ESE 650 -PROJECT 3
LEARNING IN ROBOTICS
GESTURE RECOGNITION

Anushree Singh
4th March 2014

Introduction:

In this project, our aim was to design and implement a Hidden Markov Model (HMM) that matches or predicts the set of training sequences provided with the project. We were given the IMU data, accelerometer, gyroscope and magnetometer data. These are the observation sequences. The underlying gestures are thus converted into sequential symbols. Every gesture is represented by an HMM whose parameters are learned from the training data. Based on the most likely performance (Log-Likelihood), the gestures can be detected by evaluating the trained HMM.

Collecting IMU and Vector Quantization:

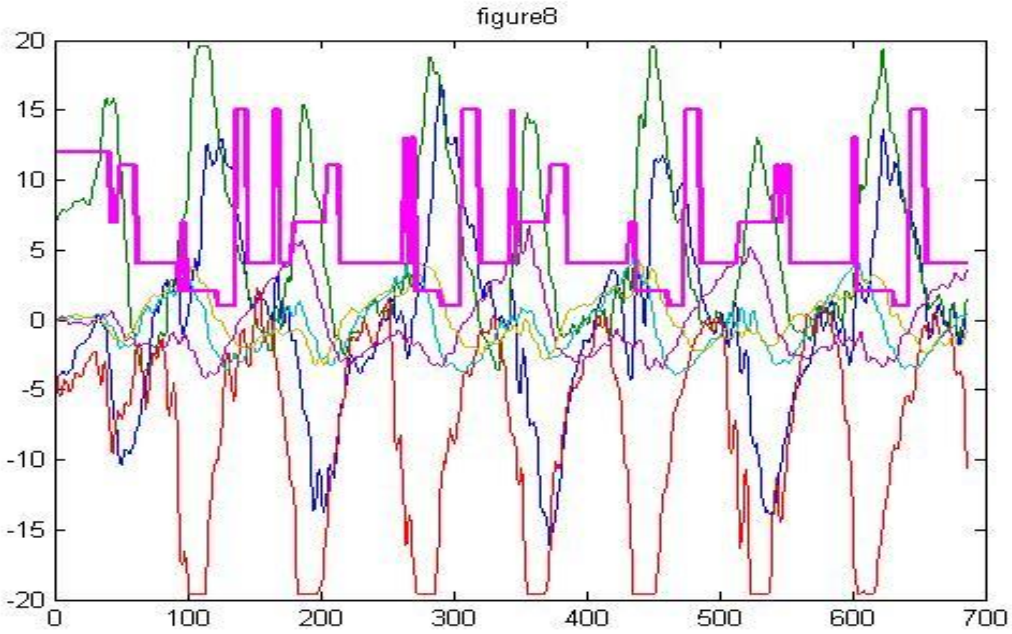
As the IMU data was collected from a mobile phone, there was no need of taking care of the bias and sensitivity. However the some of the training data had redundant values at the start of the motion which was pruned out for efficient training. The IMU file consists of timestamps, accelerometer, gyroscope and magnetometer data of which I used only the accelerometer and gyroscope data as suggested.

For quantizing the data (Vector Quantization), Kmeans is used on the data, giving input as 15. This figure was obtained by cross validation, wherein I kept aside one or two examples of each of the gesture and trained the models on the remaining ones, testing them on the left out ones.

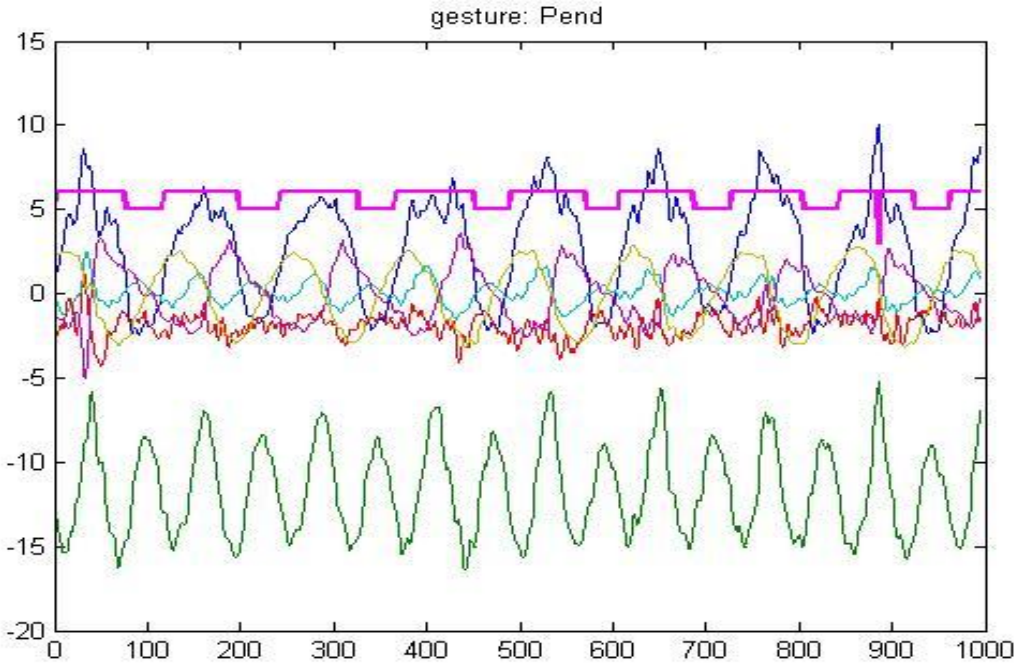
Vector Quantization converts the observation values to unique observation symbols. Here these symbols are denoted by one of the 15 clusters that it may belong to. The centroids obtained from Kmeans are stored in a matfile, loaded when the data comes in, the Euclidian distance between the data and the centroids are compared and the one closest to it is assigned to it as its observation symbol.

The plots of the training sequences (examples) after Vector Quantization and pruning are as below.

FIGURE 8 GESTURE:



GESTURE PENDULUM:



Recognition:

As described in the Tutorial on Hidden Markov Model and Selected applications in speech recognition, the basic Recognition problems were taken up one by one.

1. Given the observation sequence and model, efficiently computing the probability of the Observation sequence given model.

For this stage, I implemented the Forward-Backward Algorithm. The transition probability matrix A, emission matrix B, and the initial probability Pi are randomly initialized. In the forward procedure, the probability of observing partial observation sequence until time t and state Si at time t given the model is calculated. (alpha_t(i)). Also the loglikelihood of the probability of an observation sequence given a model was calculated.

In the backward procedure, the probability of partial sequence from t+1 to end given the state Si and the model is calculated. (beta_t(i)).

An important factor was to scale the values of alpha and beta in order to avoid any underflow error in the future.

2. Adjusting the model parameters in order to maximize the probability of observation symbol given the model.

First calculated the probability of being in state Si at time t and at state Sj at time t+1 given the model and the observation sequence. (epsilon_t(i,j)). Then calculated the probability of being in state Si at time t given the observation sequence and the model. (gamma_t(i)). Then calculated the new A, B and Pi from gamma and epsilon and repeated this procedure iteratively until convergence as described by the Baum-Welch method.

These steps were followed and a set of model parameters for each gesture were generated that basically defined each of them.

Testing and Results:

The training examples kept aside were used for testing the HMM models that I made for each of the gesture. The input data was converted to observation symbols by comparing them to the cluster centroids stored initially. Then running this through every gesture model the log-likelihood of the model given the sequence was stored and finally ranked according to the values. This gave fine results on held out training samples (testing with and without pruning).

Finally the models for built on all the training data and stored in a matfile. I have written a test script which will be used when the test set comes in, this will predict the unknown gestures giving the first three ranks according to the log-likelihood.

Future Work:

I plan to self implement the vector quantization part instead of using Kmeans Algorithm, using the Linde–Buzo–Gray algorithm (LBG). And also instead of randomly initializing the parameters of the model (A,B,Pi) I would like to look into other means of a more intelligent initialization.