DSPLAB Project (1) Implementation of Pathological Voice Detection Using KNN

1. Goal

Use speech analysis technology to detect pathological features in the voice, examine whether acoustic analysis technology can be applied to patients with vocal discomfort, and correctly predict cases with true voice diseases. We will studies the K-Nearest Neighbor (KNN) algorithm , use the Mel-scale Frequency Cepstral Coefficients (MFCC) tool, and conduct simulations and discuss the results of speech detection on Matlab.

1. Experimental tools (Matlab、Audio file)

mfcc.m Input the audio file into this program to generate MFC file.

mfcc\_v2.m Enter the audio file and other parameters.

mfcc\_all\_v1.m Program that handles mfcc\_v2.m.

label\_person.m Give a label of 0 or 1.

folder audio files In this folder there are im (sick folder) and nor (normal folder).

folder im There are two kinds of throat-related lesions in the folder, with a total

of 20 sick mixed male and female sound files.

folder nor There are 20 normal mixed male and female voice files in this

folder.

folder other There are 10 mixed male and female voice files in this folder.

1. Data description

**This topic is related to vocal cord diseases, so we will briefly introduce eight types of vocal cord-related lesions.**

1. Cancer–Swallowing discomfort or pain, irritating cough, blood in sputum, difficulty breathing and neck mass in severe cases.
2. CYST–Hoarse voice, difficulty speaking, bifurcated voice, and inability to sing high notes.
3. Atrophy–Vocal cords can easily become loose, causing fatigue when speaking and hoarseness of the voice, which can lead to inability to work and sing.
4. Nodules–Hoarseness or other voice changes, a lump in your neck, sore throat or feeling like something is stuck in your throat, or persistent coughing.
5. Polyp–Benign proliferative lesions that occur in the superficial layer of the vocal cords are also a special type of chronic laryngitis.
6. SD & Tremor - A clinical disease that causes abnormal voice due to excessive tension in the throat muscles.
7. Sulcus–The patient's vocal cords appear like wrinkles on the skin, causing hoarseness.
8. Vocal palsy–When the motor nerves of the throat are damaged, three types of paralysis occur: vocal cord abduction, adduction, or muscle tone relaxation.

This experiment provides 40 males and females mixed voice wav files, including 20 normal wav files and 20 sick wav files with two types of throat-related lesions: vocal cord cancer and CYST. The voice content is single. Sound "Ah", and the voice time range: 4 to 30 seconds.

1. Flowchart Concept Illustration (the theory is on p.9~p.11)

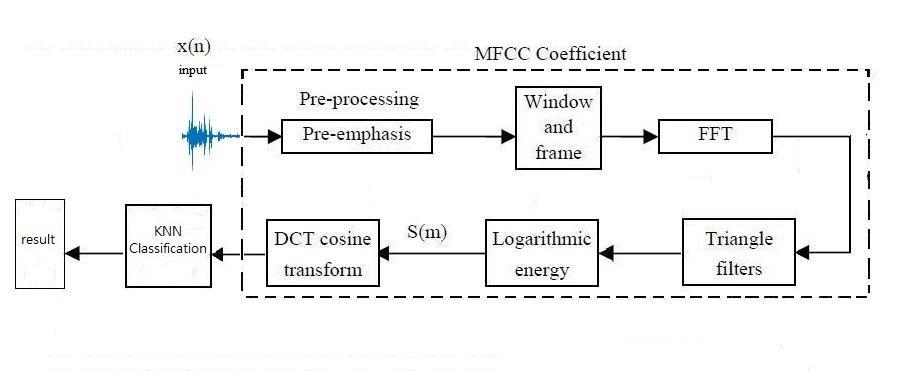


Figure 1、Flow Chart. First, send the input speech (wav file) to MFCC to select features,

and then observe the KNN classification performance.

1. Experiment process

Experiment 1、Generate MFCC (Mel-scale Frequency Cepstral Coefficients)

In this experiment, we use the tool mfcc.m program provided in the attachment to generate mfcc. Firstly, we need to create a Input files, and put im files and nor files into the Input file. Secend, we create a empty Output file for storing the mfc files from transforming wav files.

This code already contains the mfcc\_v2 tool. First, enter the various parameters required (capture sound frame size, sampling frequency, dimension size, etc.). The parameters for this topic are adjusted to sample\_rate = 44100, frame\_time = 30 and frame\_move\_time = 15.

indir2 = ‘.\ Place the wav file and enter the folder path’; % Enter the voice file path

outdir2 = ‘.\ Place the mfc file output folder path’; % Output mfc file path

out\_ext2 = ‘.mfc’; % Generate mfc file

mfcc\_all\_v1(indir2,outdir2,out\_ext2); % Use mfcc\_all\_v1.m file to execute

fprintf(‘NORMAL complete\n’); % Display the mfc file in the command window

as completed

Hint：mfcc\_v2.m parameter frame\_time = frame\_move\_time \* 2.

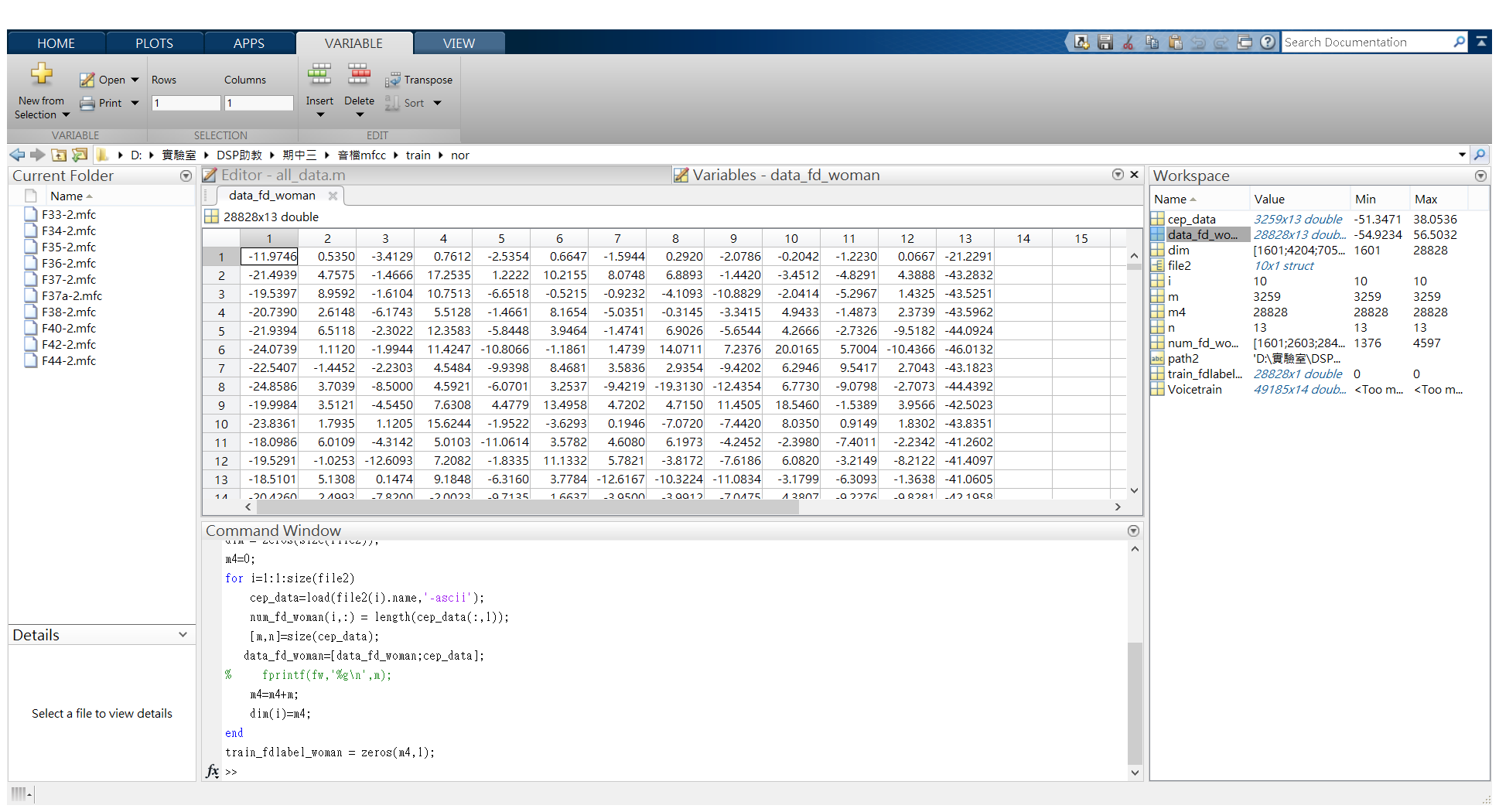


Figure 2、This picture is the result of referring to the mfc file. From left to right, there are C0, C1,

C2,..., C25, a total of 26 dimensional features.

**Experiment 1 Questions and Discussion:**

**Question 1: Try to draw the spectrogram and waveform diagram (refer to LAB6) of sick and normal sounds (choose one each), and explain the differences? (20%)**

**Question 2: Try to adjust the parameters sample\_rate, frame\_time, and frame\_move\_time in mfcc\_v2.m to explain the numerical changes and significance of the mfc file for sound capture features. (20%)**

**Parameter Description:**

Sample\_rate: The sampling frequency of a piece of audio (unit: HZ)

Frame\_time: The sampling time length of a sound frame(unit: ms)

Frame\_move\_time: The time when one sound frame overlaps with the next sound

Frame(unit: ms)

Experiment 2、Conbine sick and normal data in training and testing and

give labels(Label)

Select the mfc files in order of name, and select the 1st to 10th normal data as training data, and the 11th to 20th as testing data. Select the 1st to 5th data for each of the two sick data as training data, and the 6th to 10th data as testing data. The distribution of the training and testing data as shown in Figure 3:



Figure 3、 Data distribution situation

If the data is directly classified without conbining and giving labels, more models will be generated, which will take more time in terms of efficiency. Moreover, the model only has one label, it will affect the classification of another label. In order to avoid the above error occurs, the sick and normal MFCCs in the training data are conbined together. The order of conbining together is the normal sound followed by the sick sound. The same steps are also performed for the test data. At the same time, the sick MFCC is given the label 1 (matlab function : ones), and normal MFCC is given label 0 (matlab function: zeros), as shown in Figure 4, which stores the length of each mfc file of normal and sick in the stringed test data, training data and test data.

**Reference program：**

path = 'Select path '; % Enter the mfc file path folder

cd(path) % Go to selected folder

file=dir([path '\\*.mfc']);

Data= [];

for i = 1:length(file)

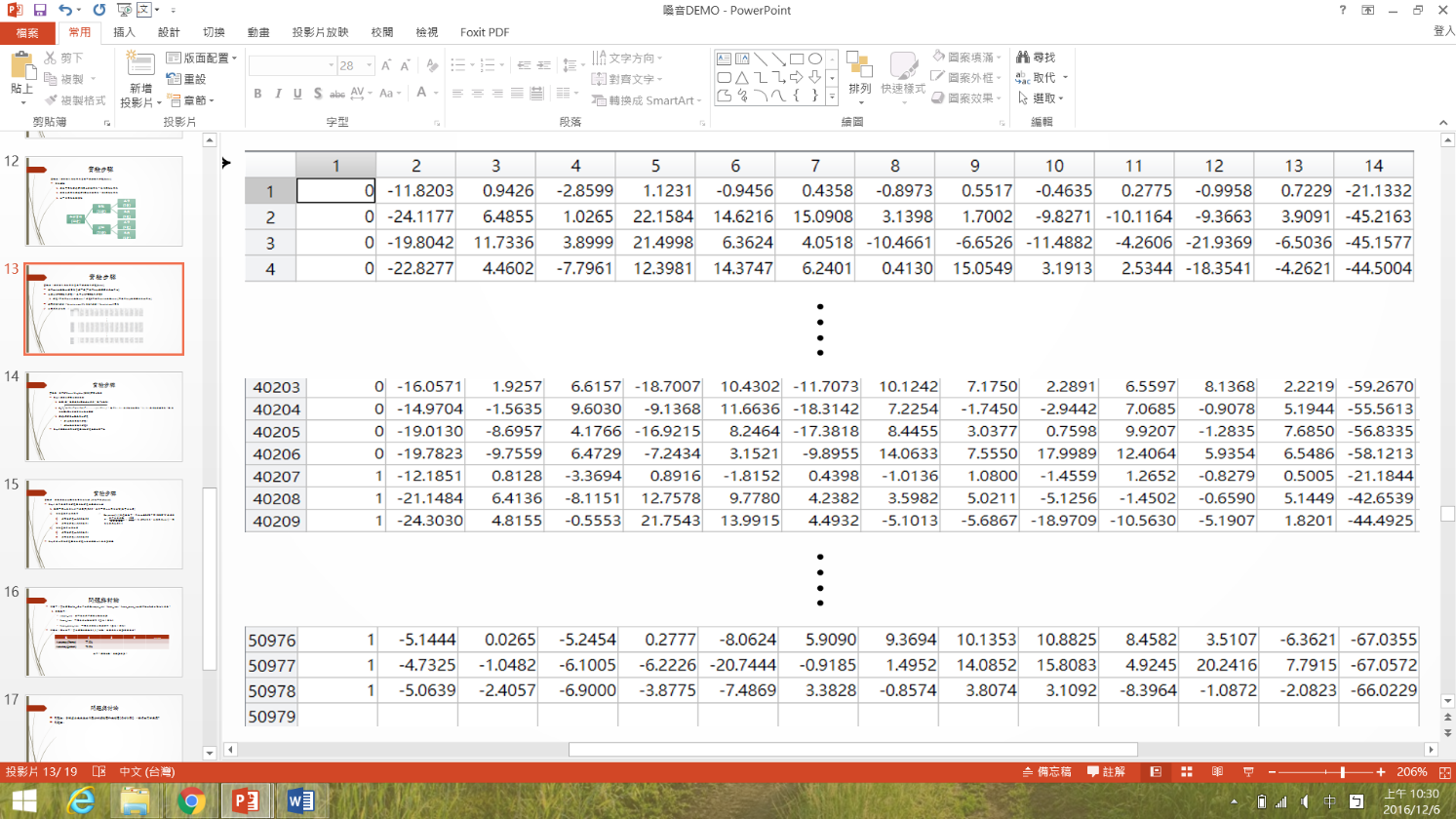
cep\_data = load(file(i).name); % Read each mfc file

number\_person(i,:) = length(cep\_data(:,1)); % The length of each mfc file

Data= [Data;cep\_data]; % Concatenate with the previous mfc file

end

Parallel functional expression = [label Data];

Figure 4、Reference results (taking training data as an example), label + 26-dimensional features

Experiment 3、Classification using K-Nearest Neighbor (KNN) algorithm

For this experimental step, we choose the KNN algorithm for classification. First, set the k value, which is defined as k close neighbors, where k is first set to 1.

Among it, are 26-dimensional T column testing data, are 26-dimensional training data, is the k nearest distance between the testing and training data. You can refer to the matlab function dist. When the experimental setting is k = 3, 5, 7..., a voting system is adopted, and the label that appears more frequently is the predicted category. For example: when there are 3 neighbors around a testing sound frame which there are 2 sick people and 1 normal person, the predicted sound frame will be regarded as sick. And use the predicted labels to compare with the testing labels.

**KNN matlab operating procedures:**

1. Use label\_person.m to concatenate the normal and sick audio file labels and mfcc in the training data to generate Voicetrain; concatenate the normal and sick audio file labels and mfcc in the testing data to generate Voicetest. Afterwards, Voicetrain is put into KNN model training, and the accuracy is calculated based on the Voicetest prediction results.
2. Click in matlab: APPS->Classification Learner->New Session.
3. Select Voicetrain for Dataset and Resubstitution Validation(No Validation) for Validation Scheme->Start session.
4. Select Fine KNN as the model, set number of neighbors(1、3、5、7).
5. Once completed, you can start training.
6. After training, as shown in Figure 5, please save the model.
7. After completing the above steps, please continue to complete Experiment 4 and conduct model testing.

一張含有 文字, 地圖, 螢幕擷取畫面, 軟體 的圖片

自動產生的描述

Figure 5. KNN training results when Number of neighbors is 1. The blue dots represent the mfcc distribution of normal sounds, and the orange dots represent the mfcc distribution of sick sounds. The horizontal and vertical axes correspond to different fields in Voicetrain.

Hint: The accuracy shown in the green box in Figure 5 is not the accuracy required for the questions and discussions in Experiment 3. Please edit it in m file and predict Voicetest, then compare the predicted results with the correct labels, and then count the number of correct predictions. The number of correct predictions divided by the number of Voicetest is what is sought.

**Experiment 3 Questions and Discussion:**

**Question 3: Complete table 1 and try to adjust the K value to odd numbers of 3, 5, and 7. What is the best setting for the most accurate value? (20%)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| K | 1 | 3 | 5 | 7 |
| Accuracy (frame) | 77.8% | ? | ? | ? |
| Accuracy (person) | 75.0% | ? | ? | ? |

Table 1. Adjust K value and record accuracy

Experiment 4、Replace the sound frame with a human and compare it with the testing data

In real life, the object of research by doctors is not scattered sound frames because it is difficult to detect which people are normal or sick in these sound frames. They need to know the actual number of people so that the doctors can know the people who are sick voice cases or normal voice cases. We will discuss people in these sick cases, so we need to convert people from sound frames. First, set the sound frame ratio threshold to 0.5 to determine a person's predicted status of normal or sick. Within the normal testing range, when the predicted label is greater than 0.5, it is considered normal (0), and when the predicted label is less than 0.5, it is considered sick (1). Within the sick testing range, when the predicted label is greater than 0.5, it is regarded as sick (1). When the predicted label is less than 0.5, it is regarded as normal (0). There are the following four situations:

1. When it is known that the person being tested is normal (0), a prediction result above the threshold of 0.5 is normal (0).
2. When it is known that the person tested is normal (0), the predicted result is sick (1) if it is lower than the threshold value 0.5.
3. When it is known that the person being tested is sick (1), the predicted result is sick (1) if it is higher than the threshold value 0.5.
4. When it is known that the person being tested is sick (1), the predicted result is normal (0) if it is lower than the threshold value 0.5.

Example

The first patient has a normal voice (0), and its sound frame length is 3420. There are 1500 sound frames in "0", and its threshold is

≒ 0.439＜0.5

This is the second situation above. The person's voice is normal, and the prediction result is that he is sick (1). Compare the person in the predicted state with the person in the actual situation, and finally record the accurate value of the predicted person and sound frame (question 2).

**Experiment 4 questions and discussion:**

**Question 4: Use experiments 3 and 4 to predict the 10 voice files in the folder other (20%)**

**Written report section**

**The codes for Experiment 1 to Question 4 and their results must be attached.**

**Question 1: Try to draw the spectrogram and waveform diagram (refer to LAB6) of**

**sick and normal sounds (choose one each), and explain the differences?**

**(20%)**

**Question 2: Try to adjust the parameters sample\_rate, frame\_time, and frame\_move\_time in mfcc\_v2.m to explain the numerical changes and significance of the mfc file for sound capture features.**

**Parameter Description：**

sample\_rate：The sampling frequency of a piece of audio (unit：HZ)

frame\_time：The sampling time length of a sound frame (unit：ms)

frame\_move\_time：The time when one sound frame overlaps with the next sound frame (unit：ms)

**Question 3: Complete Table 1 and try to adjust the K value to odd numbers of 3, 5 and 7. What is the best setting for the most accurate value?**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| K | 1 | 3 | 5 | 7 |
| Accuracy (frame) | 77.8% | ? | ? | ? |
| Accuracy (person) | 75.0% | ? | ? | ? |

Table 1、Adjust the K value and record the accuracy

**Question 4: Use experiment 3 and 4 to predict the 10 voice files in the folder other**

**Grading policy**

Report section(80%)

Experiment 1 to 4 (40%)

Questions 1 to 4 (40%)

Submit the paper report before and upload the PDF file.

M file part (20%)

Please upload the code to the portal()

**Experimental theory**

Mel-scale Frequency Cepstral Coefficients (MFCC)

MFCC is a speech feature in the field of acoustics. Since human ears have different perceptions of sound frequencies, it is widely used in speech recognition and speaker recognition. Sound signals are continuous and change frequently. In order to simplify the continuously changing sound signals, it is assumed that an audio signal with a small change amplitude is generated in a short time scale. Multiple points of this signal are sampled and assembled into a unit, called a sound frame. The duration of a sound frame is 20-40 ms. If the length of the frame is too short, the sampling points in each frame will not be enough for reliable spectrum calculation. If the length is too long, the signal of each frame will change too much big. This feature extraction will require several steps, and these steps have been shown in Figure 1.

**Step 1、Pre-emphasis**

Send a piece of sound signal x(n) to a high-pass filter, the formula show as：

Where is the pre-emphasized signal in the time domain, and *a* is a constant between 0.9 and 1.0, obtained after filtering the z-transform：

Where H(z) is the enhanced signal in the upper frequency domain, and its purpose is to remove the effects of the vocal cords and lips during phonation to compensate for the high-frequency part of the speech signal that is suppressed by articulation.

**Step 2、Window and Frame**

The input voice signal is divided into sound frames of 20ms ～ 30ms. There is an overlapping area of ～ the size of the sound frame between two sound frames to avoid excessive changes in adjacent sound frames. In order to facilitate the use of subsequent FFT, generally speaking, the sampling frequency (sampling frequency) of the audio used in speech recognition is 8 kHZ or 16 kHZ. Taking 8 kHz as an example, assuming the sound frame length is 200 sampling points, the corresponding time length is

In addition, if there are 100 points in the overlapping area, the frame rate will be

**Step 3、Fast Fourier Transform, abbreviation FFT**

Since it is difficult to see changes in the signal in the time domain, it is necessary to observe the energy distributed in the frequency domain. Different energy distributions can represent different speech characteristics.

**Step 4、Triangular Filters**

Multiply the spectral energy by a set of 20 triangular filters to find the log energy of each filter output.

It must be noted that these 20 triangular filters are evenly distributed at the "Mel Frequency", and the relationship between the Mel Frequency and the general frequency *f* is as follows:

Or

Mel frequency represents the normal human ear's perception of frequency. From this, it can be observed that the human ear's perception of frequency *f* shows a logarithmic change. For the low-frequency part, the human ear feels more sensitive. For the high-frequency part, the human ear feels increasingly rougher and duller.

**Step 5、Logarithmic Energy**

The energy of a sound frame is an important feature of speech, so it is usually added to the logarithmic energy of a sound frame (defined as the sum of the squares of the signals in a sound frame, then taking the logarithmic value with base 10, and then multiplying by 10) , so that the speech feature of each sound frame has 1 logarithmic energy.

**Step 6、Discrete Cosine Transform, abbreviation DCT**

Enter the 20 logarithmic energies Ek from step 4 into the discrete cosine transform to find the 12th step Mel-scale Cepstrum parameters. The discrete cosine transform formula is as follows:

Among it, *Ek*is the inner product value of the triangular filter and the spectrum energy calculated in step 4, and *N* is the number of triangular filters.

K-Nearest Neighbor algorithm (KNN)：

In model identification, the K-Nearest Neighbor algorithm (original K-Nearest Neighbor algorithm, also known as the KNN algorithm) is a statistics used for classification and regression, and this project is used for classification methods. In K-NN classification, the output is a classification group. The classification of a testing object is determined by the "majority vote" of its neighbors. The k nearest neighbors determine the category assigned to the object, where k is 1, 3, 5, 7...etc. odd numbers. If k = 1, the same category as the nearest point is directly assigned. Figure 6 below is a conceptual diagram:



Figure 6. This picture is a KNN concept diagram. Assume that the testing data x is used as KNN. (a) is the nearest one neighbor and the result is －. (b) is the nearest two neighbors. Because there are two values ＋－, they will not be discussed. (c) is the nearest three neighbors and the result is ＋.