# Gesture Recognition Report

The report mainly contains five parts, introduction, problem formulation, technical approach, tests results and brief summary.

The key points in the project includes converting continuous measured data into discrete states; constructing HMM model for training data; dealing with the issue of underfitting and overfitting.

## I. Introduction

This project is aimed to use IMU sensor readings from gyroscopes and accelerometers to train a set of Hidden Markov Models and recognize different arm motion gestures. After estimating models for various gestures, it should then be able to classify unknown arm motions in real-time using the algorithm.

## II. Problem formulation

### A. *Feature Vector Selection*

For training algorithm, it's necessary to select proper types of data and convert that into corresponding feature vector then apply to the algorithm. For this project, IMU gives us 6 – dimension data at each time step. We need to generate suitable feature vector based on obtained data. Note that sometimes higher dimensional vector may cause overfitting problem while low dimensional case is likely to lead an underfitting. Therefore, we need to find the suitable form of feature vector which can bring us the higher accuracy.

### B. *Discretization of Measurement Data*

Since we use HMM to train the model which is good for discrete space while our measurements are continuous, therefore, it's necessary to come up some approaches to discrete dataset correctly. After discretization, each data has its own observation state which we can apply into the HMMs.

### C. *Hidden Markov Model Construction*

We need to use HMM models to describe the corresponding motion gestures.

The basic parameters of HMM are transition matrix T, observation matrix B and the prior Pi. The basic idea of HMM estimation is the EM algorithm, that is, use the initial guessing of above parameters to estimate some underlying variables, and then update basic parameters with training data in the M step.

During the training process, we need to deal with some practical issues including initialization problem, overflow problem, probability computation problem and so on.

### D. *Underfitting and Overfitting Issue*

Underfitting and overfitting issues are common seen in learning problem. The basic idea is, if we train the model so well, it might not be generalized for new coming data which we call the overfitting issue.

However, if the model is not good, it might cause underfitting problem which is also bad for recognition. In this project, we should consider the dimension of feature vector, number of hidden and observation states, and also the iteration times in the EM step matters a lot.

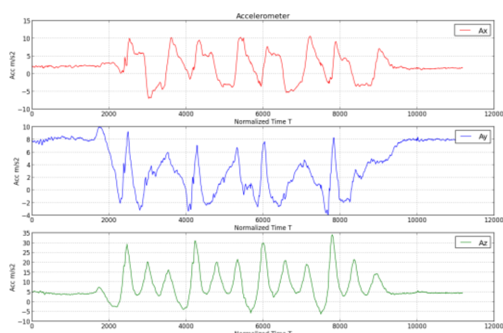
## III. Technical Approach

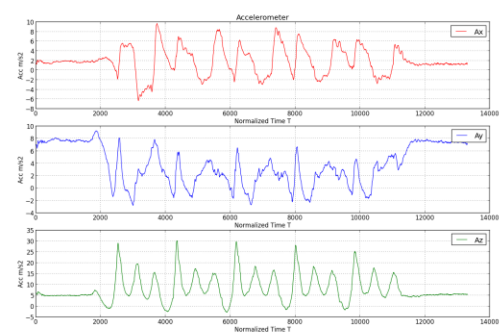
This section shows some several technical approaches in my code to deal with mentioned problems above.

### A. *Feature Vector*

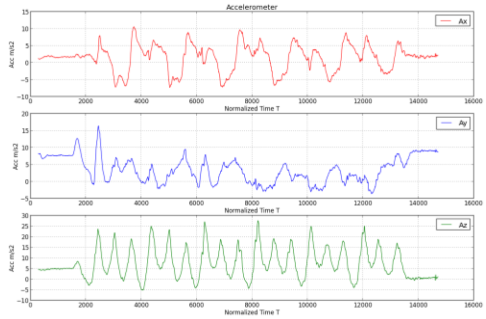
Since we are given 3 accelerometers and 3 Gyroscope measurements, in fact, we can create a 6 – dimensional feature vector along the time steps to represent each motion gesture. However, higher dimension of feature vector may lead to a potential overfitting problem. In that case, I choose to plot out data from different gesture set to figure out whether there exists some certain distinguish dimension to represent each gesture well.

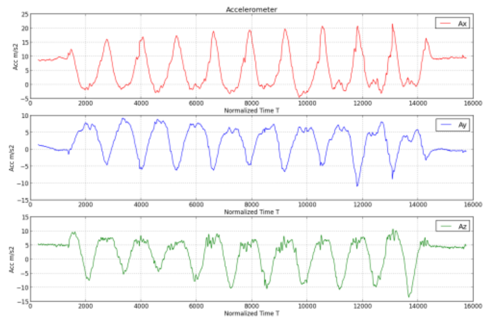
Below figures show the accelerometer of 'beat3'.




Two figures are pretty similar, and we can find the accelerometer in Z axis keeps the most consistency. Now we compare accelerometer from different gestures, such as 'beat3' and 'wave'.




It’s obvious that two accelerometers are totally different. That means, it’s reasonable that we only use three dimensional feature vector to represent our gesture which can also distinguish various motions well.

For me, I still use 6 – dimensional feature vector since that is the safest approach in my view, since we are given enough number of training data so that reduce the probability of overfitting problem, however, the training time and memory will increase, but we can use vectorization to deal with that issue.

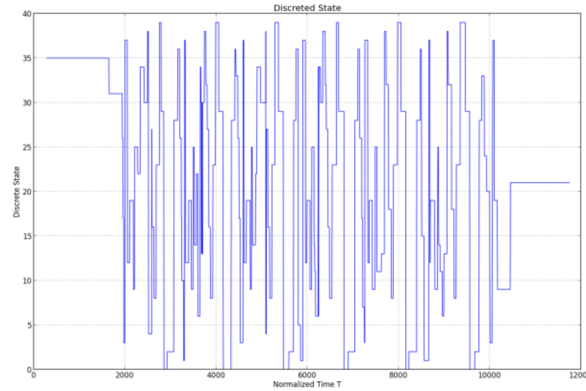
### *B. Discretization Data*

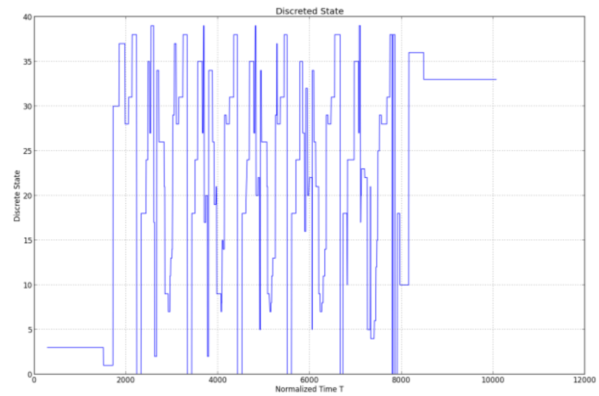
In order to apply data into HMM model which is optimal for discrete space, it’s necessary to convert measured continuous data into discrete version.

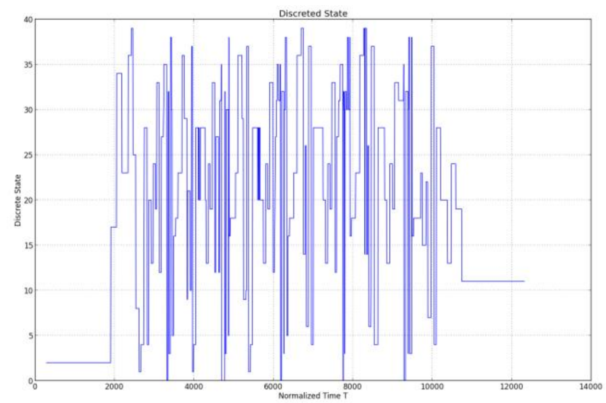
One reasonable approach is using K – means algorithm to find underlying relationship or clusters of data. The issue is, we need to decide the number of expected clusters of data. We can just guess some value of K to run the cluster function, but the most recommended way is using the cross validation set and apply them into trained model with different value of K, then get the max likelihood as well as the optimal K value. If the cluster number is so large, we may have overfitting problems as well as code speed issue.

For example, we I set the cluster number equal to 30, I will get about 70% accuracy when applying my trained model to my training dataset, but my test result seems more reasonable. However, after I increase the K value to 45, the accuracy of training dataset increases to almost 95%, but that might lead to overfitting problem. So K value selection is a trade – off issue and we need to consider about that carefully.

Below shows discretization result of ‘wave’, ‘circle’ and ‘eight’, you may find some distinguish properties between them.




### *C. Hidden Markov Model Construction*

The idea of HMM model is quite straightforward, but we need to deal with following practical problems carefully.

First, each EM algorithm needs us to guess the initial state of parameters. For HMM model, initialization is quite important. If we cannot handle that well, the algorithm may result a local optimal problem. Unluckily, there is no simple or straightforward way to deal with that initialization problem. Experience shows, randomly or uniform initialization for matrix T and Pi is good, but we need to deal with matrix B carefully. The safest way is still using cross validation set to find the optimal initialization of parameters.

Second, value scaling issue. When implementing the re – estimation procedure in HMMs, we need to compute two helper matrix in forward and backward step. If we implement that without value scaling, it's highly possible to meet overflow problems such as dividing by zero. In order to deal with that problem, value scaling is necessary. According to the reference paper, adding certain scaling coefficient is fine.

Third, model selection. We need to decide the type of HMMs such as ergodic, left-right or other forms, choice of model size which is the number hidden states, and the choice of observation symbols. There is no simple way or equation to figure out above issues, what we can do is trying different combinations of models to get the optimal one.

## IV. TEST RESULTS

I trained total six models to represent corresponding motion models, including beat3, beat4, circle, wave, inf and eight. Also I apply the single and multiple test sets into my estimated algorithm to recognize its motion model.

Below is my result when applying training dataset into my algorithm, and I can get about 100% accuracy of them. Please feel free to check the result by running the code directly and it will only cost 5 sec.

Train Data (42/42, 100%)

```
The test file [beat3_01] is predicted as the gesture [beat3] with confidence 57.1041%.
The test file [beat3_02] is predicted as the gesture [beat3] with confidence 47.5008%.
The test file [beat3_03] is predicted as the gesture [beat3] with confidence 48.9089%.
The test file [beat3_06] is predicted as the gesture [beat3] with confidence 82.4797%.
The test file [beat3_08] is predicted as the gesture [beat3] with confidence 80.3095%.
The test file [beat3_31] is predicted as the gesture [beat3] with confidence 48.9145%.
The test file [beat3_32] is predicted as the gesture [beat3] with confidence 48.2277%.
The test file [beat4_01] is predicted as the gesture [beat4] with confidence 82.1286%.
The test file [beat4_03] is predicted as the gesture [beat4] with confidence 83.5474%.
The test file [beat4_05] is predicted as the gesture [beat4] with confidence 80.6000%.
The test file [beat4_08] is predicted as the gesture [beat4] with confidence 79.9527%.
The test file [beat4_09] is predicted as the gesture [beat4] with confidence 81.9155%.
The test file [beat4_31] is predicted as the gesture [beat4] with confidence 83.4703%.
The test file [beat4_32] is predicted as the gesture [beat4] with confidence 81.3205%.
The test file [circle12] is predicted as the gesture [circle] with confidence 92.9721%.
The test file [circle13] is predicted as the gesture [circle] with confidence 93.2801%.
The test file [circle14] is predicted as the gesture [circle] with confidence 92.7851%.
The test file [circle17] is predicted as the gesture [circle] with confidence 93.4266%.
The test file [circle18] is predicted as the gesture [circle] with confidence 93.4751%.
The test file [circle31] is predicted as the gesture [circle] with confidence 94.7280%.
The test file [circle32] is predicted as the gesture [circle] with confidence 94.9208%.
The test file [eight01] is predicted as the gesture [eight] with confidence 81.2628%.
The test file [eight02] is predicted as the gesture [eight] with confidence 84.2362%.
The test file [eight04] is predicted as the gesture [eight] with confidence 85.8612%.
The test file [eight07] is predicted as the gesture [eight] with confidence 84.4820%.
The test file [eight08] is predicted as the gesture [eight] with confidence 84.3152%.
The test file [eight31] is predicted as the gesture [eight] with confidence 84.2900%.
The test file [eight32] is predicted as the gesture [eight] with confidence 84.2541%.
The test file [inf11] is predicted as the gesture [inf] with confidence 86.3779%.
The test file [inf112] is predicted as the gesture [inf] with confidence 84.9995%.
The test file [inf13] is predicted as the gesture [inf] with confidence 85.1589%.
The test file [inf16] is predicted as the gesture [inf] with confidence 84.8939%.
The test file [inf18] is predicted as the gesture [inf] with confidence 87.3175%.
The test file [inf31] is predicted as the gesture [inf] with confidence 84.4235%.
The test file [inf32] is predicted as the gesture [inf] with confidence 80.6952%.
The test file [wave01] is predicted as the gesture [wave] with confidence 88.1089%.
The test file [wave02] is predicted as the gesture [wave] with confidence 89.3383%.
The test file [wave03] is predicted as the gesture [wave] with confidence 87.0251%.
The test file [wave05] is predicted as the gesture [wave] with confidence 87.1178%.
The test file [wave07] is predicted as the gesture [wave] with confidence 86.9063%.
The test file [wave31] is predicted as the gesture [wave] with confidence 78.9329%.
The test file [wave32] is predicted as the gesture [wave] with confidence 83.6188%.
```

Test Data Single

```
The test file [test1] is predicted as the gesture [wave] with confidence 78.9329%.
The test file [test10] is predicted as the gesture [circle] with confidence 94.9208%.
The test file [test11] is predicted as the gesture [inf] with confidence 80.6952%.
The test file [test12] is predicted as the gesture [beat4] with confidence 81.3205%.
The test file [test2] is predicted as the gesture [beat4] with confidence 83.4703%.
The test file [test3] is predicted as the gesture [eight] with confidence 84.2900%.
The test file [test4] is predicted as the gesture [beat3] with confidence 48.9145%.
The test file [test5] is predicted as the gesture [wave] with confidence 83.6188%.
The test file [test6] is predicted as the gesture [inf] with confidence 84.4235%.
The test file [test7] is predicted as the gesture [eight] with confidence 84.2541%.
The test file [test8] is predicted as the gesture [circle] with confidence 94.7280%.
The test file [test9] is predicted as the gesture [beat3] with confidence 48.2277%.
```

Test Data Multiple

```
The test file [test1] is predicted as the gesture [wave] with confidence 85.3864%.
The test file [test10] is predicted as the gesture [circle] with confidence 54.3151%.
The test file [test11] is predicted as the gesture [inf] with confidence 85.0991%.
The test file [test12] is predicted as the gesture [beat4] with confidence 79.3236%.
The test file [test2] is predicted as the gesture [beat4] with confidence 82.6477%.
The test file [test3] is predicted as the gesture [eight] with confidence 85.0971%.
The test file [test4] is predicted as the gesture [beat3] with confidence 49.3704%.
The test file [test5] is predicted as the gesture [wave] with confidence 86.8091%.
The test file [test6] is predicted as the gesture [inf] with confidence 44.6583%.
The test file [test7] is predicted as the gesture [eight] with confidence 85.6531%.
The test file [test8] is predicted as the gesture [circle] with confidence 89.8125%.
The test file [test9] is predicted as the gesture [beat3] with confidence 50.0333%.
```

## V. BRIEF SUMMARY

After completing three robotics projects, I get to know the importance of data. Since all trained data is obtained from actual robot measurements which is raw, converting data into correct digital becomes quite important for the project. Even with a perfect algorithm, you cannot get the correct result without proper transformed data. For this Gesture recognition project, generating suitable feature vector and discretization data is quite important.

What's more, the understanding of HMMs is also important. It's better to master the basic algorithm of forward, backward and Baum Welch method to estimate the model. And it's for sure that we will meet more problems when implementing the model such as valuing scaling, initialization issue and model selection. In addition, in order to estimate the observation sequence better, most of HMMs will include Viterbi decoding part.

Besides, overfitting and underfitting issue always happen in such kind of machine learning problem. In this project, many factors need the consideration, such as the number of hidden states and observation states, the dimension of feature vector, and the iteration times which contributes to the convergence state of estimated model.

After all, when testing new coming data, we should use log likelihood instead of common probability computed from forward step since we have implement the value scaling. And we choose the class with the max likelihood value as recognition result.