech signals have the same identity .

Experimenting on the data set provided by UCLA Medical School, we find that an error rate of 7.11 % is achievable using three vowels with Bagged Decision Trees and that an error rate of 3.78% is achievable using three vowels with Support Vector Machine. The result o SVM, however, should be treated with caution as there might be some overfitting involved. Nonetheless, this experiment shows that speaker recognition using vowel-to-vowel is a promising approach.

There are, however, several possible improvements. First, there are several explored features that could be used in conjunction with this approach. For instance, harmonic amplitudes and the like could be a promising feature. Second, using all the MFCCs and higher order MFCCs on all vowels increase the feature dimension very rapidly. Performing hand-selection of the cepstral coefficients for each vowel could effective reduce the feature dimension. Third, the use of Principal Component Analysis (PCA) proved to be useless in the system. The most likely explanation is that the variation in the data does not translate directly to inter-speaker variation. Thus, Fisher Discriminant Analysis could be more useful for dimensionality reduction. Fourth, the Support Vector Machine (SVM) current uses generic Gaussian Kernel. Other types of kernel could provide more useful results. Moreover, other popular models such as Gaussian Mixture and Convolution Neural Network remains unexplored. Thus, there is a multitude of possible future directions.

8. References

[1] L. Rabiner and R. Schafer, *Theory and Applications of Digital Speech Processing*, 1st ed. Upper Saddle River, NJ, USA: Prentice Hall Press, 2010.

[2] M. Sahidullah and G. Saha, "Design, analysis and experimental evaluation of block based transformation in mfcc computation for speaker recognition," *Speech Commun.*, vol. 54, no. 4, pp. 543–565, May 2012. [Online]. Available: http://dx.doi.org/10.1016/j.specom.2011.11.004

[3] V. Tyagi and C. Wellekens, "On desensitizing the mel-cepstrum to spurious spectral components for robust speech recognition," in *Proceedings. (ICASSP '05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.*, vol. 1, March 2005, pp. 529–532.

[4] M. A. Turk and A. P. Pentland, "Face recognition using eigenfaces," in *Proceedings. 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Jun 1991, pp. 586–591.