Speaker Identification

E6820 Spring '08 Final Project Report

Prof. Dan Ellis

Yiyin Zhou

1. Introduction

People use biometric information to distinguish between different persons. Visually, face is one most important feature, other unique features, such as finger-prints, iris, are often used. Another way to identify a person is from the acoustic fact that each person's voice are different, this forms one area of speech processing, automatic speaker recognition. For the past few decades, many solutions have come out. Although many of them have a very good performance, none of them are perfect. The difficulty of this is caused by several reasons based on the nature of speech. First, voice is not so unique as visual cues such as finger-prints, some people's voice are very similar which results in the difficulty of separation. Another difficulty is that for speaker recognition, we are not only dealing with the variation between people, but also the huge variability of voice from one person, this includes the large amount of phonemes one can utter as well as the variation when speaking in different emotions.

Speaker Recognition can be divided by two ways. One way is to divide it into speaker verification and speaker identification. For speaker verification, the test is based on the claimed identity and a decision for accepting or rejecting is made. For speaker identification, there is no claim of identity, the system chooses the speaker from the database or in open set system, the identity can be unknown. Another way is to divide speaker recognition based on the text used for test. It can be text-dependent, which use the fixed or prompt sentence for testing, or text-independent, in which any utterance can be used. In this project, the focus is on the text-independent speaker identification in closed set.

One of the state of the art text-independent speaker identification system was proposed by D. Reynolds in 1995[2] using MFCC and GMM. This method reaches a high correct rate and performs well in several NIST speaker recognition evaluation[4] (although most of the evaluation is on speaker verification). In this project, GMM will be used and several different filter bank based cepstral methods will be compared.

2. Feature Extraction

The feature used in this project is filter bank based cepstral coefficients. Three different filter banks are used, namely, Mel-scale filter bank, uniform filter bank and non-uniform filter bank. The system diagram for feature extraction is shown in figure 1

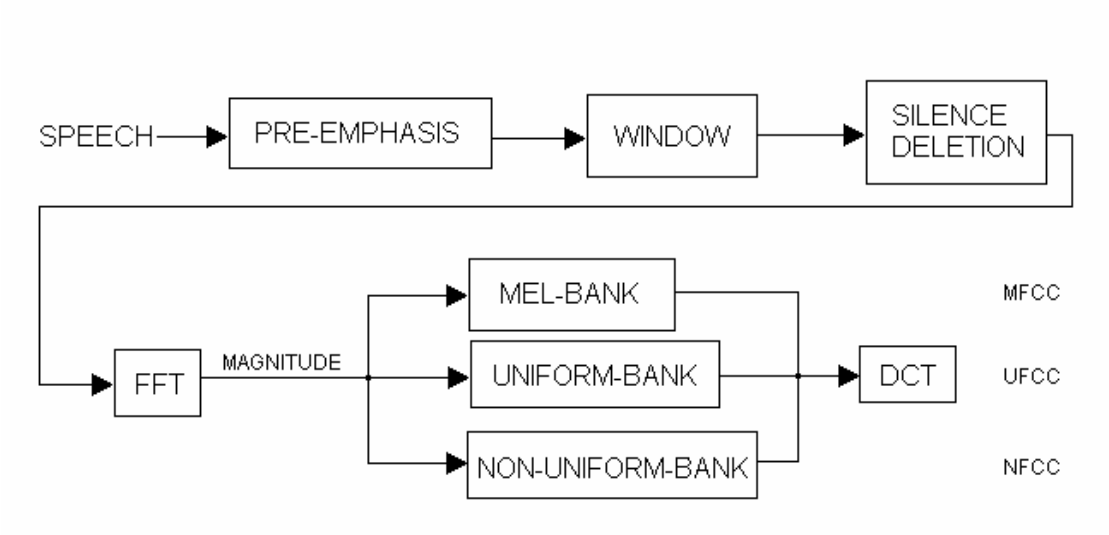


Figure 1. feature extraction diagram

The speech is first pre-emphasized by coefficient -0.97 for equal loudness process, then windowed to 32ms per frame and 16ms increase, this corresponds to 512 point and 256 points, respectively, for 16kHz sampling rate speech. A silence detection based on standard deviation of each frame is followed. The frames with standard deviation smaller than 3 times the smallest deviation in the whole speech (except zero) will be discarded. FFT will be performed on each hamming windowed frame and magnitude of the spectrum is passed through three different filter banks, each with 40 filters. Then, cepstral coefficients are generated by taking the DCT of the output from the filter banks. Based on their filter banks, the cepstral coefficients are called mel-frequency cepstral coefficients (MFCC), uniform-frequency cepstral coefficients (UFCC) and nonuniform-frequency cepstral coefficients (NFCC).

2.1 MFCC

The mel-scale is the nonlinear scale based on human perception of the different frequency content of sounds. The conversion from frequency to mel frequency here uses the following formula

$$mel = 2595 \log\left(1 + \frac{f}{700}\right)$$

The resulting mel-scale frequency bank from 0 to 8000Hz is shown in figure 2

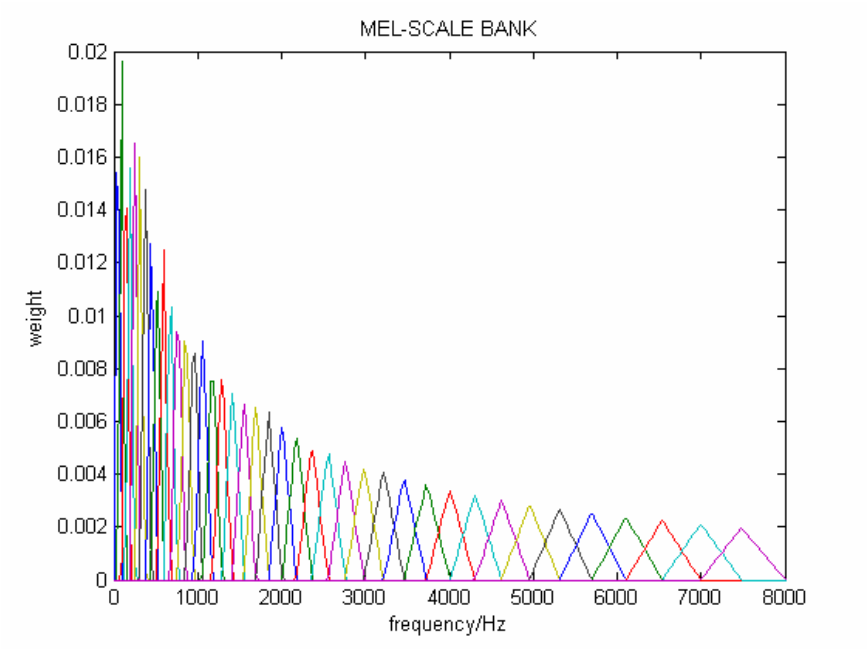


Figure 2. Mel scale filter bank

Figure 3 shows the first 14 MFCC of the same sentence spoken by 4 different persons, the first two are female, and the last two are male.

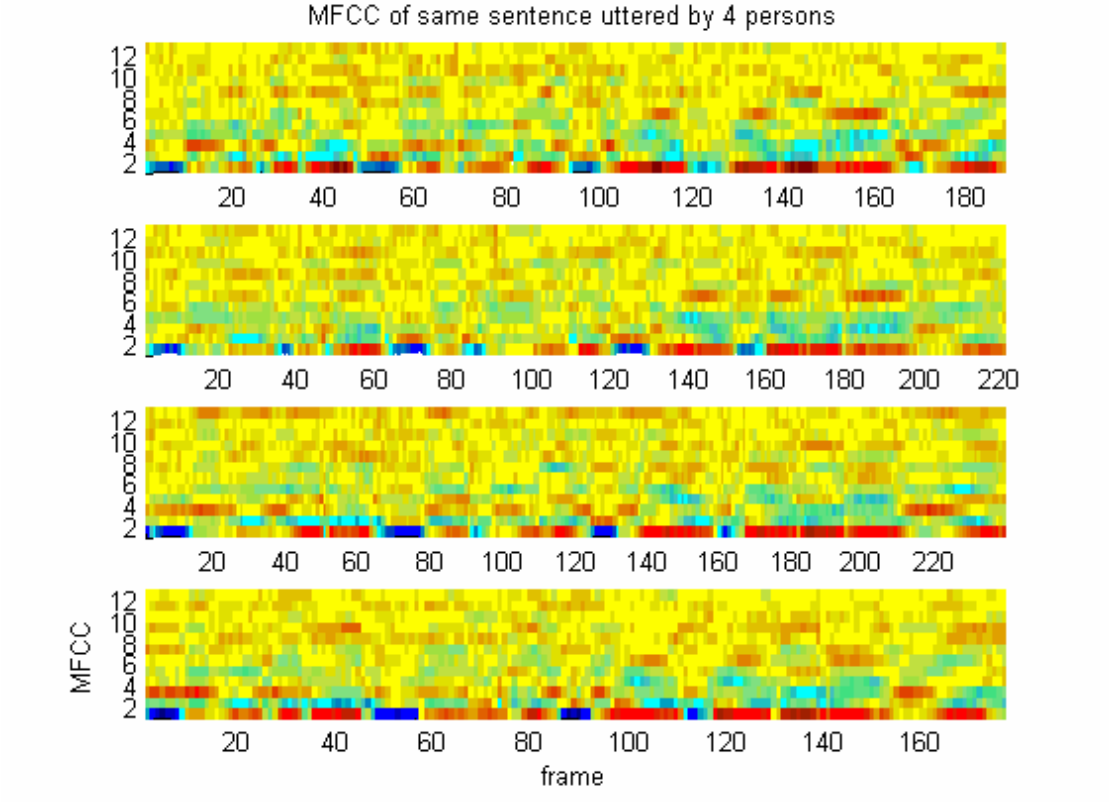


Figure 3. MFCC for four persons speaking the same sentence

As we know, the lower coefficients of the cepstrum describes the envelop of the

spectrum, which captures the vocal shape information. From these four figures, the MFCC of the four speaker shares some similar patterns. Using MFCC in speech recognition is because that the MFCC of the same phoneme or broad class are similar, while here in speaker recognition, I think efficiency of MFCC lies on another dimension. Although the MFCC for the same phoneme is similar among the speakers, they are still different. The difference also comes from the vocal shape of different speakers. The MFCC of one phoneme from one speaker tend to be more similar than others, although on a broad scale, they all look similar. So this slightly difference or 'deviation' from the common MFCC of the phoneme is what MFCC is used for in speaker recognition. Since there are a lot of phonemes, the use of GMM or Vector Quantization is powerful to distinguish between speakers. Each phonemes or acoustic class can be captured by one mixture component mean or one codebook vector, and for GMM, the covariance also captures the variations of the average spectral shape.

2.2 NFCC

Although the importance of MFCC cannot be neglected, it places a lot more emphasis on lower frequency part, this leads to some disadvantage in speaker recognition problems. As an recent research[3] shows that speaker dependent information does not only exists in low frequency parts, but also in higher frequency. In order to capture these information, non-uniform frequency bank is used.
The non-uniform frequency bank design is based on training data. Calculation of F-ratio of different frequency bands in the spectrum is performed.

$$\text{F - Ratio} = \frac{\text{inter - speaker variance}}{\text{intra - speaker variance}} = \frac{\frac{1}{M}\sum_{i=1}^{M}(u_i - u)^2}{\frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}\left(x_i^j - u_i\right)^2},$$

Where x_i^j is one subband energy of the jth speech frame of speaker i with j=1,2,...,N and i=1,2,…,M. u_i and u are the subband energy averages for speaker i and for all speakers, respectively.

The F-ratio for TIMIT corpus and the corresponding non-uniform filter bank are shown in figure 4

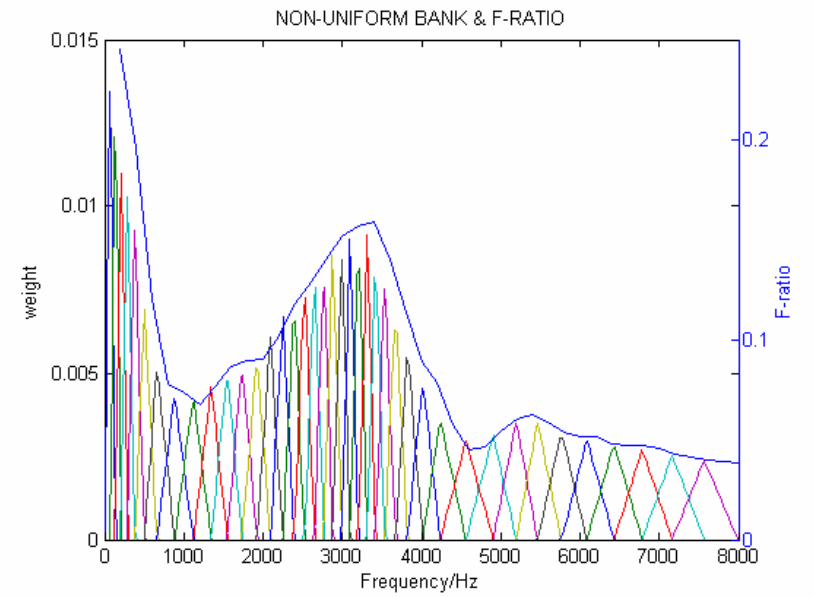


Figure 4 F-ratio of TIMIT corpus and non-uniform filter bank

The F-ratio still shows that the importance of lower frequency below 500Hz, and also the frequency range around 3000 Hz. It also shows the difference from the [3], in which higher F-ratio region is at 4000 to 6000Hz. This can caused by the language difference. The speech used in [4] is Japanese while here it is English.

2.3 UFCC

In comparison, uniform frequency bank is also used, which is shown in figure 5

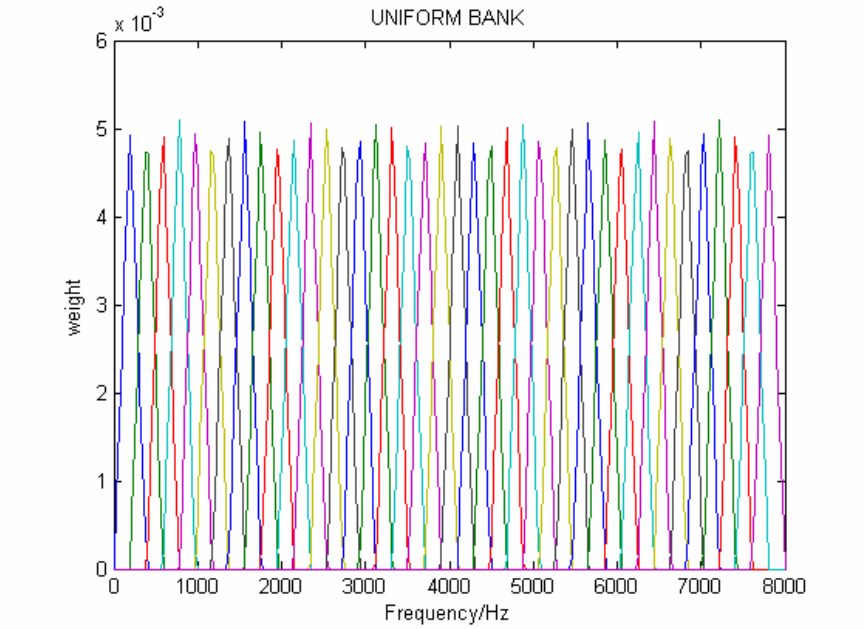


Figure 5 uniform filter bank

And surprisingly, it gives the best performance among the three filter banks, which will be shown later in the experiment part.

3. Classification Model

Gaussian mixture model is used for classification. A Gaussian mixture density is a weighted sum of M component densities given by the equation

$$p(x \mid \lambda) = \sum_{i=1}^{M} p_i b_i(x)$$

Where x is a random vector, i=1,2,…,M are the mixture components, p_i is the mixture weight and $\sum_{i=1}^{M} p_i = 1$, $b_i(x)$ are the component densities, where

$$b_i(x) = \frac{1}{(2\pi)^{\frac{D}{2}} \left|\Sigma_i\right|^{\frac{1}{2}}} \exp\left\{-\frac{1}{2}(x-u_i)'\Sigma_i^{-1}(x-u_i)\right\}$$

where u_i and Σ_i are the mean vector and covariance matrix, respectively.

So for each model, it has three parameters, p_i, u_i and Σ_i, in this project, diagonal covariance is used.

GMM can be seen as a hybrid of unimodal Gaussian model and Vector Quantization model. It does not have a hard distance as in VQ but instead using probabilities, which makes it capable of approximating arbitrarily-shaped densities. And as discussed before in the MFCC section, GMM may model some underlying acoustic classes by separating each class to a Gaussian mixture.

3.1 Training

The training of GMM model in the project uses Expectation Maximization algorithm. Given a set of training vectors $X = \{x_1, x_2, ...x_T\}$, the *a posteriori* probability for each mixture component is calculated by

$$p(i \mid x_t, \lambda) = \frac{p_i b_i(x_t)}{\sum_{k=1}^{M} p_k b_k(x_t)}$$

then, in the maximization step, new model parameters are computed according to the *a posteriori* probability, which is

$$p_i = \frac{1}{T}\sum_{t=1}^{T} p(i \mid x_t, \lambda)$$

$$\mu_i = \frac{\sum_{t=1}^{T} p(i \mid x_t, \lambda) x_t}{\sum_{t=1}^{T} p(i \mid x_t, \lambda)}$$

$$\sigma_i = \frac{\sum_{t=1}^{T} p(i \mid x_t, \lambda) x_t^2}{\sum_{t=1}^{T} p(i \mid x_t, \lambda)} - \mu_i^2$$

Using the EM algorithm requires an initial model. Because it does not guarantee to go to the global maxima, and also the training iteration are limited by some number, a good initial model will give a better recognition performance. In this project, two different initialization methods have been tried. One is to randomly select M vectors from the training vectors as mixture means, where M is the number of mixtures. The other one is that to first using k-means algorithm on the training data to get M mean vectors first. The k-means algorithm is performed 5 times and the one with the smallest mean squared error is used. Both of the two methods use uniform mixture weights and covariance of all the training vector as M mixtures covariance.

3.2 Identification test

Given an observed sequence, the test is to find the speaker model which gives the maximum *a posteriori* probability. If there are S speakers,

$$Sc = \arg\max_{1 \le k \le S} \Pr(\lambda_k \mid X) = \arg\max_{1 \le k \le S} \frac{p(X \mid \lambda_k)\Pr(\lambda_k)}{p(X)}$$

Assuming equally likely speakers and observation is the same for all models, the classification rule can be simplified to be

$$Sc = \arg\max_{1 \le k \le S} \sum_{t=1}^{T} \log p(x_t \mid \lambda_k)$$

4. Evaluation on TIMIT corpus

The experiments on speaker identification are done on TIMIT corpus. I divide it into 2 evaluation sets. One contains all the training portion which has 462 speakers, called 462test, the other one contains all the testing portion which has 168 speakers, called 168test. TIMIT corpus has 2 ‘sa’ sentences which are spoken by all speakers. These two sentences are used for testing because they have the same speech information but different voices. Other 8 sentences are used for training. The statistics of the training

and testing speech of two evaluation sets are shown in table 1

Table 1 corpus statistics

	168test (avg/max/min length (s))	462test (avg/max/min length (s))
Training data (all si and sx combined)	24.6/35.7/18.5	24.5/41.5/17.4
Testing data (two sa seperately)	3.1/5.28/2.09	3.1/5.35/2.0

The number of mixtures of the GMM in the tests are M=4,8,16. This is a relatively small number comparing to common systems with 32 to 128 mixtures. This is because adding more mixtures makes it computational slower to adapt to real time environment, so the effectiveness of models with fewer mixtures are examined.

4.1 Comparison of two initialization methods

Here, two different initialization methods as mentioned before are compared with each other. For using k-means algorithm for initialization, its own initialization is done by randomly choosing M vectors, then clustered to convergence or at most 10 iterations. Then, EM is performed for at most 20 iterations, which is also the case for randomly choosing M vectors for initialization. The results are shown in figure 6, only MFCC and UFCC are shown. The blue and red lines refer to MFCC and UFCC, respectively, while solid line shows the result of performing k-means in training. We can see that using k-means for initialization do improve the performance of the recognition, especially when error rate is relatively large. Since k-means perform much faster than EM, and gives a better start up point which makes EM converge faster, it does not have a large influence on training time, although for long training data, this may need to be take into account.

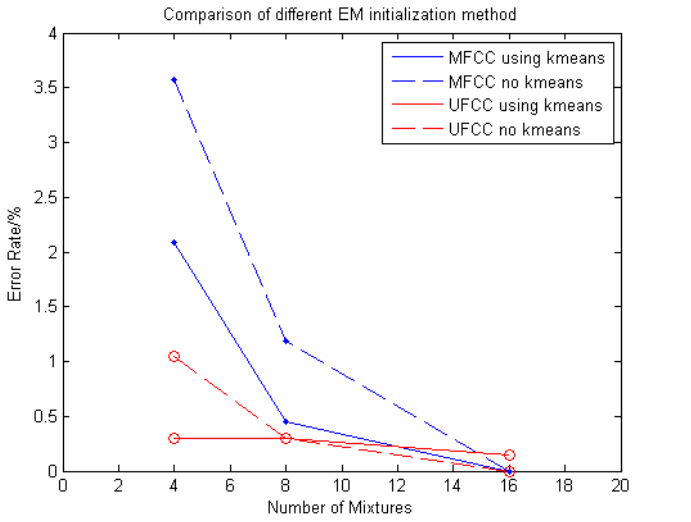
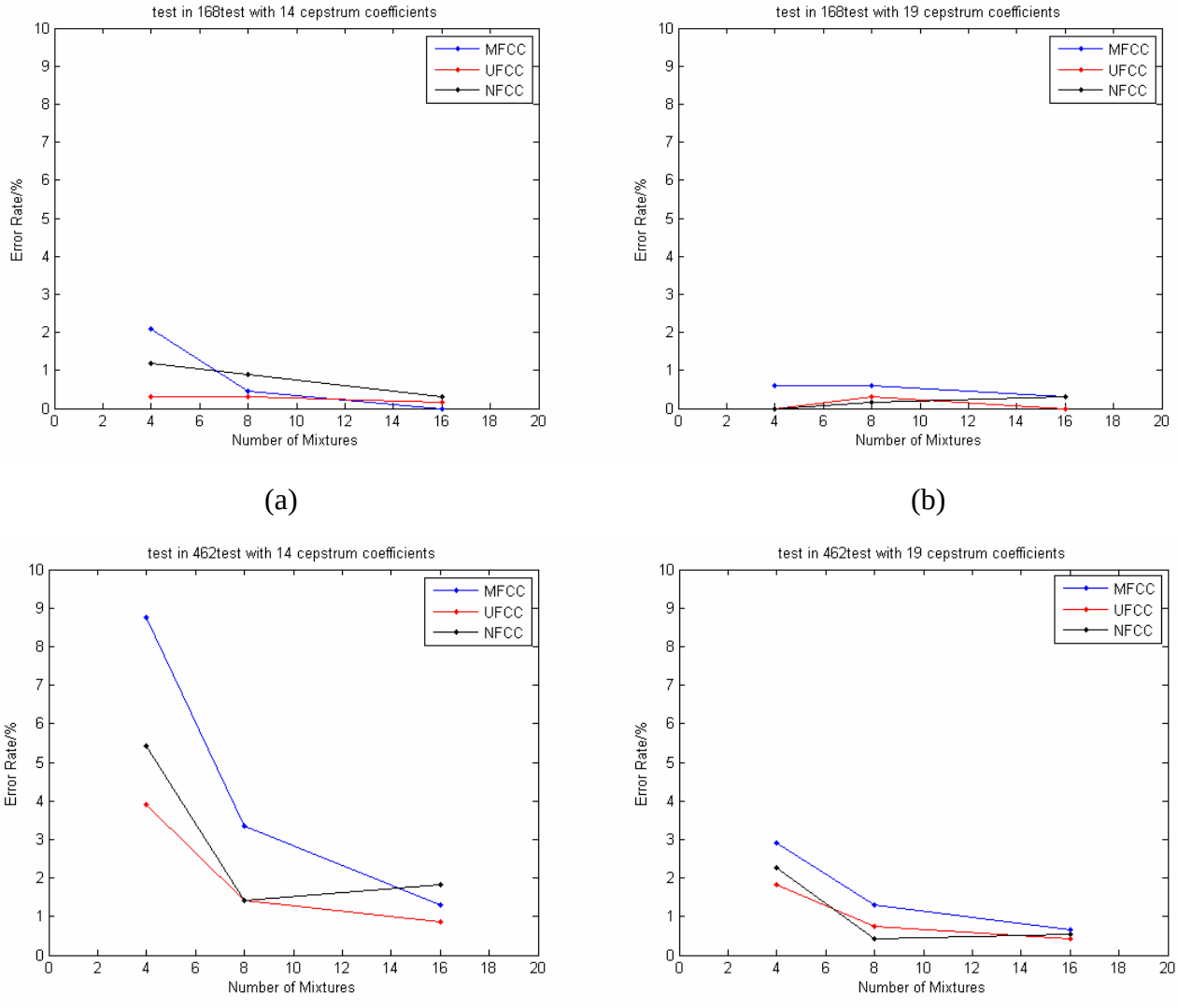


Figure 6 comparison of different initialization methods

4.2 Test results

Next, many tests are performed to observe the general performance of the system, as well as the performance of different number of mixtures, number of cepstral coefficients and population.

As mentioned above, number of mixtures being tested are M=4,8 and 16. The number of cepstral coefficients being tested are the first 15 and 20 coefficients except the c0 which is the log energy, this makes the number of cepstral coefficients to be 14 and 19. Tests are both conducted on 168test and 462test. The result are shown in figure 7, they are set to the same scale for comparison.



(a) (b)

(c) (d)

Figure 7 results of speaker identification

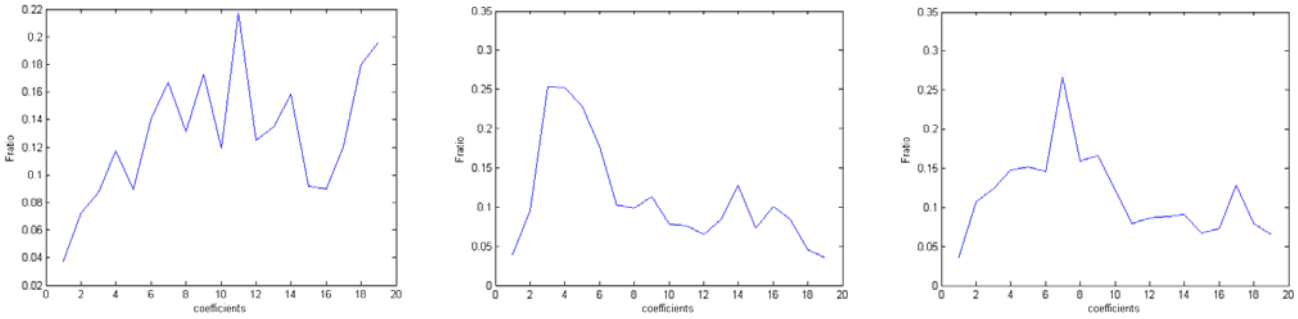
(a) test in 168test, 14 coefficients are used (b) test in 168test, 19 coefficients are used

(c) test in 462test, 14 coefficients are used (d) test in 462test, 19 coefficients are used

From the four figures, we observe several expected trends. The error rate increases when population grows, and decreases with the number of mixtures or number of cepstral coefficients increases.

In most cases, UFCC has the lowest error rate while that of MFCC is the highest. Although NFCC do perform better than MFCC, it is still slightly worse than UFCC, which, presumably, means the design of NFCC still does not capture some higher frequency speaker dependant information.

If we compute the F-ratio for each cepstral coefficients for them, which is shown in figure 8. UFCC has three coefficients which are has large F-ratio. We can get some idea why the performance is like this for three of them, although the f-ratio test is not so valid here because that they are assumed to be multimodel.



(a) MFCC (b) UFCC (c) NFCC

Figure 8 cepstral coefficients f-ratio

Put this aside, the performance of this method works pretty well. Of course, speeches in TIMIT corpus are perfectly recorded with high SNR, which makes it easy to achieve a high correct rate even with small number of mixtures. Noisy speech, or speech with background music should also be examined, but it is not discussed so far in this project and this may be involved in future works.

4.3 Test result on compressed speech

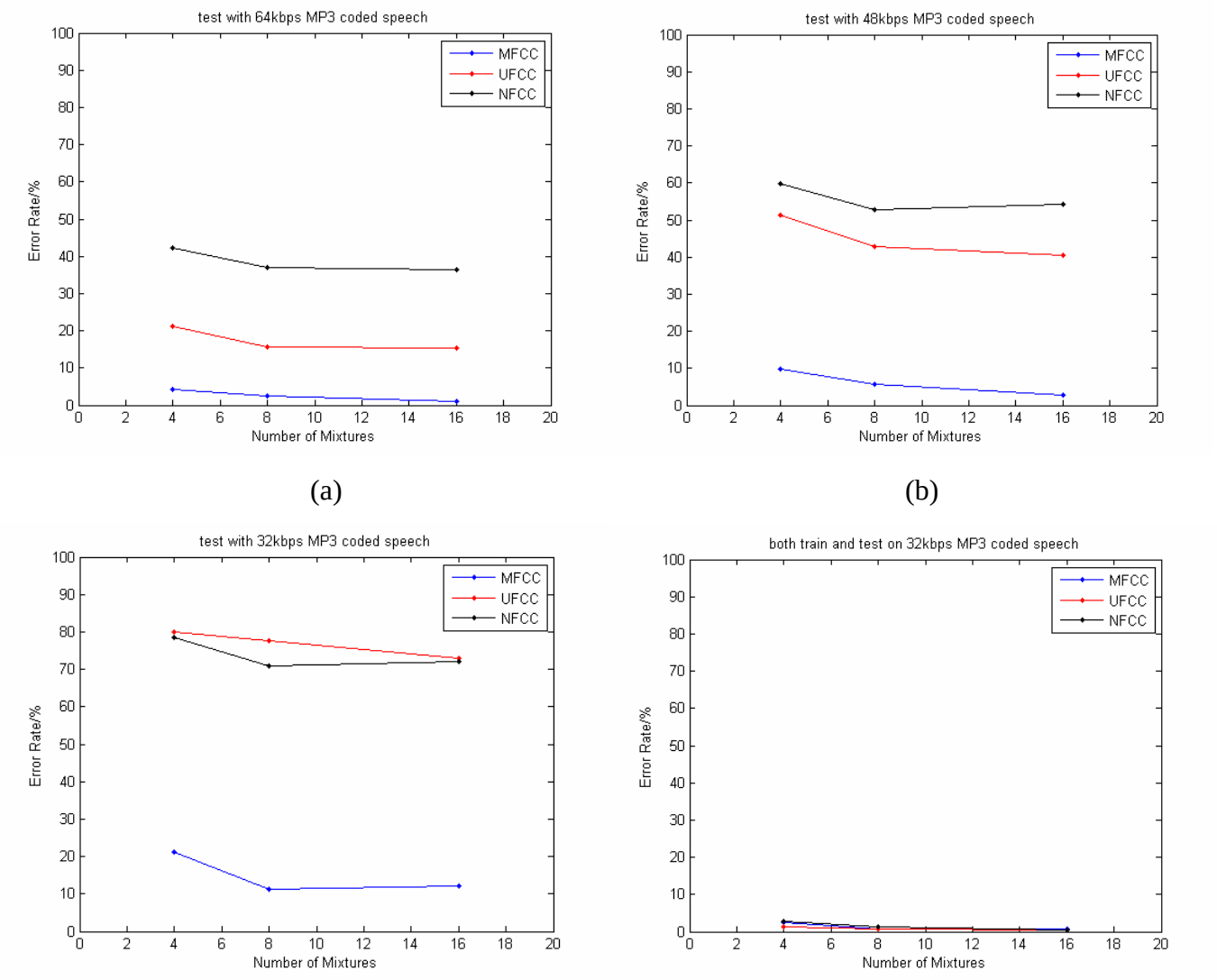
As most audio content in TV or movie are compressed, we sometimes may encounter the situation in which the training data and testing data does not come from the same compression scheme.

In order to observe the effect of compression distortion on the three cepstral coefficients, the following test is performed.

Firstly, the models for each speaker are trained with original clean speech. The testing speech are compressed by lame 3.97 into mp3 and then converted back to wav file by Switch Sound File Converter. Three different bit rates, 64kbps, 48kbps and 32kbps,

are used for increasing distortion. The compressed speeches are used for testing. Another test is to train the speech on mp3 compressed speech and then use these models to test compressed speech of the same bit rate. The bit rate for this is the worse case, which is 32kbps.

The results are shown in figure 9. Obviously, with the decrease of bit rate, all the three cepstral coefficient performs worse. However, error rate of NFCC and UFCC degrade severely whereas MFCC still got 80 to 90 percent correct rate in the worse case of 32kbps. From this perspective, MFCC is much better dealing with compression distortion than the other two. Since most phycoacoustics based compression such as mp3 tend to mask or discard higher frequency part, this can be a reason why MFCC which has larger resolution in high frequency does not suffer a lot from these compression distortions.



(a) (b)

(c) (d)

Figure 9 test results on compressed speeches
(a) 64kbps (b) 48kbps (c) 32kbps (d) train and test on 32kbps

The last figure shows the system performance is close to that of both training and testing on clean speech. This means that cepstral coefficients do not do well with unseen features which is the distorted testing data here. When trained on compressed speech and sees the distorted data, it performs much better. Hence, the same coding scheme as in testing speech is suggested to be used for training, or more training data containing different compressed data need to be used to get better recognition.

5. Real time testing

When it comes to recognizing speaker in real time speech, several new issues adds in, which makes it more difficult. First, in order to have a tolerable delay, the testing speech should be shorter. Since we are doing statistical classification, using shorter speech is less accurate. Second, we assume that the test speech before only contains only speaker's speech, while in real time recognition in conversations, a speech segment can contain several person's speech. Hence a segmentation is needed first to separate different voice, and then test on even shorter speech segment. Third, computational efficient model is preferred in real time recognition, this means the number of mixtures cannot be too big when population is large.

As shown in the testing result in the previous section that 4 to 8 mixtures gives a high accuracy as well as efficiency, it will be used in the test for 168test in real time. Also a speech segment of 2 seconds will be used for testing.

5.1 Segmentation

In a speech segment of 2 seconds, assume that there are only two cases: the speech contains one speaker, or contains two speakers consecutively (which means segmenting the speech result in two subsegments containing one speaker each). Although in a fast changing conversation, one segment could have multiple (more than 3) subsegments, it can always be simplified to two-subsegment case using a speech segment of shorter length. Apparently, a trade off should be made to choose more precise segmentation or more accurate recognition since shorter segment will give worse recognition.

To test if the 2-second speech need to be segmented into two subsegments, a BIC segmentation like scoring is performed. The score

$$\log \frac{\max(L(X_1))\max(L(X_2))}{\max(L(X_0))}$$

is calculated for each possible segmentation point, where X0, X1 and X2 denotes the

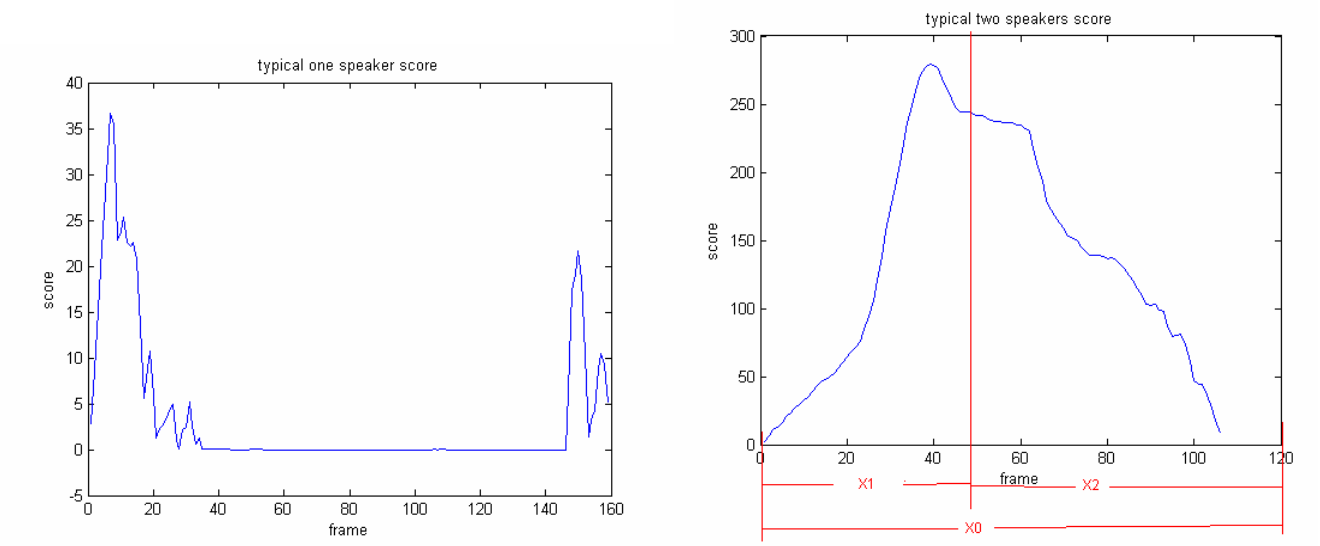
whole 2-second segment, the first and second subsegment, respectively, max(L(X)) is the maximum a posteriori probability for the given segment X. When everything is computed in logarithm, this becomes

$$\log(\max(L(X_1))) + \log(\max(L(X_2))) - \log(\max(L(X_0)))$$
$$= \max \sum_{t=1}^{l} \log p(x_t \mid \lambda_k) + \max \sum_{t=l+1}^{T} \log p(x_t \mid \lambda_k) - \max \sum_{t=1}^{T} \log p(x_t \mid \lambda_k)$$

This seems exhaustive because each possible segmentation point need to be calculated. However, when computing the *a posteriori* probability for the whole segment, the probability of each frame is calculated, and L(X0) is the sum of logarithm of all the frames, while L(X1) and L(X2) are the sum of logarithm of frames in their corresponding subsegments. This means the score calculation only requires some extra additions, which is still efficient.

If and only if the maximum of the probability of X0, X1 and X2 are all chosen from the same model k, the score will be zero, if they are chosen from different models, the score will be larger than zero, since there is a larger score from other models than the score of the whole segment.

The typical one speaker segment and two speakers segment are shown in figure 10



(a) typical one speaker score (b) typical two speakers score

Figure 10 typical score pattern of one speaker and two speakers

For one speaker score, the maximum of score is usually small, while that of two speaker segment is large, and the peak shows the segmentation point. A threshold can be set to separate two types of segments.

It is also interesting to look at the score for one speaker segment. Ideally, if the recognition for even one frame is always correct, figure 10(a) should be zero everywhere. The non-zero values appeared on the two sides is caused by error recognition using too small amount of frames or too short length of speech. On the

other hand, the zeros in the middle indicate that if the recognition for the whole segment is correct, the recognition for the two subsegments is also correct. That means using around 40 frames for testing will give a high correct rate, and 40 frames correspond to about 0.7s in 16kHz sampling rate.

If the score shows the segment need to be divided into two subsegments, the first subsegment will be tested if the segment has larger than 40 frames, otherwise, it will be considered the same speaker as the segment before. The second subsegment will not be tested, instead, the next 2 second segment will start from the segmentation point.

5.2 Real time test on 168testset

The model used here is UFCC with 19 cepstral coefficients modeled with 4 mixtures.
A long speech is generated by combining testing speeches from random speakers. The testing using 50 random speakers, and an average of 6 errors occurs each time. Most of these errors appears in the changing segment from one speaker to another where a short subsegment is not correctly identified.

5.3 Test on real TV drama

This test is performed on soap pop drama ‘Friends’. The audio file is 48kbps mp3 coded with 48kHz sampling rate. The training speeches are cut from season six episode one to five, with 1 minute for each of the six main characters. Since the drama contains a lot background laughing, a laughing model is also trained with 20 seconds of training data. The filter bank used is a uniform filter bank with frequency range from 0 to 8000, frame length is 30ms, with 50% overlap, 19 cepstral coefficients and 10 delta cepstral are used, model is 32 mixtures. The speeches have much more variation than the neutral speech in corpus, hence more features and mixtures are used here to have a better recognition.
I haven’t come across a testing metric so far, most tests are based on perception. From my observation, 70 to 80 percent of time, the decision is correct, even with some background noise such as foot steps, coffee house noise, etc. Most of the errors are still caused by the variation of a speaker’s voice, such as speaking emotionally.

6. Conclusion and Future work

In this project, speaker identification system using GMM with small number of mixtures for filter bank based cepstral coefficients are examined. Three different filter

bank, namely, Mel-scale, uniform, and non-uniform filter banks are used and compared. For TIMIT corpus, while they all give a pretty good correct rate, UFCC gives the best speaker identification performance, NFCC is the second and MFCC has the highest error rate. This proves that some important speaker dependant information are located at high frequency range. The design of non-uniform filter bank still has space to be improved. The test on compressed speeches shows that MFCC has the best robustness against compress distortion. Training and testing on the same source will give a good performance and is suggested, however, this could not always be meet, hence longer training data with difference source may give a better robustness, presumably. Real time segmentation and identification are developed for conversations. The preliminary test on TIMIT speeches shows a good result and test on TV show 'friends' gives a promising perceptual experience, although it is not totally satisfactory since there are noise and emotion issues involved here.

There are several future directions. One is to deal with the speech with noise and background music. Second one is to come up with a better identification scheme which performs faster such as proposed in [6] and with out search for probability from all the models so that larger number of mixtures can be used for harder identification task. Third one is to deal with the normalization for different emotions since this is one most common reason of errors in real time test.

References:

[1] Campbell, J.P., Jr. 'Speaker recognition a tutorial' Proceedings of the IEEE, vol. 85, no. 9, pp. 1437 – 1462, 1997

[2] Reynolds, D. and Rose, R. 'Robust text-independent speaker identification using Gaussian mixture speaker models', IEEE Trans. Speech Audio Processing, vol.3, no.1, pp. 72-83, 1995

[3] Lu, X. and Dang, J. 'An investigation of dependencies between frequency components and speaker characteristics for text-independent speaker identification', Speech Communication, vol. 50, no. 4, pp. 312-322, 2008-05-08

[4] Doddington, G., Przybocki, M, et al. 'The NIST speaker recognition evaluation - overview methodology, systems, results, perspective', Speech Communication, vol. 31, no. 2-3, pp. 225-254, 2000

[5] Duda, R., Hart, P. and Stork, D. 'Pattern Classification (2nd edition)', Wiley, 2001

[6] Kinnunen, T., Karpov, E. and Franti, P. 'Real-Time speaker identification and verification', IEEE Trans. Audio, Speech and Language Processing, vol. 14, no. 1, pp.277-288, 2006.

[7] Reynolds, D., Quatieri, T. et al. 'Speaker verification using adapted Guassian mixture models', Digital Signal Processing, vol. 10, no.1, pp. 19-41, 2000

[8] Chakroborty, S. Roy, A. et al. 'Improved closed set text-independent speaker identification by combining MFCC with evidence form flipped filter banks', International Journal of Signal Processing, vol. 4, no. 2, pp.114-121, 2008

[9] Cordella, P., Foggia, P. et al. 'A real-time text-independent speaker identification

system', IEEE Proc. 12th international conference on image analysis and processing, no. 17-19, pp. 632-637, 2003

[10] Ellis, D. and Lee, 'analysis of everyday sounds', 'http://www.ee.columbia.edu/~dpwe/e6820/lectures/L12-archives.pdf'