Assignment 04: Hidden Markov Models: Convolutional Coding Application

1. For this problem, we will use convolutional codes as an example application for Hidden Markov Models using the built in functionality for convolution code encoding and decoding in MATLAB.

   The MATLAB program "GenReceivedConvEncodedData.m" on the course website gives a segment of MATLAB code that generates a random message, encodes it with a convolutional code, and then propagates it over a binary symmetric channel to obtain a corresponding set of received data bits. Note that running the program again generates a different set of bits unless the random number generator is consistently set to the same state before invoking this script. Data generated from one run of this program is provided in the MATLAB file "receivedData.mat".

   (a) Decode the data in the file "receivedData.mat" using a Viterbi decoder (for the same code used in the encoder). Identify and list the locations of the bits in the received data that the decoder identifies are in error.

   Hint: The MATLAB function find() will be helpful for you in this exercise. See the provided program for an example for how it is utilized.

   (b) Assuming that the coded bits were communicated over a channel with bit error probability $p_a = 0.03$ (3% error rate), decode the data in the file "receivedData.mat" using a bit-by-bit maximum a posteriori probability decoder. Identify and list the locations of the bits in the received data that are identified as being in error based on the decoded data. Compare the error locations you have found in this part with those in Part 1a and indicate if these are same or different. Also examine the estimated posterior probabilities for the decoded bits to see how confident the decoder was in its decisions.

   Note: You will have to use the value of the bit error probability for the Binary Symmetric channel in this process and understand the input format for the MATLAB bit-by-bit MAP decoder.

   (c) For the decoder, you assumed a bit error probability $p_a = 0.03$ for the Binary Symmetric Channel, whereas you did not know the true bit error probability. Estimate the bit error probability for the using one iteration of the so-called "hard" EM algorithm, i.e., estimate the bit error probability for the Binary Symmetric Channel as

   $$\hat{p}_a = \frac{\text{Number of Postulated Errors}}{\text{Length of Encoded Data}} \tag{1}$$

   (d) **Bonus:** Estimate the bit error probability p_a for the binary symmetric channel using the actual EM algorithm. Give the estimates $\hat{p}_a^{(i)}$ for iterations $i = 1, 2, 3, 4, 5$ of the EM algorithm. Comment on what you see in the estimates, with regard to convergence and to what you see as the limiting factor in improving the estimate of p_a. Note that in order to do this part of the problem you have to either write your own HMM decoder, or carefully understand the Matlab convolutional decoder and use its output appropriately.

2. **Bonus:** Repeat the exercises in Parts 1a, 1b, and 1c by generating your own data for different values of the binary symmetric channel bit error probability p, while keeping the assumed bit error probability value at $p_a = 0.03$. Comment on the results.