Parametric Speech Emotion Recognition Using Neural Network

Rui Ma

January 2014

Examiner: Magnus Isaksson
Supervisor: Javier Ferrer Coll
Abstract

The aim of this thesis work is to investigate the algorithm of speech emotion recognition using MATLAB. Firstly, five most commonly used features are selected and extracted from speech signal. After this, statistical values such as mean, variance will be derived from the features. These data along with their related emotion target will be fed to MATLAB neural network tool to train and test to make up the classifier. The overall system provides a reliable performance, classifying correctly more than 82% speech samples after properly training.
Acknowledgement

The author wants to thank Javier Ferrer Coll not only for his effective suggestions and help, but also his consistently encouragement to make this thesis possible.
# Table of contents

ABSTRACT .......................................................................................................................... 1

ACKNOWLEDGEMENT ....................................................................................................... 2

TABLE OF CONTENTS ........................................................................................................ 3

1 INTRODUCTION ............................................................................................................... 4
  1.1 BACKGROUND .......................................................................................................... 4
  1.2 OBJECTIVE OF THESIS .......................................................................................... 5

2 THEORY ............................................................................................................................ 7
  2.1 PREPROCESS ............................................................................................................ 7
  2.2 FEATURES ................................................................................................................ 9
    2.2.1 Speech rate ......................................................................................................... 9
    2.2.2 Energy .............................................................................................................. 10
    2.2.3 Pitch ................................................................................................................. 10
    2.2.4 Formant ............................................................................................................ 11
    2.2.5 Mel-Frequency Cepstrum ................................................................................ 12
  2.3 NEURAL NETWORK SYSTEM .................................................................................... 13

3 METHOD AND SIMULATION RESULTS ....................................................................... 15
  3.1 SPEECH CORPS RECORDING ................................................................................. 15
  3.2 FEATURE EXTRACTION ........................................................................................... 15
    3.2.1 Speech Rate ...................................................................................................... 15
    3.2.2 Energy ............................................................................................................. 16
    3.2.3 Pitch ............................................................................................................... 16
    3.2.4 Formant ......................................................................................................... 17
    3.2.5 Mel-Frequency Cepstrum ................................................................................ 18
  3.3 CLASSIFICATION WITH NEURAL NETWORK ......................................................... 18

4 DISCUSSION AND CONCLUSION ............................................................................... 24

REFERENCES ..................................................................................................................... 26

APPENDIX .......................................................................................................................... 27
1 Introduction

1.1 Background

Human-computer interaction has become the most popular subject nowadays, lots of attentions have been put on in this area and it still has great potential. One important aspect of human-computer interaction is to teach computer to understand human’s emotion through voice, so that it can give different response. There have been a lot of success applications about voice recognition in the market so far. People can use their voice to give command to car, cellphone, computer TV and many electrical devices. Hence make the device understand human emotion and give a better experience of interaction becomes a very interesting challenge. Many researches have been done for this purpose, for instance in [1] Fadi. A. Macht et.al investigate an application of speech emotion recognition to avoid traffic accident. The work performed in [2] utilizes a recognition machine to classify the voice message in phone answering machine and gives priority.

Typically, the most common way to recognize speech emotion is to first extract important features that are related to different emotion states from the voice signal (i.e. energy is a important feature to distinguish happy and sad), then feed those features to the input end of a classifier and obtain different emotions at the output end. This process is shown in Figure. 1.
There have been many studies about this topic in recent years, different features as well as different classification methods are used, i.e. [3] Nogueiras et al. use hidden markov model to recognize emotions from pitch and energy. [4] B. Schuller et al. use four different classification methods to compare their performance using a combination of acoustic features and linguistic information. Further more those studies use their own speech corps so that they give various results and conclusion.

1.2 Objective of Thesis

In this thesis, the aim is to classify a batch of recorded speech signal in four categories using MATLAB, namely: happy, sad, angry, nature. Before extract the features, these speech signals need to be preprocess. First take samples from the speech to convert analog signal to digital signal. Then the normalization makes sure each sentence is in the same volume range. At last, segmentation is to separate signal in frames so that speech signal can maintain its characteristic in short duration. Five commonly used features are chosen to study and extract use MATLAB. Speech rate and energy are the most basic features of speech signal but they still have significant different between emotions such as angry and sad. Pitch is frequently
used in this topic and autocorrelation method is used to detect the pitch in each frame. After that statistical value such as mean, variance, max value of pitch will be calculated for speech signals. Formant is another important feature in this paper. Linear predictive coding (LPC) method is used to extract the first formant. Similar to pitch, statistical values are calculated for first formant. Mel frequency cepstral coefficient (MFCC) is a representation of short-term power spectrum on a human like mel scale of frequency. First three coefficients of MFCCs are taken to derive means and variances. All 15 features of 60 sentences are put into an input matrix along with a target matrix, which indicate the emotion state for each sentence composed the input of neural network. MATLAB neural network pattern recognition tool is used to train and test the data and perform the classification, in the end figures of mean square error and confusion will be given to show how good the performance is.
2 Theory

This chapter contains the theoretical concepts used for classifying a speech signal. Next section, describes the process of recording the data, sampling, normalization and segmentation. The second section contains the extracted speech features clarifying the process of the extraction. Finally, the classification of the speech samples using neural networks is provided in the third section.

2.1 Preprocess

Before extract the features, there are some necessary steps to take to manipulate speech signal. Preprocess mainly includes sampling, normalization and segmentation as shown in Figure 2.

![Figure 2. Preprocess of speech signal.](image)

Speech voice is analog signal and it needs to be converted into digital signal to process in computer. Sampling theory provides a way to transform the analog signal $x(t)$ into a discrete time signal $x(n)$ and remains the characteristic of original signal. According to sampling theorem [5], when the sampling frequency is larger or equal to 2 times of the maximum of analog signal frequency, the discrete time signal is able to reconstruct the original analog signal. As indicated in Figure 3, sampling is performed by collecting points from analog signal in a certain rate $T_s$. Generally the sampling frequencies for speech signals are 8000Hz, 16000Hz and 44100Hz. In MATLAB sampling is applied automatically after the recording function.
This thesis use microphone to record all speech signals and in reality it is impossible to control the recording volume for each sentence to remain in the same level. Volume is an important fact when calculating speech energy and other features. Normalization process use the signal sequence divided by maximum value of the signal to make each sentence has a comparable volume level. The formula of normalization is shown below in Eq. (1).

\[ \hat{s}(n) = \frac{s(n)}{s_{\text{max}}}, n = 1, 2, ..., N \]  

(1)

s(n) is the original sampled signal, \( \hat{s}(n) \) is the signal after normalization, \( s_{\text{max}} \) is the absolute maximum value of the signal sequence, \( N \) is the length of the sequence.

Speech is a random signal and its characteristic is changing with time, but this change is not instant. Generally it assumes in a short duration (10-30ms) the signal is stable. In this sense, segmentation process divides the signal sequence into many frames with overlap as shown in Figure. 4. Overlapping is used to avoid loss of data due to aliasing [5]. The signal \( s(n) \) becomes \( s_i(n) \) once framed, where \( i \) indicates the number of frames. After preprocess characteristics of whole speech signal could be study from statistical values.
Figure 4. Example of segmentation process.

2.2 Features

2.2.1 Speech rate

Speech rate is a representation of how fast people speak. It has strong connection with emotions like happy, angry and sad. This feature is extracted from Eq. (2)

\[ S_r = \frac{t_v}{n_w} \]  \hspace{1cm} (2)

Which \( S_r \) is speech rate, \( t_v \) represent total duration of voice part and \( n_w \) is number of words used in a sentence. By studying waveform of recorded speech signal in time domain (after normalization) like shown in Figure 5, the duration of the voice part could be calculated using MATLAB function to find the points that are bigger than zero and then divide by the sampling frequency. To set a higher threshold level and define a good range between each word, the number of words could be found using MATLAB function as well. As an example shown in Fig. 5, there are 5 words in this sentence.
2.2.2 Energy

Energy is a basic feature in speech signal process and it plays an important role in emotion recognition, e.g. speech signals of happy and angry emotion have much higher energy than sad. Because in this study the record function used in MATLAB has a threshold level for minimum voice, which highly reduced the effect of noise that energy of speech signal could simply be calculated using Eq. (3) [5] after normalization.

\[ E_n = \sum_{n=1}^{N} x(n)^2 \]  

Where \( E_n \) stands for energy and \( x(n) \) is the signal sequence after sampling, \( N \) is sequence length. Each sampled point will be multiplied by itself and added up for the overall signal.

2.2.3 Pitch

Pitch known as the perceived rising and falling of voice tone, is the perceptual form of fundamental frequency. It represents the vibration frequency of vocal folds during speaking. Huang [7] states, “It’s called fundamental frequency because it sets the periodic baseline for all higher-frequency harmonics contributed by the pharyngeal and oral resonance cavities above. It is the source of speech model.” Besides, it is the most frequently used feature in speech emotion recognition.

There are many ways to estimate pitch from a speech signal. In this paper, autocorrelation
method is used for it is a commonly used method and is easy to practice. The method use short-term analysis technique to maintain characteristic for each frame, which means pre-process should be fully applied before extract pitch. Since autocorrelation can decide the period of a periodic signal, for each frame apply the autocorrelation use Eq. (4)

\[ R(k) = \sum_{m=1}^{FL} x(m)x(m + k) \]  

(4)

Where FL is frame length, x(m) is the signal frame and k is shift parameter and R(k) represents result of the autocorrelation.

![Autocorrelation result of a frame](image)

Figure. 6. Autocorrelation result of a frame

In Fig. 6, an example of autocorrelation result of one frame is shown. For a periodic signal it is convolved to itself with a shift parameter, the correlation function reaches maximum at shift position of 0, +p, +2p, etc. Where p is the period of signal. In other words, find the position of first peak other than the one at zero point and then take reciprocal of it and multiply with sampling frequency will give the pitch of one frame.

### 2.2.4 Formant

Formant frequencies are defined as resonances in vocal tract and they determine characteristic timbre of vowel [7]. It is also a very useful feature for speech recognition and could be found in many speech emotion studies. A graphical representation of formants is shown in Fig.7. The peaks of the frequency response from a linear prediction filter are the formants. This illustration also provides a way to obtain the formants. That is to compute the roots of a linear prediction coding (LPC) polynomial [8]. This should be done at frame level as well. The linear prediction coding as its name indicate that it predicts current sample as a linear
combination of its past samples \[7\].

\[
\hat{x}[n] = \sum_{k=1}^{p} a_k x[n - k]
\]

(5)

Where \( \hat{x}[n] \) is the predict sample, \( p \) is the number of past samples and \( a_k \) is the coefficient factor. The prediction error goes as

\[
e[n] = x[n] - \hat{x}[n] = x[n] - \sum_{k=1}^{p} a_k x[n - k]
\]

(6)

\[
M(f) = 1125 \ln(1 + f/700)
\]

(7)

The process of extract MFCC from a speech signal is shown in Fig. 8. After preprocess the periodogram of power spectrum is estimated for each frame by take the complex discrete Fourier transform (DFT) and square it. The complex DFT of is calculated use Eq. (8).

\[
S_l(k) = \sum_{n=1}^{N} S_l(n) h(n) e^{\frac{j2\pi kn}{N}} \quad 1 \leq k \leq K
\]

(8)
Where $h(n)$ is an $N$ sample window, and $K$ is the length of the DFT.

Then the periodogram is given by Eq. (9).

$$P_t(k) = \frac{1}{N} |S_t(k)|^2$$  \hspace{1cm} (9)

Then apply mel filter bank to the power spectrum from last step using Eq. (7). Take the logarithm and then discrete cosine transform (DCT) of all filter bank energies. Generally only the first 13 coefficients are used at last.

Figure. 8. Process of extract MFCC

### 2.3 Neural Network system

The inventor defines neural network as ‘a computing system made up of a number of simple but highly interconnected processing elements, which process information by their dynamic state response to external inputs.’ [9] As its name, neural network algorithm imitates biological system but with a much smaller scale. Those highly interconnected processing elements are called neurons and belong to different layer as shown in Fig. 9.
Neural network could have one or more hidden layers depend on the complexity of system. Patterns are sent through input layer while targets are sent to output layer. With the communication between hidden layers the system trains the data by modifying the weights of each pattern so that gives the most desire performance.

*Figure. 9. Simple demonstration of neural network*
3 Method and Simulation Results

This chapter contains practical process of classifying speech signal using MATLAB. MATLAB codes and comments are attached in Appendix. Besides, some functions from MATLAB voice toolbox are used to process the signals. Next section describes the recording state. The second section provides the working function of commands of feature extraction. Last section contains the final result using neural network as classifier.

3.1 Speech corps recording

The author recorded 15 different sentences for four emotions (angry, happy, sad and nature). MATLAB command “audiorecorder” is used with default sampling frequency of 8000 Hz and the duration is set to 7 seconds. So each sentence or signal now becomes a sequence with 56000 elements. The codes are attached in Appendix A.1.

3.2 Feature Extraction

Five MATLAB functions were created for five main features. A main function (in A.6) will collect all the features for given speech signals and put them into input form of classifier. This separate coding structure is used to collect as well as append big set of data into the system.

3.2.1 Speech Rate

Two thresholds were defined thr1=0.03 and thr2=0.3. The lower one sets for voice duration and the other for number of words. ‘Find’ command is used to give the position information of desire elements. Then voice part could be easily derived use the number of desire elements (e.g. those have higher value than the first threshold thr1) divided by the sampling frequency. For words number, first a reasonable range of gap between each word is studied and defined. Then a loop is made to add words whenever a desire element (one higher than the second threshold thr2) reaches the defined range. Table.1 shows the average speech rate for each
emotion.

Table 1. Average speech rate for four emotions

<table>
<thead>
<tr>
<th>Emotions</th>
<th>Angry</th>
<th>Happy</th>
<th>Sad</th>
<th>Nature</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average Speech Rate</td>
<td>0.35</td>
<td>0.27</td>
<td>0.17</td>
<td>0.16</td>
</tr>
</tbody>
</table>

The number shows that people speaks faster when they are angry and quite slow when they are sad. This result appears the same with the real world situation.

3.2.2 Energy

As mentioned in chapter 2, extract energy of a speech signal can simply apply the Eq. (3) to the signal sequence after normalization. Use ‘sum’ command in MATLAB, the result of average energy for each emotion is shown in Table. 2.

Table 2. Average energy for four emotions

<table>
<thead>
<tr>
<th>Emotions</th>
<th>Angry</th>
<th>Happy</th>
<th>Sad</th>
<th>Nature</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average Energy</td>
<td>1351</td>
<td>767</td>
<td>30</td>
<td>124</td>
</tr>
</tbody>
</table>

Similar to speech rate, angry mood has the highest energy happy is the second and sad has the least energy. It is also a reasonable outcome. MATLAB code for energy is placed within the main feature extraction function in A.6.

3.2.3 Pitch

First, the frame length and overlap for speech voice are defined. Then use MATLAB voice box function ‘enframe’ to implement the segmentation. With this setting the ‘enframe’ function returns a 279x240 matrix for one speech signal, which stores 279 frames with each frame has 240 elements. A loop is made in MATLAB to take each frame one at a time to implement auto-correlation using ‘xcorr’ command according to Eq. (4). Then to avoid the confusion caused by peak values that appear around zero point, amount of points are taken
out temporarily. At last ‘find’ command was used to give the position of first peak, now pitch is obtained by divide the sampling frequency by first peak position. Besides, for those unvoiced parts positions of first peak have big difference with the voice part. A condition function will carry out in MATLAB to set their pitches to zero. Fig. 10 shows the result of pitches extract from a speech signal.

![Pitches of an angry speech](image)

*Figure. 10. Pitches of an angry speech*

The pitches vary with time but are all in the reasonable range, mean value and variance will be studied and calculated for each speech signal.

### 3.2.4 Formant

The same preprocess is taken as in previous section, after that use “filter” command to apply a highpass all-poke filter. “lpc” command used to obtain linear prediction coefficients. Only take the first 3 formant frequencies, so set the lpc model to 8 (two times the expected number of formants plus 2). Find the roots returned by lpc using “roots” command. The roots would be in complex conjugate pairs and only keep the roots with one sign for imaginary part then determine the angles to the roots. To calculate the bandwidths of the formants first convert angular frequencies to hertz then set a suitable range for bandwidth. An example of the first formant for each frame is shown in Fig. 11.
As shown, most formants are in range of 350 – 1000 Hz, which is logical. Those have value of 4000 are due to the edge effect, which is caused by the confusion of the predictor for frames contain both voice and unvoiced part. The mean and variance will also be calculated for formant.

![Formant of a happy speech](image)

Figure. 11. Formant of a happy speech

### 3.2.5 Mel-Frequency Cepstrum

For short time process, preprocess is again full applied for speech signals. Another MATLAB voice toolbox function “melbankm” is used for mel scale filter bank wrapping in Eq. (7). For a chosen 256 points FFT, 32 filters are used in filterbank. Then Fast Fourier transformation is carried out use “fft” command and “log” for the logarithm. After the discrete cosine transform carries out, wipe out the first and the last frame. The Mel-Frequency cepstrum coefficients are returned in a 693 x 32 matrix and in this paper only the first three were selected.

### 3.3 Classification with Neural Network

Enter “nnstart” to start the MATLAB GUI of neural network. It has several functions and choose pattern recognition tool then the general working flow is shown in Fig. 12. Press the next bottom to load input data and target data. Input data here is a 15 x 60 matrix of 15 features of 60 sentences. Target data is a 4 x 60 matrix, which indicates the emotion state for
these 60 sentences i.e. 1000 for angry, 0100 for happy, 0010 for sad and finally 0001 for nature mood. After filling those data, next step is to randomly choose the percentage of input data into 3 categories namely training, validation and testing. Where training set is used to fit the parameters of the classifier i.e. to find the optimal weights for each feature. Validation set is used to tune the parameters of a classifier that is to determine a stop point for training set. Finally test set is used to test the final model and estimate error rate. The default value sets training in 70 percent and 15 percent each for the rest. Use default value and next step is to choose the number of hidden layers. As introduced in previous chapter more hidden layers for more complicate system and gives better result. 10 is the number of default hidden layer. Finally, last step is to press the train bottom to let the system train several times and after that mean square together with error rate will indicate how good the results are. The result plots are shown below in Fig. 13-16.

In Fig. 13, it shows classifier reaches the best performance at epoch 18, where epoch means number of times for all the training vectors used once to update the weights to the features. Table 3, gives a description of classification result in Figure 14, 16 and 17. As shown, these four emotions are listed together with the error rate for each row and column represent for target class and output class. Therefor number in cell ‘1’ stands for how many angry speeches have been classified into the angry output. Cell ‘2’ shows how many happy speeches have been misclassified into angry class et.al. Classification results of all three sets of data and an
overall result are given. This gives a clear idea of how good the classification system is. For example in Figure 14 overall matrix, all of angry speeches have been put in the correct output. However 9 happy speeches are misclassified into the angry output and 6 of them are in the nature output. For sad speeches, there is only one misclassified in nature output. At last, 3 nature speeches are put into sad output and the rest are correct. The overall error rate is 31.7%, the system needs to be improved.

![Figure 13. Mean square of error after two times trained](image)

**Table. 3. Description of classification result.**

<table>
<thead>
<tr>
<th></th>
<th>Angry</th>
<th>Happy</th>
<th>Sad</th>
<th>Nature</th>
<th>Error Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Angry</td>
<td>1</td>
<td></td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Happy</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sad</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Nature</td>
<td></td>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Error Rate</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Figure 14. Classification result of two times of training

The performance can be improved by clicking training button more times to let system reach an optimal setting. Fig. 15 shows after few more times training the mean square error increases to 0.071 from 0.118 and the confusion matrix shows a lower error rate in Fig. 16. This time correct rate reaches 95%.
Figure. 15. Mean square after 6 times training

Figure. 16. Confusion result after 6 times training
Further more, after a suitable train process i.e. with a low error rate performance the neural network is ready to classify extra speech signals. E.g. with six times training in previous section, the system is now assigned to classify a completely new speech corps. This time there are 10 sentences for each emotion. The classification result in terms of error rate is shown in Fig. 17. As shown, 9 angry speeches are put into the correct output; 8 happy speeches are recognized as happy emotion; 7 sad speeches are correctly classified; 9 nature speeches are in the nature output and the total classification rate is 82.5% for the new speech samples.

![Confusion Matrix](image)

*Figure. 17. Classification result of additional samples*
4 Discussion and Conclusion

In this section, a discussion about the methodology used to extract features and classification results will be presented. In the end, according to the classifying result and discussions a conclusion will be given to this thesis work.

4.1 Discussion

First, for the recording process, author recorded all the speech corps. Besides, each emotion is performed in a similar acting type. E.g. Sad speeches are all sound slow, depress and powerless. Those make it comparably easier for the classifier to decide the category of each speech. The further work could be for a more accurate system to classify speeches from different performer with different acting skill.

In the feature extraction process, the accuracy of features is an important factor. For speech rate, the number of words in a sentence could be examined to compare the simulate result and counting of the words. The method used in this paper defined a reasonable range between each detected word sample. It doesn’t work well for polysyllabic words or fast speakers. E.g. “methodology” might be counted for two words. However, this method provides a quick and practical way to extract speech rate. For pitch and formant, it is not likely to examine the accuracy for each frame, so the reasonable ranges for pitches and formants are defined to filter other error values out. This is a quick and simple implementation as well, but it might also cause the edge effect as shown in Fig.11. One way to improve this is by applying a start and end point detection algorithm to indicate the start and end sample point of real speech part.

For the classification process, the error rate is highly depended on the training times. Compare the confusion results in Fig. 16 and Fig. 14 the overall error rate varies from 5% to 31.7%. However, the suitable number of training times is neither fixed nor predictable in
neural network system. For the random process of choosing training, validation and test data, the weights assign for each feature changes for different choice. So to able to have a desirable result, it is necessary to test the training process several times.

4.2 Conclusion

MATLAB neural network is a powerful tool for pattern recognition and classification. With its simple user interface, the chosen features of speech signals could be easily load into the system and training for the target emotions. After suitable times of training process, extra test signals are load into the system for emotion recognition. With a desired result of 85% classification rate those selected features (speech rate, energy, pitch, formant and MFCC) are proven to be good representations of emotion for speech signal. For the further work the system could be improved by increase the accuracy of extracted features to classify more complicate speech samples i.e. for multiple speakers and more emotions.
References


Appendix

The program codes are attached below

A.1
%% Voice record of speech mood analysis

clear
fs= 8000; %sampling frequency
t=7;%the record duration
time= linspace(0,t,t*fs); %vector of time

recorder = audiorecorder; % creates an 8000 Hz, 8-bit, 1-channel
audiorecorder object
disp('Start speaking.'); %notice when the recorder
starts record
blocking(recorder, t); %stop recording with the setting time
disp('End of Recording. '); %notice the end of the recording

play(recorder); % Play back the recording
y = getaudiodata(recorder); % Store data in y
plot(time, y, 'LineWidth',0.5); % Plot the waveform

x = find(abs(nSig)>thr1); % low threshold to find the voice part

v = find(abs(nSig)>thr1); % low threshold to find the voice part
voicePart = length(v)/fs;

w = find(abs(nSig)>thr2); % high threshold to find the words
n = length(w);
for i = 1:n-1
    if w(i+1) - w(i) > 800 % define the gap between words

A.2
%% Speech rate detector
function [nw sr]=speechRate(Sig);
%[Sig,fs]=wavread('test5'); % get signal and sampling frequency
fs=8000;
nSig = Sig / max(abs(Sig)); % normalization
thr1=0.03; % set the threshold to decide the voice part
thr2=0.3; % threshold to decide how many words
word=1; % variable to calculate words

v = find(abs(nSig)>thr1); % low threshold to find the voice part

w = find(abs(nSig)>thr2); % high threshold to find the words
n = length(w);
for i = 1:n-1
    if w(i+1) - w(i) > 800 % define the gap between words
word=word+1;%count words
end
end
nw=word;
sr=voicePart/nw;
%plot(abs(nSig))

A.3
%% pitch detect use autocrosscorrelation
function mfps=pitchs(Sig)
%[sig,fs]=wavread('test4.wav');
fs=8000;%
FrameLen = 240;
FrameInc = 200;
Sig=enframe(Sig,hamming(FrameLen,'periodic'),FrameInc);

for i=1:279 %279 frames for 7 second speech signal
    x=fSig(i,:);%calculate autocorrelation for each frame
    rxx=xcorr(x);
    y=rxx(255:end);%define distance to first peak
    [k,I]=max(y);%find the first peak, value and position
    if I<30 % define the unvoice part
        fps(i)=0;
    elseif I>160
        fps(i)=0;
    else
        fps(i)=fs/(I+14);%add the cut points back to
        %give the right position to
        %calculate the pitches
    end
end

%stem(fps)

ind=find(fps);%find those nonzero values in fps for calculation
mfps=fps(ind);

A.4
%% formants use LPC
function mFormant=formant(Sig);
% [Sig, fs] = wavread('test4.wav');
fs = 8000;%
FrameLen = 240;
FrameInc = 200;
fSig = enframe(Sig, hamming(FrameLen, 'periodic'), FrameInc);

for i = 1:279  % 279 frames
    x = fSig(i, :);
    if 1 && ~any(x)
        Formant(i) = 0; % for empty frame returns zero
    else
        pe = [1 0.63]; % preemphasis
        x = filter(1, pe, x);
        A = lpc(x, 8); % lpc coefficient
        rs = roots(A);
        rs = rts(imag(rs) >= 0);
        ang = atan2(imag(rs), real(rs));
        [frq, indices] = sort(ang.*((fs/(2*pi)));
        bw = -1/2*(fs/(2*pi)) * log(abs(rs(indices)));
        nn = 1;
        for k = 1:length(frq)
            if (frq(k) > 90 && bw(k) < 400)
                formants(n) = frqs(k);
                n = n + 1;
            end
        end
        Formant(i) = formants(:, 1); % dimension mismatch, here only take the first formant
    end
end

ind = find(Formant); % find those nonzero values in fps for calculation
mFormant = Formant(ind);
end
end

A.5

%%% MFCC
function cc = mfcc(Sig);

end
%[Sig,fs]=wavread('test1.wav');
bank=melbankm(32,256,8000,0,0.5,'m'); %mel space filterbank [10]
ban=full(bank);
bank=bank/max(bank(:));
for k=1:16 %DCT coefficient
n=0:31;
coef(k,:)=cos((2*n+1)*k*pi/(2*32));
end
w=1+8*sin(pi*[1:16]./16);
w=w/max(w);
x=double(Sig);
x=filter([1-0.9375],1,x);
x=enframe(xx,256,80); %segmentation
for i=1:size(x,1) %Calculate MFCC for each frame
y=x(i,:);
s=y'.*hamming(256);
t=abs(fft(s));
t=t.^2;
c1=coef*log(bank*t(1:129));
c2=c1.*w';
m(i,:)=c2';
end
dtm=zeros(size(m)); %differential coefficient
for i=3:size(m,1)-2
dtm(i,:)=-2*m(i-2,:) - m(i-1,:) + m(i+1,:) + m(i+2,:);
end
dtm=dtm/3;
c=[m dtm];
c=c(3:size(m,1)-2,:);
c(isnan(c))=0;

A.6
%% main function feature
function f=features(Sig)
%fs=8000;%sampling rate
[nw sr]=speechRate(Sig); %Speech rate

nSig = Sig / max(abs(Sig)); %normalization
eg = sum(nSig.^2); % energy

mfps=pitches(Sig); % pitch
mep=mean(mfps);
vp=var(mfps);
map=max(mfps);
mip=min(mfps);

mFormant=formant(Sig); % formant
meF=mean(mFormant);
vF=var(mFormant);
maF=max(mFormant);

cc=mfcc(Sig); % MFccs
mf1=cc(:,1);
mf2=cc(:,2);
mf3=cc(:,3);

mf1m=mean(mf1);
mf2m=mean(mf2);
mf3m=mean(mf3);

mf1v=var(mf1);
mf2v=var(mf2);
mf3v=var(mf3);

f=[sr eg mep vp map mip meF vF maF mf1m mf1v mf2m mf2v mf3m mf3v].';

end