Project 5&6

(2015 Fall)

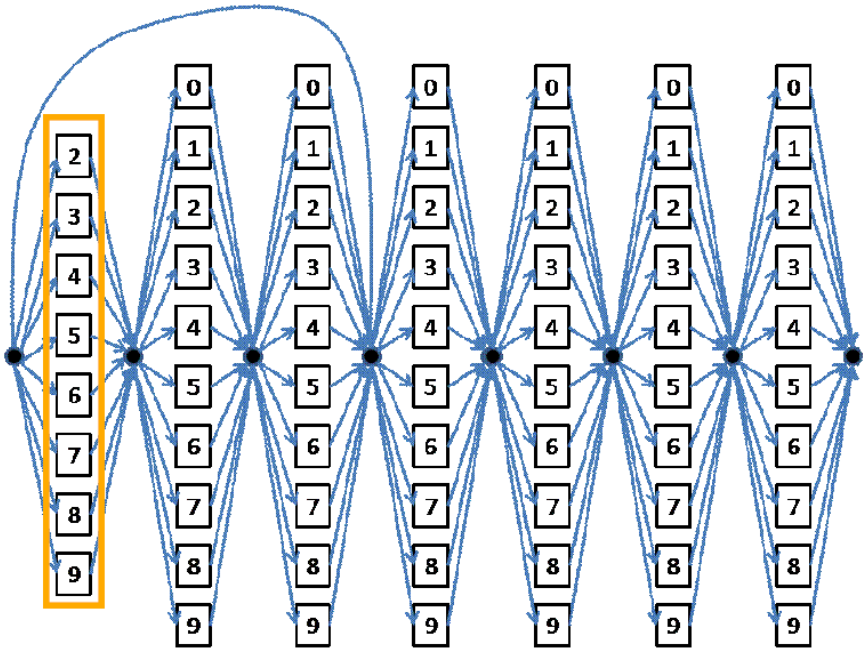
Course name：J1799d　Instructor：Ming Li　TA：Wenbo Liu

ID number	15213786	Name	GuoShouyan
Tel	15521272755	Email	2276970350@qq.com
ID number	15213795	Name	Chen Zhun
Tel	15986325893	Email	459761617@qq.com
Starting date	2015-11-20	Finished date	2015-12-13

1、 Project requirement

1.1 hw5.1

Build a continuous-speech recognition system that can recognize telephone numbers.
Speakers will say either telephone numbers with 7 digits or 4 digits.
The first digit is never 1 or 0.
Grammar is showed below:

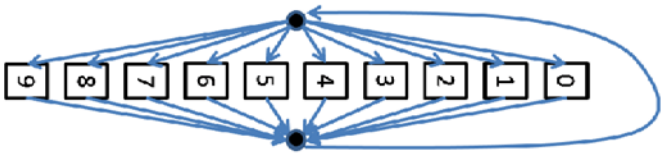


1.2 hw5.2

Do the above with backpointer table.

1.2 hw5.3

Build a system to recognize unrestricted digit strings.
It's useful to assign an "insertion penalty" to the loopback to ensure that large numbers of digits are not hypothesized randomly.
Grammar is showed below:



1.3 hw6.1

Record digits sequence as a continuous recording and train models for all ten digits for each digit string by concatenation the models for the individual digits.
Recognize all the continuous digits sequences recorded for assignment6 using these models.

1.4 hw6.2

Training models from a medium size corpus of recording of digit sequence.
"0" is recorded both as "zero" and "oh".
Use the setup from the second problem of assignment 6 to recognize.

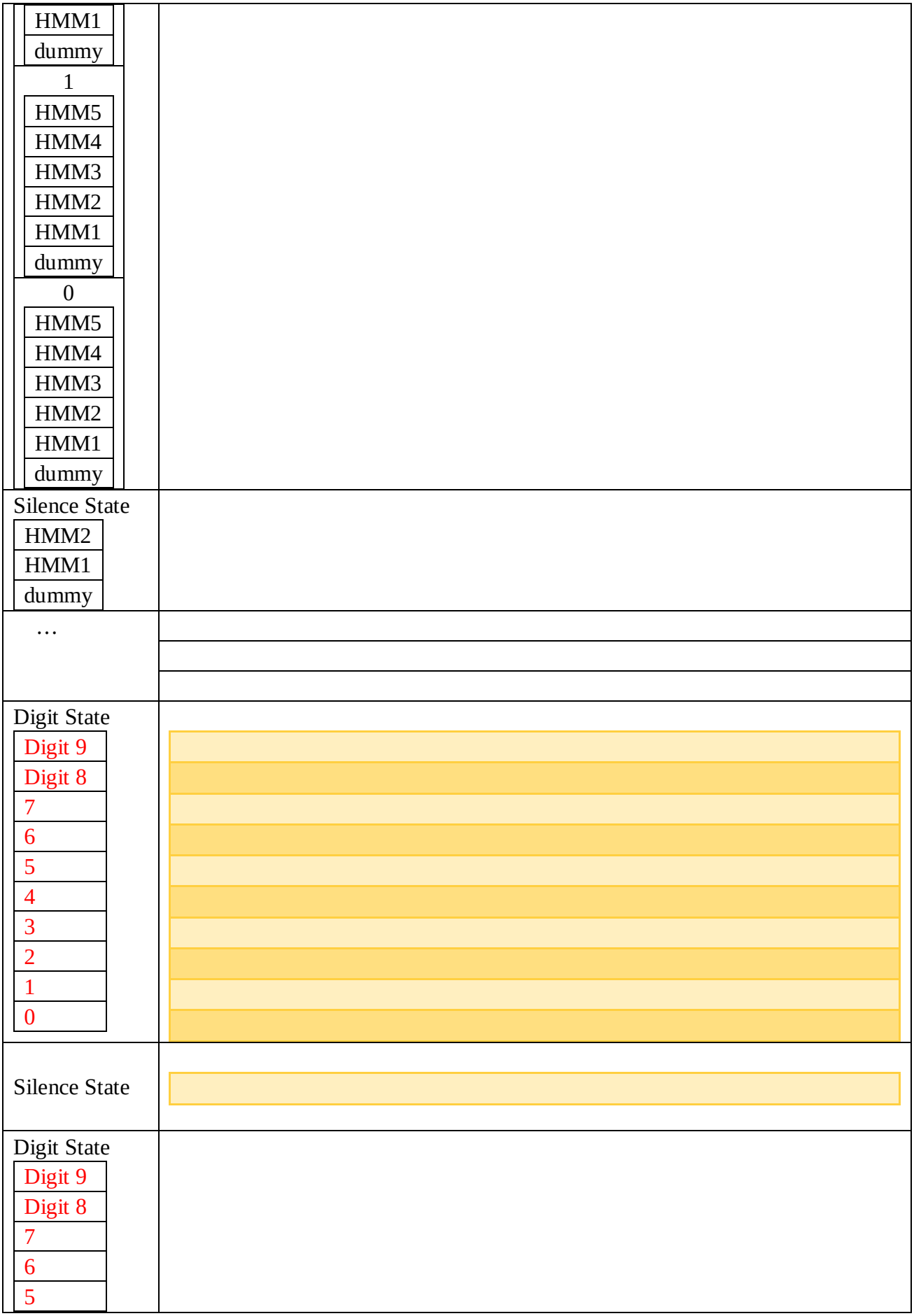
2、 Design your program

2.1 4/7digits

We separate the Tel-number into several levels of states:

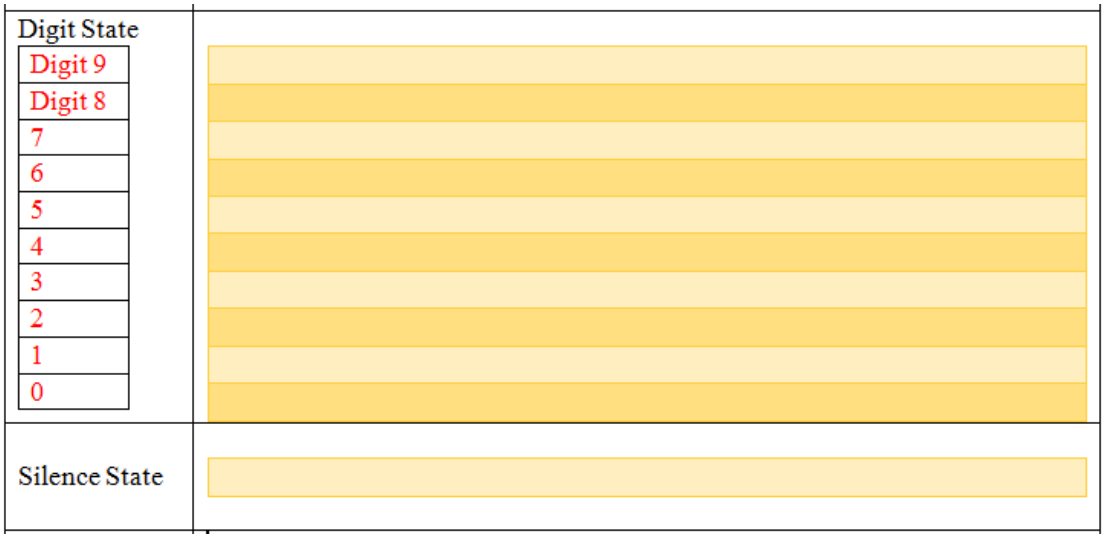
Digit State	
9 HMM5 HMM4 HMM3 HMM2 HMM1 dummy 8 HMM5 HMM4 HMM3	Trellis for dtw

		HMM2	
		HMM1	
		dummy	
	7		
		HMM5	
		HMM4	
		HMM3	
		HMM2	
		HMM1	
		dummy	
	6		
		HMM5	
		HMM4	
		HMM3	
		HMM2	
		HMM1	
		dummy	
	5		
		HMM5	
		HMM4	
		HMM3	
		HMM2	
		HMM1	
		dummy	
	4		
		HMM5	
		HMM4	
		HMM3	
		HMM2	
		HMM1	
		dummy	
	3		
		HMM5	
		HMM4	
		HMM3	
		HMM2	
		HMM1	
		dummy	
	2		
		HMM5	
		HMM4	
		HMM3	
		HMM2	

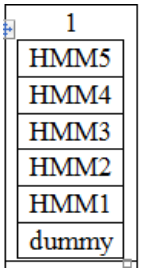

HMM1
dummy
1
HMM5
HMM4
HMM3
HMM2
HMM1
dummy
0
HMM5
HMM4
HMM3
HMM2
HMM1
dummy
Silence State
HMM2
HMM1
dummy
…
Digit State
Digit 9
Digit 8
7
6
5
4
3
2
1
0
Silence State
Digit State
Digit 9
Digit 8
7
6
5

4 3 2	
Silence State	

One TEL state is :



One digit state is:



Each Tel state is serially related and each digit state is parallel independent. We use the dtw to make the trellis in these two main levels and trace back to find out the recognized digit sequence.

2.2 unrestricted numbers of digit sequence

Based on the program of hw6.1, we change the way of just use one Tel- state to implement out program and adding loopback penalty.

The process is shown below:

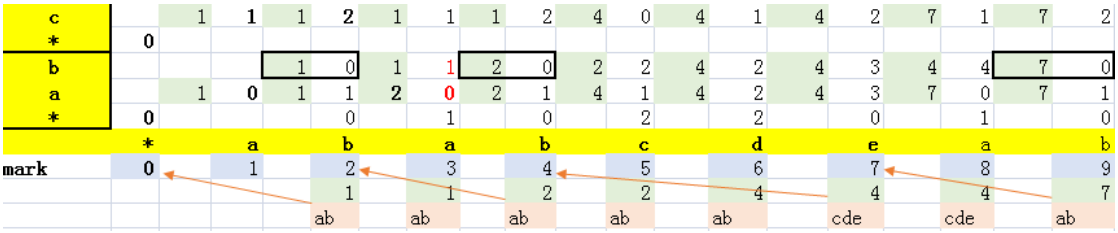
1. Initialize the boundaries for every samples
2. Calculate the mean, variances and transition probability for every segments
3. Calculate the boundaries using the dtw.
4. We repeat 2 and 3 until the score in the dtw don’t change, which means the boundaries don’t change.

2.3 backpointer table and pruning

1. Backpointer table

As there's only serial relation between Tel states for 4/7 digits recognition, it's no use to using backpointer table for hw6.1. So we applied backpointer tables in hw6.2 the recognition of unrestricted number recognition.

The main idea of backpointer table is as follow:



2. pruning

Also, by analyzing the trellis of the dtw in hw6. We think it is no use of pruning, because we can only do "pruning" inside one column while keep the access for later columns, otherwise, we cannot recognize the digits appears if we prune out the digit.

2.4 training segmented digits from continuous sequence

We do this by the following steps:

1. Read in the training materials (continuous digit sequence), and it's ground truth digit sequence.

2. Using the isolated digit templates (training into 5 HMM states , single Gaussian model), concatenate them one by one according to the ground truth digit sequence. This is the first level of Templets.

3. Fill the dots of the trellis of dtw, and trace back to find out the boundaries of each digit. Record the length of each segments and repeat until converge.

4. Place the segments of each digits into bins of the corresponding index.

5. After doing the segmentation of the current digit sequence, training the HMM states for each digit (single Gaussian) of each digit bins, and get the HMM models for each digit, update the mean, covariance and transition probability of each digits and use this new parameters for step 2.

6. Do the above step until all digit sequence are finished.

2.5 training using medium data set

We do this by the following steps:

1. Turn the audios into mfcc.
   Sample rate:8000Hz, mfcc window length:256, mfcc window's moving step 136 points of mfcc.
2. Mark down the position of the digits that we want to train, using Matlab to visualize and picking out points, calculate their corresponding positions of mfcc sequence.
3. Picking out the mfcc sequences of the corresponding digits, and sent them to train HMM models of single Gaussian, so we can get isolated digit's HMM model.
4. After we get the isolated digit's model as well as the silence. We do the segmental

training using the method mention in 2.4 to get the more accuracy models for each digit and silence.

3、 Program implementation and testing

3.1 4/7digits recognition

Main Functions:

```
float DP (vector<vector<vector<vector<float> > > > &trellis,
          vector<vector<vector<vector<float> > > > &templets, vector<vector<float> > &words,
          vector<vector<float> > &silenceTran,vector<vector<vector<float> > > &digitTran,
          vector<vector<float> > &silenceVari,vector<vector<vector<float> > > &digitVari)
```

```
vector<int> dtw(const char *Mean[],const char *Vari[],const char *Tran[],vector<float*> &mfcc)
```

```
vector<int> traceback(vector<vector<vector<vector<float> > > >
&trellis,vector<vector<float> > &silenceTran,vector<vector<vector<float> > >
&digitTran,float &cost)
```

Supporting Functions:

```
vector<vector<float> > makeWord(vector<float* > &mfcc)
```

```
vector<vector<vector<vector<float> > > > makeTrellis( vector<vector<vector<vector<float> > > >
&templets, vector<vector<float> > &words)
```

3.2 unrestricted number of digit recognition

Main Functions:

```
void DP(vector<vector<vector<float> > >  &templet, vector<vector<float> > &words,
vector<vector<vector<float> > > &score,vector<vector<vector<float> > >
&vari,vector<vector<vector<float> > >  &tran)
```

```
vector<int> dtw(const char *Mean[],const char *Vari[],const char *Tran[],vector<float*> mfcc)
```

3.2 segmental training of digit sequence

Main Functions:

```
void segment(vector<vector<vector<float> > > mfcc, vector<vector<int> > digit)
```

```
void SEGtraceBack(vector<vector<float> > &trellis,
                  vector<vector<vector<vector<float> > > > &segmented,
                  vector<int> &boundaryOfT, vector<int> &digit,
                  vector<vector<float> > &tran,vector<vector<float> > &words)
```

```
void SEG_DP(vector<vector<float> >  &templet, vector<vector<float> > &words,
vector<vector<float> > &trellis, vector<vector<float> > &vari, vector<vector<float> > &tran)
```

4、Experimental results and discussion

4.1 4/7digits recognition

We use the following sequence to test, the accuracy is 73%, and the result is as follow:

(1) 4 digits

5	0	2	7	(all correct)
3	4	9	7	(all correct)
2	3	1	6	(2 2 3 1 6)
7	6	1	4	(all correct)
4	1	6	8	(4 9 6 7 6 8 6) (6 4 1 2 6 8 6)
7	0	8	2	(7 0 7 8 6 2 7) (5 6 7 0 8 6 2)

(2) 7 digits

5	8	2	4	7	9	1	(5 8 2 4 7 9 9) all(641268)
9	5	0	1	6	7	4	(all correct)
6	9	7	0	8	3	5	(all correct)
3	9	1	2	4	6	8	(all correct)
3	4	7	5	2	6	8	(all correct)
6	4	3	0	1	2	9	(all correct)
8	6	9	1	5	0	4	(all correct)

4.2 unrestricted number of digit recognition

We use the following sequence to test, the accuracy is 80.7%, and the result is as follow:

1	5						(all correct)
8	0						(all correct)
6	3	5					(all correct)
1	3	6					(all correct)
1	7	4	5				(1 7 7 4 5) (1 7 3 4 5)
3	4	5	8				(3 4 5 8 2)
8	4	9	3	7			(all correct)
0	1	2	4	6			(0 9 2 4 6)
5	7	0	6	2	8		(all correct)
7	3	0	5	9	8		(all correct)
5	9	6	2	0	4	8	(all correct)

8	2	4	6	9	0	1		(all correct)
0	4	3	1	8	5	6	7	(0 4 3 1 8 6 5 6 7)

4.3 medium data testing

We have trained the single model, but there are some bugs inside the program. So we need more time to implement this part.

5、 Discussion

1. The first 2 part of proj5 perform well. What need to pay attention to is that we have to modify the transition probability so that we can concatenate the hmm model of single digits into a serial sequence.
2. We use single Gaussian models, and we will improve it to multi-Gaussian in the later work.
3. When doing sampling or calculating the mfcc, we must pay attention to the sampling rate, so as the mfcc features can represent more information of the audio.

6、 Division of labor

We discussed the data structure and algorithm for 1~2days. Then implement the function independently in the first stage (1~2)days. We analyzed each other's programs and results, then merged the advantages to get a more advanced version, focusing more on the overhead.

Guo Shouyan responds for the presentation, while Chen Zhun responds for the report.