

# Problem Set 3

## Isolated Digit Recognition Using HMM

Prof. Jim Rehg  
CS 7635 Computational Perception  
College of Computing  
Georgia Institute of Technology

March 22, 2002

In this problem set you will use some “canned” Matlab code to explore the performance of an HMM recognizer on the isolated digit recognition task. Please hand in your solutions at the beginning of class on the due date. If you do your write-up in long hand, it must be organized and legible so Hao and I can read it without any effort.

## 1. Front End [30 points]

Download and extract the files *digits.zip* and *hmm.zip* from the class web site. The digits file contains 135 wave files (e.g. *one15.wav*, *three09.wav*, etc.). The wave files consist of speech samples of 15 subjects speaking the digits one through nine. The speech data was sampled at 16 KHz and the files are of varying duration. They can be played using standard applications like winamp or Windows Media Player.<sup>1</sup>

The wave files have been processed to extract Mel-frequency cepstral coefficients (mfcc).<sup>2</sup> The file *cep.mat* can be found in *hmm.zip*. The cell array *cep* contains 13-coefficient mfcc vectors.  $cep\{i, j\}$  contains a 13 by  $d$  matrix corresponding to the  $i$ th digit spoken by the  $j$ th subject, with a duration of  $d$  frames. Each column of the matrix contains the mfcc coefficients for a frame of speech data.

The matlab functions *show\_cc* and *show\_cc\_dig*, found in *hmm.zip*, can be used to visualize the chunks of mfcc data. For each digit utterance, it produces a log spectral image, which looks like a spectrogram. Use these tools to examine the mfcc data and answer the following questions. For each question, illustrate your answer by printing out one or two well-chosen examples of the mfcc “spectrograms” and annotating them appropriately.

- a. Describe all of the types of variability that you see in the digit data. Which variations in the data do you think will cause the most trouble for an HMM-based recognizer?
- b. Identify a digit that exhibits large variations in pronunciation across the subjects.
- c. Which digit do you think will be the easiest to recognize? Why?
- d. Which digit do you think will be the hardest to recognize? Which other digits is it likely to be confused with?

---

<sup>1</sup>I am grateful to Prof. David Anderson in the School of Electrical and Computer Engineering at Georgia Tech for providing the digits data. See his web page for more information on speech research at Georgia Tech: <http://www.ece.gatech.edu/profiles/dva>

<sup>2</sup>Dr. Pedro Moreno at the Cambridge Research Laboratory of Compaq Computer Corporation in Cambridge, MA was kind enough to provide the code for computing and visualizing the mfcc. See his web site for many speech-related papers and projects: <http://crl.research.compaq.com/who/people/pjm/bio.htm>

## 2. Vector Quantization [30 points]

The file *epmt.zip* contains matlab files for the Probabilistic Model Toolkit, developed by Dr. Vladimir Pavlović. Follow the instructions in *README.txt* and compile this toolkit under Matlab. The VQ subdirectory contains code for vector quantization, which will be used to convert the mfcc data into discrete features.

Use the function *makefeat* to process the *cep* array, generating two cell arrays, *map* and *cost* of the same size. The map array contains feature trajectories: each 13 element mfcc vector is mapped to a discrete feature value. The number of feature values is determined by the *numclusters* parameter, which specifies how many VQ centers should be used.

The functions *show\_y* and *show\_y\_dig* display the feature trajectories for chunks of speech data obtained from VQ. Each plot in the grid shows the feature output for one utterance of the selected digit. Using *show\_y\_dig* along with *makefeat*, experiment with the effect of *numclusters* on the data. You will find that the number of clusters directly influences the “degrees of freedom” of the feature curves. As more clusters are used, individual feature curves become more “jagged”.

For the digit “one”, see if you can find a choice of *numclusters* which results in similar-looking feature trajectories across the set of speakers. Why is this desirable? What effect do you think variability in the feature trajectories will have on HMM learning? Print out the plots for this choice, along with an example of too few and too many clusters (e.g. 15 is probably too many).

### 3. HMM [40 points]

Download the file *hmm2.zip* and extract. This file contains a new version of the *makefeat* function which must be used for this problem. Note also that for this part it is important to have the epmt directory *emf* in your Matlab path ahead of the core Matlab files, to avoid a name collision with *sinv*. You will also need *vq*, *hmm*, and *mc* in your path. It may be helpful to look at the *hmm\_demo* script in the *hmm* directory. Load the matrix *ceptest*, which contains additional testing data.

Run the script *train* to train up an HMM model. Run the script *test* to test it. The function *showscore* can be used to examine the score of a particular utterance against a particular model. Answer the following questions.

- What is the digit recognition rate (from the testing phase)?
- To what extent does your previous analysis of the spectrogram and VQ data explain the performance?
- Examine the testing performance on the digit 'one'. Use *showscore* (most useful with subplot) to examine the correct and incorrect detections. Print out and annotate the resulting plots. How can the misclassifications be explained?
- What could be done to improve the performance?

### 4. HMM Improvement [Extra Credit: +20 points]

Obtain a significant improvement in the overall word classification rate for this problem. Submit any changes to the code along with documentation of the improved result.