**ECE 8527: Introduction to Machine Learning and Pattern Recognition**

# HW No. 4: Markov Processes, HMMs and EM Estimation

1. Discrete hidden Markov models:

Recall the fully ergodic hidden Markov model discussed in class. Create *N* random sequences of length 100 for each of these models:

Class 1:       

Class 2:       

By convention, assume the output symbols *L*, *M*, and *H* correspond to the discrete symbols (e.g., a typical output sequence will be “*LLMMMHHHLLMMMHH*...*LMHL*...”). Treat each of these two sets as your training sets. Re-seed your random number generator (if applicable) and generate *M* random sequences of length 100 for your test data – again generating *M* sequences for each class.

(a) Plot the likelihood of the training data given the models as a function of the number of Baum-Welch training iterations (using only the training sets). Comment on convergence of this plot. Select a reasonable value for the remaining tasks.

(b) Set *M* = *100*, and plot the probability of error for classifying the test data as a function of *N* (the amount of training data). Do this using an ML approach – for each test vector, compute the likelihood it could have been produced by the model, and choose the model which has the greater likelihood. Justify your results.

2. Choose a reasonable value of *N* and *M*, and repeat 1(b) using HMMs with a different number of states. Plot the probability of error as a function of the number of states over the range [*1*,*10*]. Can you infer the number of “underlying states” in the model from this plot? Explain.

3. Continuous hidden Markov models: Repeat problem 1, but replace the discrete emission distributions with multivariate Gaussian distributions. Assume a mean vector of dimension 2, two Gaussian distributions per state, and use the mean and covariance parameters shown on the following page. Also experiment with the number of Gaussian mixtures. Plot the probability of error as a function of the number of mixtures components allocated to each state (using the same number of mixtures per state).

4. Computation: Plot the computation time required to train the models of problem 3 as a function of the number of training sequences, *N*, and the number of iterations of training. Similarly, plot the computation time as a function of the number of test sequences, *M*. Explain whether these plots match your theoretical predictions for computational complexity.

Class 1: 

Class 2: 