New Mexico State University
Klipsch School of Electrical Engineering

EE589 Digital Speech Processing
Spring 2011 – Project #7
Due: 8.55am Thu. May 5

Name: _____

Grade: _____

Project

The goal of this project is to construct an Automatic Speech Recognition (ASR) system based on Hidden Markov Models (HMMs) for the purpose of isolated word recognition. The project consists both of training and testing stages. The ASR constructed in this project will consist of a 4-state ergodic model, features will be extracted using the Voicebox toolbox, and HMM parameters/likelihoods will be calculated using the HMM toolbox by Kevin Murphy.

Directions

This document will guide the student through several programming steps in constructing the ASR system. Please submit the cover sheet of this project and an email to `pdeleon@nmsu.edu` with a .zip file attachment. The .zip file should contain an archive containing all program codes (HMMtrain.m, HMMtest.m) necessary for training and testing of the ASR system (do not include wav files or toolboxes).

The required project file can be found at:

`http://www.ece.nmsu.edu/~pdeleon/EE589/Projects.html`

The VOICEBOX toolbox can be found at:

`http://www.ee.ic.ac.uk/hp/staff/dmb/voicebox/voicebox.html`

The HMM toolbox can be found at:

`http://www.cs.ubc.ca/~murphyk/Software/HMM/hmm_download.html`

Evaluation

It is assumed your code will execute in a *reasonable* amount of time on a standard PC. If your project is only partially working, then a grade will be assigned based on progress through the various programming steps. A partially working project may also require a separate demonstration by the student. Finally, Prof. De Leon reserves the right to increase (curve) project grades.

Step 1: Test Stage

In the test stage, we will compute the log likelihoods, $p(O|M_i)$ of an observation sequence, O given the data models, M_i as shown in Figure 1. We decide the work corresponding to the model with the maximum log likelihood. The test stage must be performed on pre-trained models, so typically this would not be the first step in constructing a real ASR system. However, due to the simplicity of the testing stage it will be coded first using supplied models.

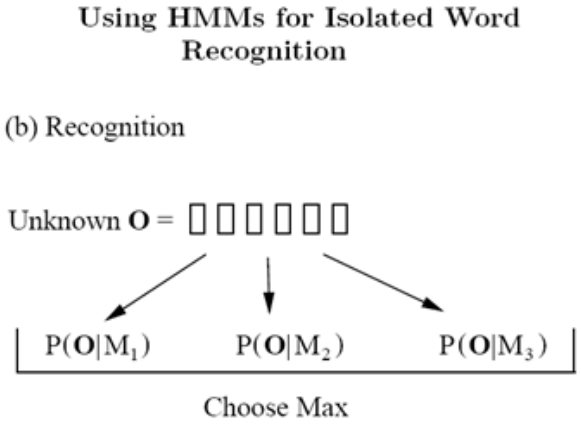


Figure 1: Test stage for ASR

(a) Complete the missing lines of code for HMMtest.m and run the script to compute the confusion matrix. Please include the confusion matrix in your report. Note that test files end in 0.wav.

(b) Determine the ASR system accuracy by selecting the maximum log likelihood for each test signal. If the model associated with the maximum, is the same as the test word, you have a correct word recognition. Accuracy is then the number of correct word recognitions divided by the number of tests.

Step 2: Training Stage

In step 1, pre-trained HMM models for each word were used for recognition. In this step, you will re-create those models and models for additional words. This will consist of selecting the initial model parameters, extracting training features, updating the HMM parameters using the EM algorithm, and saving each model to a file.

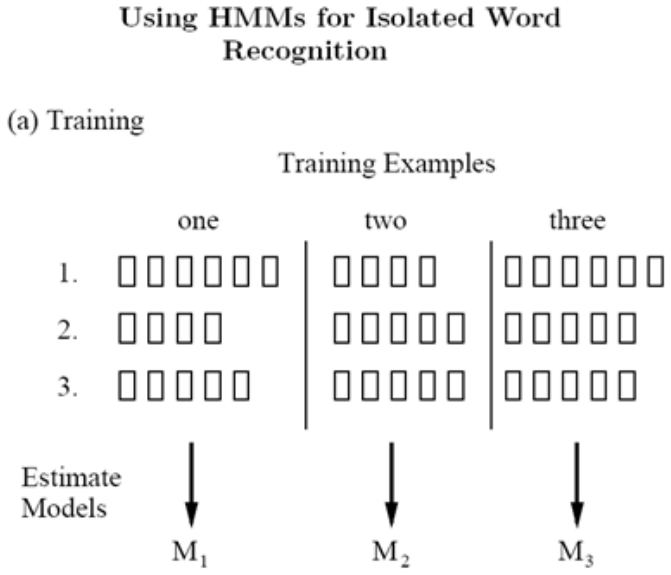


Figure 2: Training stage for ASR

Complete the missing lines of code for HMMtrain.m as follows.

(i) For each word in the dictionary, extract the training features from each of the 10 utterances and concatenate them. Note that files for training end with 1-10.

(ii) Specify appropriate initial parameters for the HMM. Remember, we are interested in good starting parameters for an *ergodic* model.

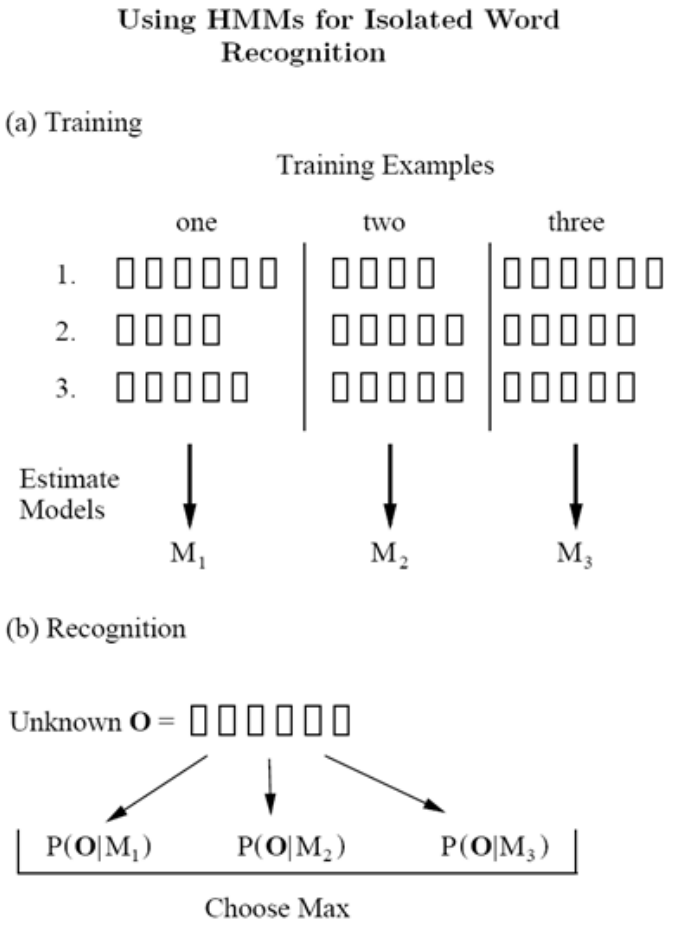
(iii) Update model parameters using the EM algorithm. Call syntax for the EM algorithm is as follows:

```
[convergedLL, priors,transition_matrix, mean_vectors, covariance_matrix,...
 mixture_weights] = mhmm_em(train_features, initial_priors, initial_transition_matrix,...
  initial_mean_vectors, initial_covariance_matrix, initial_mixture_weights);
```

(iv) Model parameters for each word are contained in the structure HMM_Params. Save the structure in a file named "word.mat" where "word" is the word being trained on.

Step 3: The Complete ASR System

Now that we have both the training and testing stages working we will put them together to make a complete isolated word ASR system.



(a) Run HMMtrain.m, this will train/save models for each of the 18 dictionary words using 10 training signals for each word.

(b) Run HMMtest.m using the "0.wav" test signals for each word and note the system accuracy for the 18 dictionary words.