Lab 2: Isolated Word Recognition

Stu.no: 1652770 Name: 梁琛

Training Progress

I rebuild the HMM model with **10** new training words: `‘你好’, ‘谢谢’, ‘再见’, '晚安', '早安', '抱歉', '梦想', '奇迹', '疾病', '灾难'`, and each one has **10** *wav* files as training samples so **10*10** samples are recorded.

1. Create audio recording samples

Firstly, I wrote a scrip `createRecords.m` to generate the training audio sample files with for loops. The code can be found in **Code 1**. With this script, I recorded 10 pieces (saved in record folder) of voice parts for each word.

```
recObj = audiorecorder;
typeName = input('Input the type:[record/test] ','s');

for  j = 1 : 10
    name = input('Input a dir name: ','s');
    mkdir(char(string(typeName) + '/' + string(name)));

    for i = 1 : 10
        disp('Start speaking: ' + string(i))
        recordblocking(recObj, 2);
        disp('End of Recording: ' + string(i));

        myRecording = getaudiodata(recObj);

        audiowrite(char(string(typeName) + '/' + string(name) + '/' +
string(name)...
            + string(i) + '.wav'), myRecording, recObj.SampleRate);
    end
end
```

The working flow is:

- Choose type `record` by typing 'record' in command line.
- Input the words in your preferable way (no matter in what language) that you are going to record.

    - By default, the recording times of the sample is **10**.
- After seeing '*Start speaking: (times)*' in the command line window, you can start to say the word.
- The recording lasts **2 seconds** and if the '*End of recording: (times)*' shows in the command line, the recording finishs.
- After that, the records will be saved in '/record'. If you are not satisfied with the records due to some noise, you can re-input the same input and the new recording will cover the old one.

2. Import the material to .mat file

Secondly, read the audio files and save the data into **training.mat** with the script `generateSample.m` (showed in **Code 2**).

```
names = {'hello', 'thanks', 'goodbye', 'goodnight', 'goodmorning',...
    'sorry', 'dream', 'miracle', 'disease', 'disaster'};
[l1, l] = size(names);
training = cell(1,l);
path = fullfile(pwd ,'record');
disp(path)
for i = 1:l
    mycell = cell(1,10);
    for j = 1:10
        filename = char(string(names{i}) ...
            + '/' + string(names{i}) +string(j) + '.wav');
        filename = fullfile(path, filename);
        disp(filename);
        [x,fs] = audioread(filename);

        x(abs(x)<0.01) = [];
        mycell{j}=x;
    end
    training{i} = mycell;
end
save('training.mat', 'training')
```

In this step, the files are imported from the `/record` subdirectory of the project directory. And I process the data so that noisy data and empty parts are filtered, by cutting the head part and the tail part whose acoustic value is under **0.01**. With **Code 3**:

```
x(abs(x)<0.01) = [];
```

3. Train the model with .mat file

Then in the `main.m`, replace the samples with **training.mat**.

```
load training.mat
```

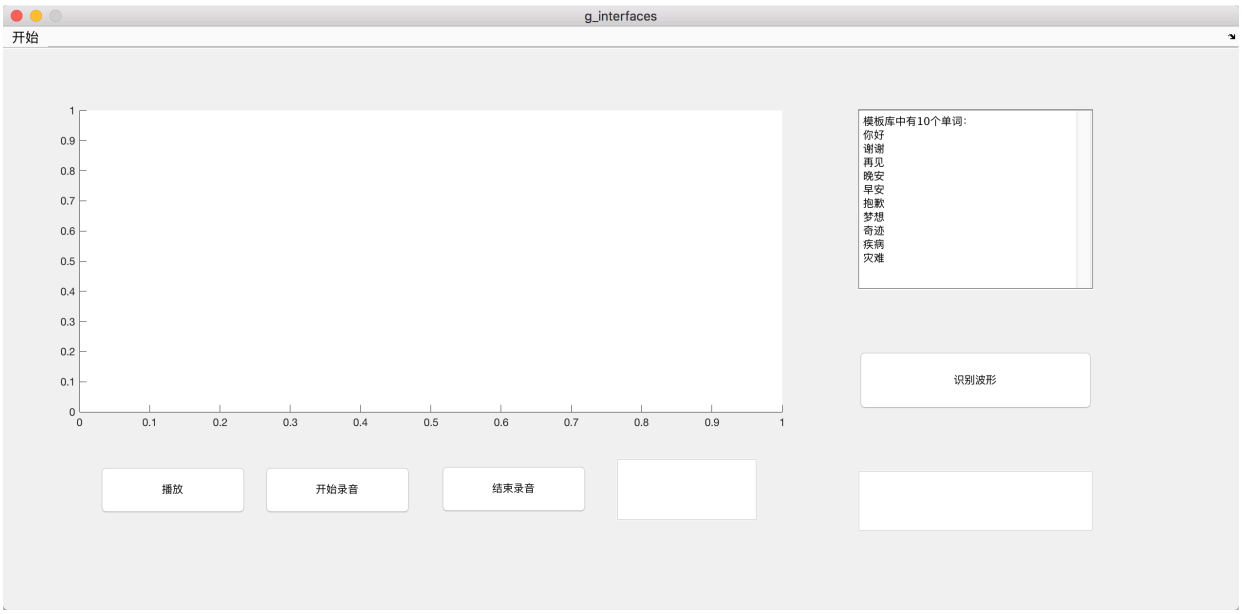
Run the script to build a HMM model and the result will be saved in the **myhmm.mat**.

I trained with 4 different HMM states: [3 3], [3 3 3], [3 3 3 3] and [3 3 3 3 3] by modifying the second parameter when invoking train function in **Code 4**, to analyze the impact of different HMM states on accuracy.

```
hmm{i}=train(sample,[3 3 3 3]);
```

4. Modify GUI

The `g_interface.fig` is modified and the new GUI is shown as below.



I added a `结束录音` button and a `开始录音` button, these two buttons help to record the test recording. The instructions will show in the text box next to the button.

Once you finish recording(press the **结束录音** button), the curve of the record will be presented and a file **test.wav** containing the record will be generated. Then you can press the `识别波形` button to recognize the word and the result will be shown in the text box locating in the right part f the GUI.

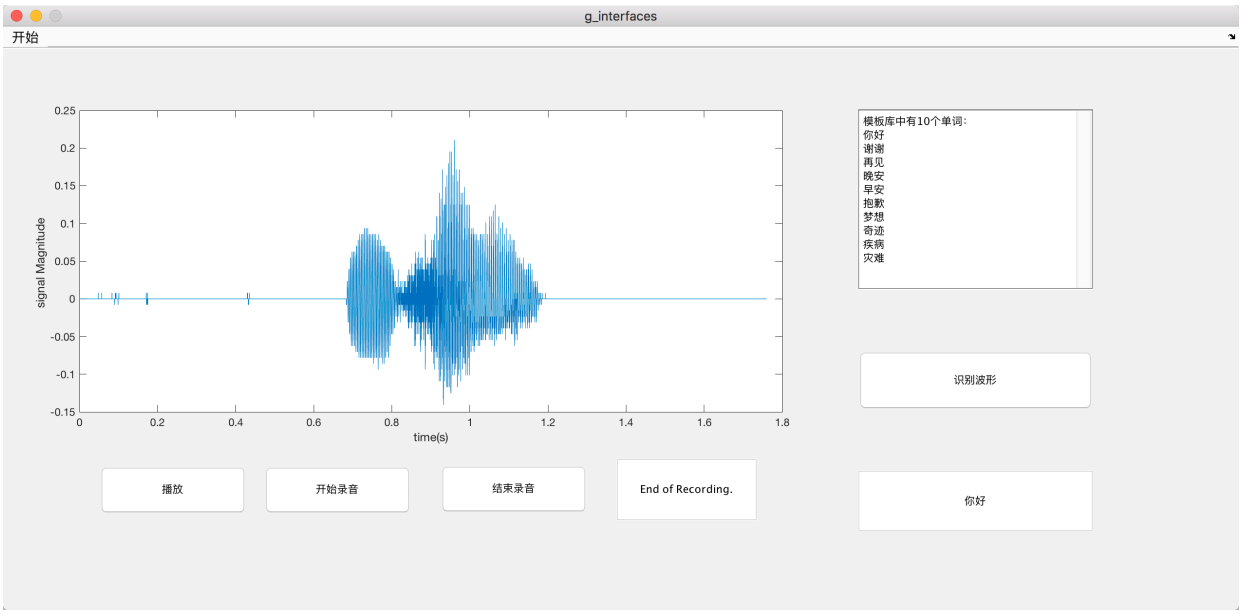
(PS: During the recognization phase, the input data has been processed using the same method, but the plot will still show the curve of the original data.)

In this part, the callback function of the `结束录音` button and a `开始录音` button have been modified. With a global variable **recObj**, it holds the recording and save it in the file. In the mean time, the content of the `recordInfoEdit` will be modified. For example:

```
set(handles.recordInfoEdit, 'string', 'Start speaking.');
```

By the way, you can still apply the original method by pressing the **start menu** to add a file and recognize the word.

After all steps finished, the related waved figure and the recognization result will be drawn accordingly, as demonstrated in the figure.



Test Progress

Test & Improvement

For getting the accuracy of recognization of each word, i wrote a script `getAccuracy.m` (shown in **Code 6**) which provide the method to test the accuracy of recognition and print the result into command line.

```
names = {'hello', 'thanks', 'goodbye', 'goodnight', 'goodmorning',...
    'sorry', 'dream', 'miracle', 'disease', 'disaster'};
[l1,l2] = size(names);
```

```
hmm = load('myhmm.mat');
path = fullfile(pwd ,'test');

for i = 1:l2
    accuracy = 0;
    for j = 1:10
        filename = char(string(names{i}) + '/' + ...
            string(names{i}) + string(j) + '.wav');
        filename = fullfile(path, filename);
        [x,fs] = audioread(filename);

        x(abs(x)<0.01) = [];
        [x1,x2] = vad(x);
        m = mfcc(x);
        for j=1:l2
            pout(j) = viterbi(hmm.hmm{j}, m);
        end
        [d,n] = max(pout);
        if n == i
            accuracy = accuracy + 1;
        end
    end
    fprintf('The accuracy of "%s" is %f\n', ...
        string(names{i}), double(accuracy) / 10);
end
```

With the same processing in the first part of this report using `createRecords.m` , and choosing `test` this time, i created the test files into the `test` subdirectory. And as i mentioned before, i tried 4 different HMM states: [3 3], [3 3 3], [3 3 3 3] and [3 3 3 3 3] , so in this part, i got 4 results and i stored them separately which will be discussed in the next part of the report.

Problem & Analysis

some problems

- There still have some uncertainty about the influence from different voice and style of speaking since I did not ask anybody else to help testing, so it remains unknow about the result from different person.(But according to the the sample provided by the teacher, i think it will.)
- Actually, before the pre-process was done to fliter the noise, The samples can't come to the convergence at all. After the disposal, it can work then. So, the pre-process seems quite important.

solutions

- Find different people with variant voice to help to test the model and summary the result.

- The pre-process to filter the noise is necessary. But if there is a better pre-process that can be used, like applying the normal distribution to fit the data and filter the data that are outside the 3-sigma range. The accuracy may be much larger.

influence of number of samples

When HMM states equals [3, 3], i calculated the average accuracy of the model with **10 * 5** samples and another model trained with **10 * 10** samples. I found that:

- The speed of convergence, in other words the speed of training, for 50 samples is **faster** than 100 samples.
- And the average accuracy of the model with **10 * 5** samples and another model trained with **10 * 10** samples is shown in the table.

P.S. Since i test the accuracy with **100** records, and the accuracy is defined as :

$$Accuracy = \frac{the\ correct\ recognization}{the\ number\ of\ all\ records}$$

so the accuracy must be an **integer**.

No.	The Number of Sample	Accuracy
1	10 * 5	69%
2	10 * 10	73%

influence of number of hmm states

The number of the states can be modified in the **main.m** . I find some articles and get the conclusion that the number of the states means how you separate your observed sequence. As a result, you can't determine the number intuitively. What we can do is to test which configure of states performs better.

And the result is listed as follows. I can't give a specific explanation about the strange results that the accuracy varies sharply, and no laws that can be followed. Maybe i still need to do more process to the records, and the model.

No.	HMM States	Accuracy
1	[3, 3]	73%
2	[3, 3, 3]	60%
3	[3, 3, 3, 3]	79%
4	[3, 3, 3, 3, 3]	67%

P.S. Since i test the accuracy with **100** records, and the accuracy is defined as :

$$Accuracy = \frac{the\ correct\ recognization}{the\ number\ of\ all\ records}$$

so the accuracy must be an **integer**.