Design and Implementation of Speaker Similarity Estimation System based on UCLA Variability Database

Yucong Wang[1], *Jingjing Zhang*[1], *Guanqun Yang*[1], *Zhengtao Zhou*[1]

[1]Department of Electrical and Computer Engineering
University of California, Los Angeles, US
yucongwang@g.ucla.edu, jinzh469@student.liu.se
guanqun.yang@engineering.ucla.edu, zhengtaozhou@ucla.edu

Abstract

As a pivotal part of automatic speaker recognition system, similarity measure and estimation of two speech segments directly determine the system performance. However, when utterance is short or the speech is polluted by noise, the accuracy of identification and verification degrads. In this paper, we tried several traditional methods and also propose a novel framework to resolve the issues related to short time duration and noise addition.

In our system, Dynamic Time Warping (DTW) algorithm is employed to measure similarity between the two input speech signals, then the resulting measures are used to find the decision threshold. With the help of threshold, the incoming two speech segments could be directly compared with each other and thereby making speaker similarity measure available.

Our system yields satisfying performance with both FPR and TPR less than 15% under clean conditions and less than 35% under noisy conditions, where 10dB babble noise is added to the original speech.

Index Terms: speech recognition, similarity estimation, Dynamic Time Warping (DTW)

1. Introduction

The objective of this project is to find a set of acoustic features and algorithms that predict whether two speech segments are uttered by the same speaker or not, which represented by 0 or 1. The database we use includes 50 male speakers saying the same sentences, “Help the woman get back to her feet.” Generally, there are two main phases of a speaker identification system. The first phase is the training system. A training system collects voice features of speakers and builds speaker models to represent the speaker specific information conveyed in the feature vectors. Many different modeling techniques have been applied to speaker recognition problems, including nonparametric and parametric approaches. Two popular and successful methods are Gaussian Mixture Models adapted from a Universal Background Model(GMM-UBM) amd Support Vector Machines using GMM SuperVectors (SVM-GSV). The features we use include MFCCs, LPCs, LPCCs, SSC, Log-filterbanks and F0.

2. Background and Related Work

Speech processing and speaker recognition are very related to our daily life and have a broad range of applications. It can be used as access control for physical facilities or computer networks and websites. The voice identification of rightful users can prevent the entrance of outsiders, which is more reliable than key or password. Another important application is transaction authentication. Voice authentication exhibited its superior compared to password or verification code since it is nearly impossible to copy. [1] Different speakers can be identified through their speech because they have different vocal tract shapes, larynx sizes, and other parts of their voice production organs. Beside the physical differences, the manner of speaking of each speaker characterizes their speech, including the use of a particular accent, rhythm, intonation style, pronunciation pattern, choice of vocabulary and so on.

In general, speech identification involves two part of work, which are feature extraction and speaker modeling. Speech signal has many features but not all are important for speech recognition. There are some features which are commonly used in speech identification which are pitch, formants, MFCCs and so on. However, there is a trade-off in those features between accuracy and robustness. After extracting feature vectors, speaker model can be trained and tested for its accuracy. Classical speaker models can be divided into template and stochastic models.[2] For template model, its basic idea is that difference between feature vectors represents the similarity. Vector quantization (VQ) and Dynamic Time Warping (DTW) are two commonly used example of template model. For stochastic model, it calculate each speaker probability and evaluate likelihood in the model. Gaussian Mixture Model(GMM) and hidden Markov model are two popular methods belong to stochastic model. In addition, support vector model (SVM) has recently been used in speaker identification because of its powerful strength in binary classification. Artificial neural network has been recently used in speaker recognition. Apart from feature extraction and speaker models, there is another important method has been used which is supervector, such as GMM supervector and i-Vector. After supervector being invented, it is used in many classical methods, which achieves great success in speaker identification.

3. Data Preprocessing

In order to better analyze the speech signal, first we need to remove the silence between the segment of the waveform. We want to focus more on the speech signal itself instead of the habit of the speakers, thus we need to reduce the effect of pause and silence. Signal energy and spectral centroid is used as thresholding in order to detect the speech segments. [3]

3.1. Silence removal

The silence remove process includes 4 steps in general: 1) Extract two features from the speech signal. 2) Set two dynamically thresholds for it. 3) Generate a thresholding criterion. 4) Detect the speech segments using the criterion.

3.1.1. Signal Energy and spectral centroid extraction

In order to extract the feature sequence, the signal is first broken into non-overlapping short term windows (frames) of 45 milliseconds length. For each frame, two features below are calculated.

- Signal energy: Let $x_i(n)$, $n = 1, ..., N$, the audio samples of the i^{th} frame, of length N. Then, for each frame i the energy is calculated according to the equation: $E(i) = \frac{1}{N}\sum_{n=1}^{N}\left|x_i(n)\right|^2$. This simple feature can be used for detecting silent periods in audio signals, but also for discriminating between audio classes.
- Spectral centroid: The spectral centroid, C_i is defined as the center of "gravity" of its spectrum. This feature is the measure of the spectral position, which high values corresponding to "brighter" sound. Experiments have indicated that the sequence of spectral centroid is highly variated for speech segments. [4]

3.1.2. Speech segments detection

After the two feature sequences are computed, the following steps are taken to compute the threshold.

1. Compute the histogram of feature sequences' values.
2. Apply a smoothing filter on the histogram.
3 .Detect the histogram's local maxima.
4. Let M_1 and M_2 be the positions of first two local maxima. The threshold is computed by $T = \frac{W \cdot M_1 + M_2}{W+1}$, where the weight is chosen to be 8.

The above process is executed for both feature sequences, leading to two thresholds: T1 and T2, based on the energy sequence and the spectral centroid sequence respectively. As long as the two thresholds have been estimated, the two feature sequences are thresholded, and the segments are formed by successive frames for which the respective feature values (for both feature sequences) are larger than the computed thresholds.

3.2. Denosing

In order to better analyze the audio files with 10dB babble noise, we would do the speech enhancement, which will reduce the noise without distorting the original (clean) signal. Adaptive Wiener filter with Two Step Noise Reduction (TSNR) and Harmonic Regeneration Noise Reduction(HRNR) methods are used to enhance the noisy speech signal [5].

3.2.1. Two Step Noise Reductions (TSNR)

Two Step Noise Reduction approach is used to refine the priori SNR estimation. Drawback of the Decision Directed approach is removed by using the TSNR, and retains the main advantage of the Decision Directed method. Decision Directed method will lower the musical noise level. The main advantage of TSNR method is the frame delay bias.

3.2.2. Harmonic Regeneration Noise Reduction (HRNR)

A characteristic of harmonics, which is present in the speech, is considered for this method. The output obtained from the TSNR method is further used in HRNR method by creating the artificial signal to regenerate the missing harmonics which was present in input signal. Then by using the artificial signal suppression gain is calculated. This helps to store all the harmonics which is present in the clean speech signal.

3.2.3. Results

Table 1: *Results on denosied system*

features: MFCC+LPC

Train	Test	FPR	FNR
clean	clean	28%	44%
clean	babble	12%	74%
multi	clean	19%	66%
multi	babble	41%	28%

(Result is performed on DTW based system.)

Even though this method can work theoretically and the effects are pretty good by human hearing, the model performs bad on the result for this project, the reason might be that it removed some useful information of speech. And we will not apply the denoise method in the following discussion. We just apply the first silence removal method as the pre-processing part, which is also helpful to control noise in some degree.

4. Feature Selection and Extraction

Since the similarity between speakers are expected to be measured and estimated, some common aspects of speech should be captured and this lead to the problem of feature selection and extraction, which we will discuss in this part.

4.1. Feature Selection

As is pointed out by both Nolan [6] and Wolf [7] independently, an ideal speech parameter should generally have following traits:

- low within speaker variability and high between-speaker variability
- resilient to attempted disguise and mimicry
- high frequency of occurrence in relevant materials
- robustness in transmission
- relatively easy to extract

These traits serve us a guideline for us to find and compare speech features available. Furthermore, according to well-recognized speech feature taxonomy, the speech features could be divided into multiple categories, which could better serve our needs for finding representative features.

Based on whether or not a feature provides information of speakers themselves, it could be either categorized into high-level features or low-level features. When considering the transmitter side and receiver side of speech production and transmission respectively, the speech features could be in turn categorized into auditory features or acoustic features. Finally, depending on the availability of the prior knowledge of the

language, they could further be considered linguistic or non-linguistic, where either of them could belongs to the category of acoustic or auditory. [8]

In this paper, due to the sparse availability of speakers' characteristics, including accents, dialects, speaking rate and speaking style, made available by the database we use and the requirement for robustness and genericness of our system, low-level and non-linguistic features are considered, providing our system with the ability to handle multiple languages and speakers with high variability. Specially, the speech features we adopt include fundamental frequency, formants, MFCC (Mel-frequency Cepstrum Coefficients), LPC (Linear Predictive Coefficients).

4.2. Feature Extraction

4.2.1. Fundamental and Formant Frequency

Fundamental frequency and formant frequency are two features that describe the principal properties of humans' speech production system. Fundamental frequency is of most interest when developing speech-related systems since it describes the periodicity of voiced region of speech and it remains almost constant when speaker utters sentence under normal conditions. Under same conditions as measuring fundamental frequency, formant frequencies describe resonance frequencies of the vocal tract tube during the utterance of speech. These two features combine to provide an overview of a given speech segment.

In this paper, the extraction of fundamental and formant frequencies is completed using publicly available software VoiceSauce [9], which in turn utilizes Straight and Snack as its backend to make possible the pitch and formant estimation.

4.2.2. MFCC

MFCC is one of the most frequently used speech features in the automatic speech recognition system (ASR). It tries to mimic the humans' speech production and perception system in two ways. One is conversion of original speech's Fourier transform into Mel-scale, which mimics cochlea's responses to sounds, while the other is the logarithm transformation of the filter bank energy, which models the nonlinear relationship between the speech energy and perceived loudness. A diagram describing the computation of MFCCs is shown below.

In this paper, the MFCC features are extracted through open-source software provided by Jame Lyons, which is accessible at the software repository hosted on GitHub (https://github.com/jameslyons/matlab_speech_feature).

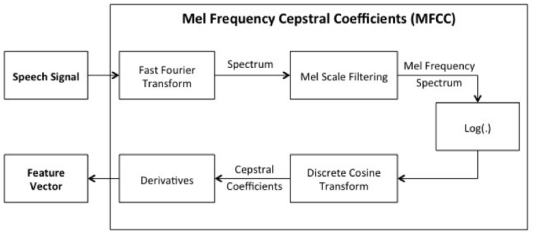


Figure 1: *Block diagram of MFCC computation*

4.2.3. LPC

Linear predictive coding is a generic method to approximate the linear system with a set of coefficients and thereby attaining the data compression and making possible the bandwidth-aware signal transmission. When applied to the field of speech processing, LPC become most useful features to describe the properties of a given speech segment. Two types of LPC exist with different computational complexity, one considers both the vowels and consonants while the other one only considers vowels, where vowels correspond to poles and consonants correspond to zeros of the linear time-invariant system. In practical application, the first type is widely applied, where a all-pole system is modeled for system synthesis, to meet the tradeoff between the model complexity and approximation accuracy.

Similar to MFCCs, the LPC features used in this paper are also extracted using James Lyons' software.

5. GMM Based System

Once the audio segments are converted to feature parameters, the next task is to decide the similar metrics between speakers. In this scheme, we treat the likelihood between GMM model and feature set as the similarity metrics.

5.1. GMM model

When we consider speaker modeling, the model must provide means of its comparison with an unknown utterance. A modeling method is robust when its characterizing process of the features is not significantly affected by unwanted distortions, even though the features are. Ideally, if features could be designed in such a way that no intra speaker variation is present while innerspeaker discrimination is maximum, the simplest methods of modeling might have sufficed. In essence, the non-ideal properties of the feature extraction stage requires various compensation techniques during the modeling phase so that the effect of the nuisance variations observed in the signal are minimized during the speaker-verification process.

Most speaker modeling techniques make various mathematical assumptions on the features, for this part we applied Gaussian distributed model to model the feature distribution and fit original signal, also known as GMM.

GMM-based identification system is first proposed by Reynolds et al.1995 [10]. The proposed method is to modeling features using GMMs, computing similarity using feature likelihood, which is exactly which we implemented in our system.

$$p(z|\lambda) = \sum_{i=1}^{M} \alpha_i N(z; \mu_i, , \sum_i)$$

z :feature vector

λ :speaker model

N:Gaussian function with mean vector μ and covariance matrix $\sum$.

α_i : the component density,

M: the number of mixtures.

We use the function fitgmdist that implemented in Matlab Toolbox to build GMM model.

$$GMMmodel = fitgmdist(featureDict, num\ of\ distributions)$$

5.2. Likelihood: posterior probability

We calculated the posterior probabilities of each component in the Gaussian mixture distribution to identify the likelihood between GMM model and feature set. Here we use the function that is implemented in Matlab Toolbox to calculate the posterior probabilities of each component in the Gaussian mixture distribution [11], in order to measure the similarity of GMM model and feature set.

$$P = posterior(object, X)$$

The function returns the posterior probabilities of each of the k components in the Gaussian mixture distribution defined by $object$ for each observation in the data matrix X. X is n by d, where n is the number of observations and d is the dimension of the data. $object$ is an object created by $gmdistribution$ or $fitgmdist$. P is n by k, with $P(i,j)$ the probability of component j given observation i.

5.3. Classifier: SVM

After decided the similarity metrics, we design a classification algorithm to train our model. One of the various methods frequently used is support vector machines (SVMs), which will be discussed next.

SVMs [12] are one of the most popular supervised binary classifiers in machine learning. In [13], it was observed that GMM supervectors could be effectively used for speaker recognition or verification using SVMs. The supervectors obtained from the training utterances were used as positive examples while a set of impostor utterances were used as negative examples. However, using GMM supervectors with SVM provided the most effective solution. The traditional GMM-SVM method was first proposed by Campbell et al. 2006.[14]The proposed method is using GMM supervector as utterance features, classify using SVMs.

An SVM classifier aims at optimally separating multi dimensional data points obtained from two classes using a hyperplane (a high dimensional plane). The model can then be used to predict the class of an unknown observation depending on its location with respect to the hyperplane. Given a set of training vectors and labels (x_n, y_n) for $n \in \{1, ..T\}$, where $x_n \in R^d$ and $y_n \in \{-1, +1\}$.

The goal of SVM is to learn the function $f : R^d -> R$ so that the class label of an unknown vector x can be predicted as

$$I(x) = sign(f(x))$$

For a linearly separable data set[15], a hyperplane H given by $w^T x + b = 0$, can be obtained that separates the two classes, so that

$$y_n(w^T x + b) \geq 1,\ n = 1, ...T$$

In our problem, in order to combine the effectiveness of adapted GMM as an utterance model and the discriminating ability of the SVM. we use SVM as a classifier to train a support vector machine model for two-class (binary) classification on a predictor data set. The train features are the likelihood matrix we got, and the train target values are the labels in which represent same or not speakers.

We use fitcsvm function that implemented in Matlab Toolbox to build the model.

$$model = fitcsvm(Likelihood\ Matrix, labels)$$

However, the problem is that SVM is working for low-through-moderate dimensional data set, actually we have a pretty high dimension matrix now. Then we applied Singular Value Decomposition(SVD) to perform a singular value decomposition of matrix to reduce dimensions.

5.3.1. SVD

In linear algebra, the singular-value decomposition (SVD) is a factorization of a real or complex matrix. It is the generalization of the eigendecomposition of a positive semidefinite normal matrix, for example, a symmetric matrix with positive eigenvalues to any $m \times n$ matrix via an extension of the polar decomposition, which has many useful applications in signal processing. [16]

5.4. Results

The results are shown in the table below.

Table 2: *Results on GMM based system*

features:MFCC+LPC

Train	Test	FPR	FNR
clean	clean	5%	90%
clean	babble	13%	71%

From the results we can observe that it's a bad prediction. GMM is widely used for speaker recognition but we didn't have enough data for each person, actually we only have 5 utterances per speaker, and it might be one of the reasons we failed to have great prediction on this system. As for SVD, it could work theoretically, but not quite satisfying for our situation, because SVD might ignored feature importance and uniqueness.

6. Neural Network Based System

Over the past few years, neural network has great success in many fields, and it is also applied to speech recognition. And in some cases, it is proved to be great improvement as to the traditional method. GMM-UBM is a popular method which widely used in speech recognition due to it combines the claimed speaker model and the alternative speaker model. It is more likely to learn accepting and rejecting decision during training. For test list, the test speech compare with each GMM and the highest score is the most likely speaker. However, even though GMM has all those advantages, its true underlying structure is low-dimension which is insufficient for the high dimensional features extracted from windows.[17] Neural network has many layer and hidden units with nonlinear function, which means it has the potential to fit high-dimensional models of data.

6.1. Neural network model

Neural network is a bunch of hidden neurons connect inputs and outputs as shown in Fig.x. Each neuron has an activation function used to produce a output with respect to the received input. The connection between one neuron output and another neuron has a weight. During learning progress, the network

modifies its weights and activation function thresholds to minimize cost function and produce favorable output.

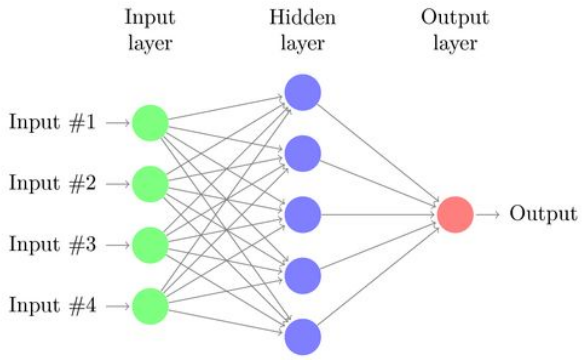


Figure 2 : An example of neural network

Firstly, we need to preprocessing for the inputs of the network, since neural network input should be a matrix, but after extracting features, each speech file transfer into a vector cell contains high-dimensional features. Since the number of windows in each speech file is different, the dimension of features also different, and we cannot find a effective way to treat for this problem, so we take means for features in all windows to make alignment. Up to now, each speech files turn into a feature vector with the same dimension.

6.2. Similarity metrics: vector distance

Then, we use vector distance to represent the difference between speech files. We tried three vector distance, which are euclidean distance, cosine distance and correlation distance. Apart from that, we also tried the vector difference between feature vectors in neural network.

To test the effect of different distance representation on different features, we first use threshold method to train model and predict outputs for test list and compare the FNR and FPR. However, we found using vector distance to train our model has very bad result. It is a reasonable result since vector distance adds up all high-dimension features which arise mistakes. For example, we use 13 MFCCs, but after calculating distance, we get one single number, and it cannot tell the difference from each MFCC. Therefore, it is not right to use vector distance as inputs to train our model. And finally, we choose to use vector difference to train our model.

At first, we use default parameters to train our model, the network will always predict two files as different speaker. Then we adjust some parameters to make neural network suit for our training data set. In our practice, we found linear function has a better performance than logistic function. It can be explained by the linear function only account for positive input. And we will discuss it later in the result of neural network.

6.3. Overfitting Compensation

Since the network has weighted connections between output and input of next layer, adding weight to the final layer compensate for the huge gap of number of 0 and 1 output. For the huge unbalanced training data set, we first make subset for training data set rather than training the whole data set. Since our data set has few 1s, it is easily for network to be overfitting.

In order to reduce the effect from this problem, we try some method like using large penalty in proportion to their squared magnitude and stop learning when performance on a subset starts getting worse[18]. We also put some random weights in the initial stage to prevent the same gradient in the later layers. Up to now, we have set proper parameters for our network. Then we use vector difference between two feature vectors extracted from any two speech file as network input. After trained our neural network, we apply it to test list and predict outputs. The results are trained by clean training dataset and test on clean test set.are shown in table. Apart from that, we also tried SVM to compare with neural network as shown in table.

6.4. Results

Table 3: *Results on neural network system*

Features	MFCCs	LPCs	LPCCS	MFCCs/ LPCs
FPR	5.6%	3%	6.4%	10%
FNR	64.4%	82%	73.3%	63%

(Train on clean set and test on clean set)

Table 4: *Results on vector difference- SVM based system*

Features	MFCCs	LPCs	LPCCS	MFCCs/ LPCs
FPR	1.43%	2.25%	2.19%	3%
FNR	88.1%	86.7%	94.1%	82.9%

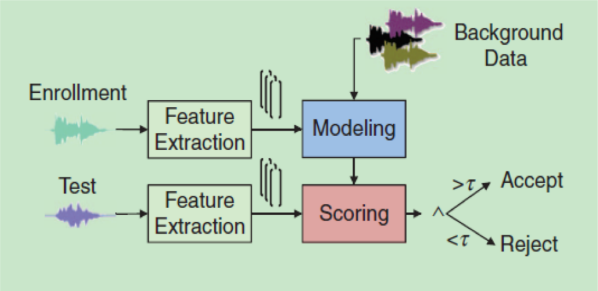
(Train on clean set and test on clean set)

From results shown above, even though the final in this approach is not favorable, however, we do see some improvement by using neural network. Since FNR is much larger than FPR, it indicates we still suffer the effect of unbalanced training dataset, which needs us to make more effort. In addition, taking means for high-dimensional features from windows is also an inappropriate way because for a sentence features may differ greatly in different windows. Taking average to make dimension alignment decrease difference between two speech files and arise mistakes for the following training part.

7. DTW Based System

When we look back on the project statement, it is a decision or verification problem rather than speaker identification. What we are going to implement is a Automatic Speaker Verification (ASV), which is to determine whether a given pair of utterances are from the same speaker or two different speakers (binary decision).

In this scheme, we use dynamic time warping distance as similarity metrics.



[FIG4] An overall block diagram of a basic speaker-verification system. (Hansen and Hasan, 2015)

Figure 3: *Overall diagram of ASV system[19]*

7.1. Dynamic Time Warping

Dynamic time warping is method that could be applied on signals with different time duration. It could measure a distance-like quantity between 2 sequences, and find the optimal match. It was widely used in time series classification. Dynamic Time Warping (DTW) is certainly the most relevant distance for time series analysis. Such relevance has been evidenced by a large body of experimental research showing that, for instance, the 1-nearest neighbor DTW (1-NN-DTW) algorithm frequently outperforms more sophisticated methods on a large set of benchmark datasets.[20]

Euclidean distance (ED) [21] is the most established distance measure between time series. The ED measures the dissimilarity between time series comparing the observations at the exact same time. For this reason, the ED can be very sensitive to distortions in the time axis. Many applications require a more flexible observation matching, in which an observation of the time series x_i at time i can be associated to an observation of the time series y_j at time $j \neq i$.

The DTW distance achieves an optimal nonlinear alignment of the observations under boundary, monotonicity and continuity constraints. DTW is usually calculated using a dynamic programming algorithm. The Equation below describes the initial condition of the algorithm.

$$dtw(i,j) = \infty, if\ i\ = 0\ or\ j = 0$$
$$= 0, if\ i\ =\ j = 0$$

Equation below presents the recurrence relation of DTW algorithm.[22]

$$dtw(i,j) = c(x_i,y_j) + min\ \{dtw(i-1,j), dtw(i,j-1), dtw(i-1,j-1)\}$$

where $i = 1...N\ and\ j = 1...M$

and $c(x_i,y_j)$ is the cost of matching two observations $x_i\ and\ y_j$, usually calculated with squared Euclidean distance.

The resulting value in dtw(N,M) is the DTW distance between x and y. Thus, the algorithm iteratively fills an array with the lowest accumulated cost for all alignments to each pair of observations to be matched. The figure below shows an example of the optimal non-linear alignment found by this algorithm and how it is represented in the DTW calculation matrix.

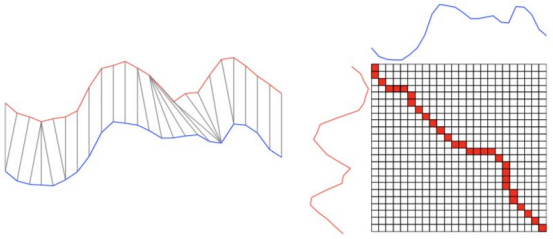


Figure 4: *optimal non-linear alignment and the matrix obtained by the dynamic time warping algorithm*

In order to improve the efficiency of DTW calculations, the use of warping windows is common [6, 3]. Warping window, or constraint band, defines the maximum allowed time difference between two matched observations. From the algorithm standpoint, this technique restricts the values that need to be computed to a smaller area around the main diagonal of the matrix.[23]

However, the exact window size that would provide the best results for a dataset is data dependent. Outside classification problems with 1-NN, there are no clear guidelines to set this parameter and possibly the best approach is to evaluate the results for several window sizes.

7.2. Similar Metrics: DTW distance

In our problem, we just utilize the function implemented in the Matlab Toolbox that shows DTW distance as similar metrics.[24]

$$dist = dtw(x,y)$$

It stretches two vectors, $x\ and\ y$, onto a common set of instants such that $dist$, the sum of the Euclidean distances between corresponding points, is smallest. To stretch the inputs, dtw repeats each element of $x\ and\ y$ as many times as necessary. If $x\ and\ y$ are matrices, then dist stretches them by repeating their columns. In that case, $x\ and\ y$ must have the same number of rows.

7.3. Classifier: threshold

Based on the superiority of DTW, we just use the function which calculated threshold by Equal Error Rate as a classifier. So that we found a distance-like threshold. and if the error rate is not higher than that, it was justified to be same speaker talking.

We use the function that already implemented in the sample program to get the threshold.

$function\ [eer, threshold]\ =\ compute\_eer(scores,\ labels)$

7.3.1. Equal Error Rate

The equal error rate (EER) [25] is defined as the FPR and FNR values when they become equal. That is, by changing the threshold, we find a point where the FPR and FNR become equal. The EER is a very popular performance measure for speaker-verification systems. Only the soft scores from the automatic system are required to compute the EER. No actual hard decisions are made. It should be noted that operating a speaker-verification system on the threshold corresponding to the EER might not be desirable for practical purposes. For high-security applications, one should set the threshold higher, lowering the false errors at the cost of miss errors. However,

for high convenience, the threshold may be set lower. On the contrary, for an automated customer service, denying a legitimate speaker will cause inconvenience and frustration to the user. In this case, accepting an illegitimate speaker is not as critical as in high-security applications.

7.4. Results and Analysis

The results are shown in the table below. We tried several combinations of these features and try to find the optimal result.

Table 5: *Results on DTW based system*

features: MFCC+LPC

Train	Test	FPR	FNR
clean	clean	15%	11%
clean	babble	36%	31%
multi	clean	13%	12%
multi	babble	33%	36%

features: MFCC+LPC+F0

Train	Test	FPR	FNR
clean	clean	34%	22%
clean	babble	40%	30%
multi	clean	33%	22%
multi	babble	38%	32%

features: MFCC+LPCC

Train	Test	FPR	FNR
clean	clean	34%	56%
clean	babble	40%	35%
multi	clean	34%	56%
multi	babble	42%	36%

features: MFCC+LPC+logfbs

Train	Test	FPR	FNR
clean	clean	15%	12%
clean	babble	39%	31%
multi	clean	12%	13%
multi	babble	35%	34%

features: F0 (baseline)

Train	Test	FPR	FNR
clean	clean	34%	30%
clean	babble	46%	38%
multi	clean	32%	33%
multi	babble	43%	42%

From the results we can observe that almost all the results of different combinations of features reached the baseline and we successfully meet the basic requirement of the project. What is more, the best results are attained with feature combination MFCC and LPC. Under this situation, the FPR and FNR are improved about 20 percent on the clean set from the baseline. However, the result is not as satisfying under noise condition, where only 8-9 percent improvement is acquired. This could be explained by the result of insufficient suppression of noise and features' sensitivity to that.

7.5. Result Visualization

The ROC curves help visualize our best result.

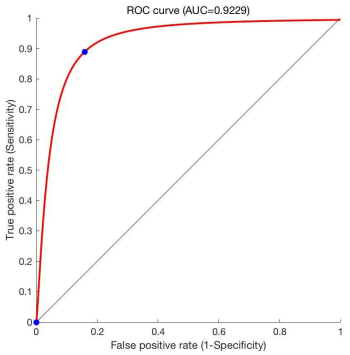


Figure 5: *Train clean and test clean set*

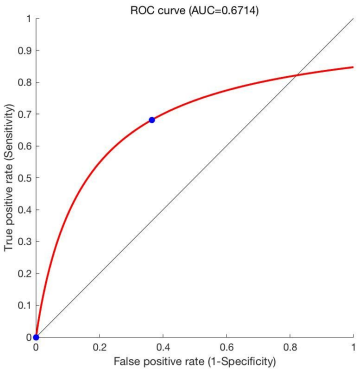


Figure 6: *Train clean and test babble set*

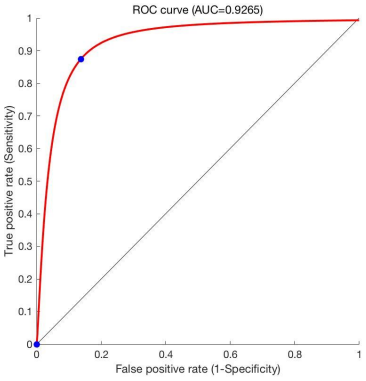


Figure 7: *Train multi and test clean set*

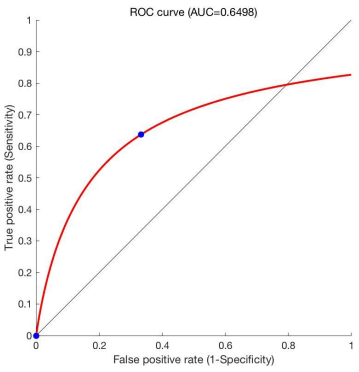


Figure 8: *Train multi and test babble set*

The receiver operating characteristic (ROC) curve can give a visual representation of the tradeoff between FPR and FNR. The area under curve (AUC) shows the accuracy of the model, when it is close to to 1, the result is generally better. As we can see from the curves, when our system is tested on clean set, the area is about 0.92, which is a definitely great result. The area reaches 0.67 under the noise condition, which is a reasonable result with potential to improve.

8. Conclusions and future work

8.1. Results and Analysis

Our method for speaker verification is robust and reliable, the average false rate under clean set is as low as 12%, which is commensurate with human-involved testing system.

Besides, the running time is quite short, the whole procedure will cost less than 2 minutes. It shows that our system is a almost real-time processing method, which has strong practicality.

There are also some existing problems in our system, for example, the performance under the noise condition is not satisfying; the feature robustness needs to be improved and the classifier does not have sufficient stability when the dataset is seriously unbalanced or is carrying extreme values.

8.2. Future work

For future work, some research could be done on noise suppression without removing useful features to identify speakers. Some modeling methods including adapted UBM and i-vector could also be explored to model the feature distribution. Besides, the intelligent method to reduce dimensions on large scale data set is worth exploring.

9. Acknowledgements

We would like to thank Prof. Abeer Alwan for her illuminating lectures, which pave the way for us to explore the world of speech processing. We would also like to thank the teaching assistant of this course - Gary Yeung, for his considerate tutoring for our coursework and insightful suggestions for our project. Finally, we would like to acknowledge anyone who has helped us during this course.

10. References

[1] Douglas A Reynolds, "An overview of automatic speaker recognition," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)(S. 4072-4075)*, 2002.

[2] Campbell, J., 1997. Speaker recognition: a tutorial. Proc. IEEE 85 (9),1437–1462.

[3] T. Giannakopoulos, "A method for silence removal and segmentation of speech signals, implemented in Matlab,"*Department of Informatics and Telecommunications, University of Athens, Greece*

[4] T. Giannakopoulos, "Study and application of acoustic information for the detection of harmful content, and fusion with visual information," Ph.D. dissertation, *Dep of Informatics and Telecommunications, University of Athens, Greece, 2009.*

[5] Shruti.O.R, Jennifer C Saldanha, "Speech Enhancement Using Filtering Techniques" in 3*rd National Conference on Emerging Trends in Electronics and Communication (NCETEC-16), September 2-6, Hyderabad, India, Proceedings*, 2018, pp. 100–104

[6] Nolan, F. J. D. The phonetic bases of speaker recognition. *Diss. University of Cambridge, 1980*

[7] Wolf, Jared J. "Efficient acoustic parameters for speaker recognition." *The Journal of the Acoustical Society of America 51.6B (1972): 2044-2056.*

[8] Hansen, John HL, and Taufiq Hasan. "Speaker recognition by machines and humans: A tutorial review." *IEEE Signal processing magazine 32.6 (2015): 74-99.*

[9] Shue, Yen-Liang. The voice source in speech production: Data, analysis and models. *University of California, Los Angeles, 2010*.

[10] Hansen, J. H. L., & Hasan, T. (2015). "Speaker Recognition by Machines and Humans: A tutorial review." *IEEE Signal Processing Magazine, Vol. 32, No. 6, pp. 74 -99.*

[11] McLachlan, G., and D. Peel. Finite Mixture Models. *Hoboken, NJ: John Wiley & Sons, Inc., 2000.*

[12] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learning, vol. 20, no. 3, pp. 273–297, Sept. 1995.*

[13] W. M. Campbell, D. E. Sturim, and D. A. Reynolds, "Support vector machines using GMM supervectors for speaker verification," *IEEE Signal Process. Lett., vol. 13, no. 5, pp. 308–311, 2006.*

[14] C. M. Bishop and N. M. Nasrabadi, Pattern Recognition and Machine Learning. *New York: Springer, 2006.*

[15] Banerjee, Sudipto; Roy, Anindya (2014), Linear Algebra and Matrix Analysis for Statistics, *Texts in Statistical Science (1st ed.), Chapman and Hall/CRC, ISBN 978-1420095388*

[16] Banerjee, Sudipto; Roy, Anindya (2014), Linear Algebra and Matrix Analysis for Statistics, *Texts in Statistical Science (1st ed.), Chapman and Hall/CRC, ISBN 978-1420095388*

[17] L. Deng, "Computational models for speech production," in *Computational Models of Speech Pattern Processing, K. M. Ponting, Ed. New York: Springer Verlag, 1999, pp. 199–213.*

[18] H. Bourlard and N. Morgan, Connectionist Speech Recognition: A Hybrid Approach, *Norwell, MA: Kluwer, 1993.*

[19] Hasan and John HL Hansen. "Acoustic factor analysis for robust speaker verification." *IEEE Transactions on audio, speech, and language processing 21.4 (2015): 842-853*.

[20] Müller, Meinard. "Dynamic time warping." *Information retrieval for music and motion (2007): 69-84.*

[21] Danielsson, Per-Erik. "Euclidean distance mapping." *Computer Graphics and image processing 14.3 (1980): 227-248.*

[22] Silva, Diego F., and Gustavo EAPA Batista. "Speeding up all-pairwise dynamic time warping matrix calculation." *Proceedings of the 2016 SIAM International Conference on Data Mining. Society for Industrial and Applied Mathematics, 2016*.

[23] Sakoe, Hiroaki, and Seibi Chiba. "Dynamic Programming Algorithm Optimization for Spoken Word Recognition." *IEEE® Transactions on Acoustics, Speech, and Signal Processing. Vol. ASSP-26, No. 1, 1978, pp. 43–49.*

[24] Paliwal, K. K., Anant Agarwal, and Sarvajit S. Sinha. "A Modification over Sakoe and Chiba's Dynamic Time Warping Algorithm for Isolated Word Recognition." *Signal Processing. Vol. 4, 1982, pp. 329–333.*

[25] Reynolds, Douglas A. "Speaker identification and verification using Gaussian mixture speaker models." *Speech communication 17.1-2 (1995): 91-108.*