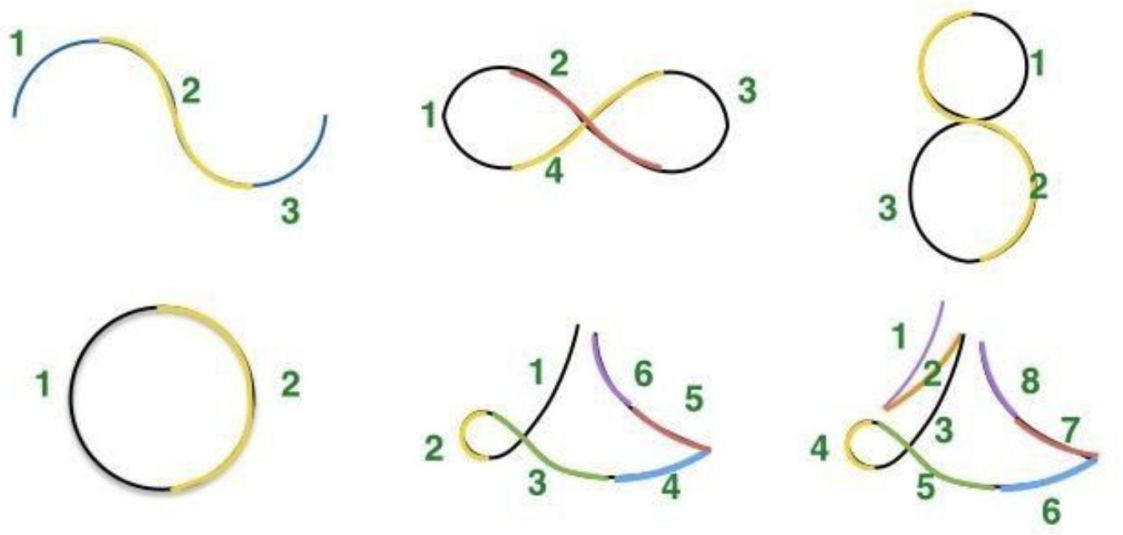
I/ Introduction

The goal of this assignment is to detect the motion gestures of a device based on the IMU readings. There are 6 different gestures: Wave, Infinity, Eight, Circle, Beat3, Beat4.



Using K-Means clustering to process IMU readings and Hidden Markov Model to predict the gestures, I achieve 80% accuracy on the Dev data.

II/ Problem Formulation

The problem is how to interpret the continuous IMU data in time, then calculate the probability that the data belongs to 1 of the 6 gestures.

III/ Technical Approach

1. Feature analysis

The training data include 30 IMU readings of acceleration and orientation over time of the 6 gestures.
To simplify the problem, I need to discretize this motion data. I use K-Means model to cluster the IMU readings into several clusters. Imagine that each cluster is a sub-motion inside a

gesture. For example, drawing a Wave gesture requires 1 clock-wise up motion, 1 sigmoidal motion and 1 anti-clockwise up motion.

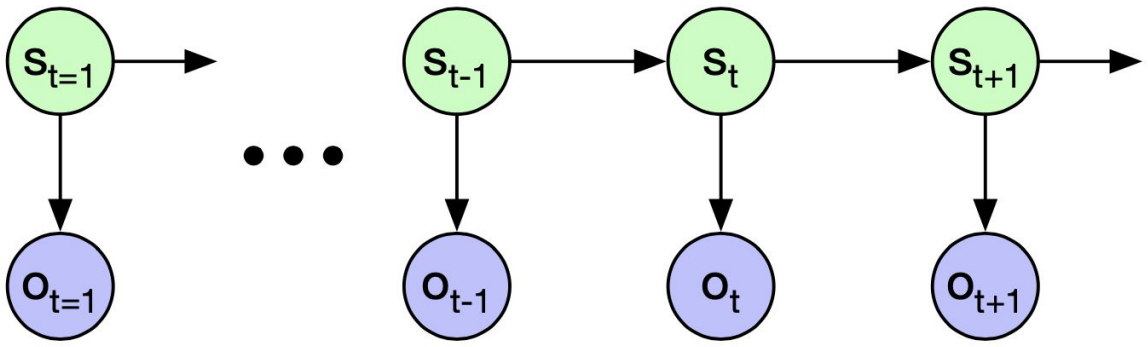
The clusters will then be input as Observations into the Hidden Markov Model (HMM) described below.

2. HMM model

2.1 Reason

The reason for choosing HMM is because of its ability to model transition & uncertainty of a sequence - which is similar to the nature of IMU readings of a motion sequence. For this problem, I use discrete, left-to-right HMM model, since our gesture motion has been discretized and only goes from one state to another following state.

The State Transition and Observation models are shown below:



Transition Probabilities

$$A_{ij} = P(s_{t+1} = \mathcal{S}_i | s_t = \mathcal{S}_j)$$

Observation (Emission) Probabilities

$$B_{jk} = P(o_t = \mathcal{O}_k | s_t = \mathcal{S}_j)$$

Initial State Distribution

$$\pi_i = P(s_{t=1} = \mathcal{S}_i)$$

2.2 Number of States & Observations

Each State in the HMM model can be thought of as 1 sub-motion in a gesture. For example, drawing a Wave gesture requires 1 clock-wise up motion, 1 sigmoidal motion and 1 anti-clockwise up motion. Looking at the pictures of the 6 gestures, I count 14 different sub-motions. Thus, I set Number of States to be 14.

I then assume that the IMU readings will have some fluctuation / errors, and we may see 1 or 2 types of readings for each sub-motion. So I experiment with Number of Observations from 14 to 28, and settle with 20 which produces a reasonable accuracy.

2.3 Training Approach

I follow the Baum-Welsh algorithm to train the HMM parameters (A, B and π) for each gesture. Before training, the Initial State Distribution π is set to be uniform, while A and B are set to be drawn randomly from an uniform distribution. To avoid underflow, the calculation is done in log space, using the useful scipy function “logsumexp” which estimates the Log of sum of exponentials terms.

Algorithm 1: The Baum-Welch algorithm

Initialization: Θ_0, $\{O_{1:T}\}$

Looping:

for $l = 1, \ldots, l_{\max}$ **do**

1. Forward-Backward calculations:

$$\alpha_1(i) = \pi_i b_i(O_1), \ \beta_T(i) = 1,$$

$$\alpha_t(i) = \Big[\sum_{j=1}^{K} \alpha_{t-1}(j) a_{ji}\Big] b_j(O_t), \ \beta_t(i) = \sum_{j=1}^{K} a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)$$

$$\text{for } 1 \le i \le K, \ 1 \le t \le T-1$$

2. E-step:

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{\sum_{j=1}^{K} \alpha_t(j)\beta_t(j)}, \ \xi_t(i,j) = \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_t(i) a_{ij} b_j(O_{t+1}\beta_t(j))}$$

$$\text{for } 1 \le i \le K, \ 1 \le j \le K, \ 1 \le t \le T-1$$

3. M-step:

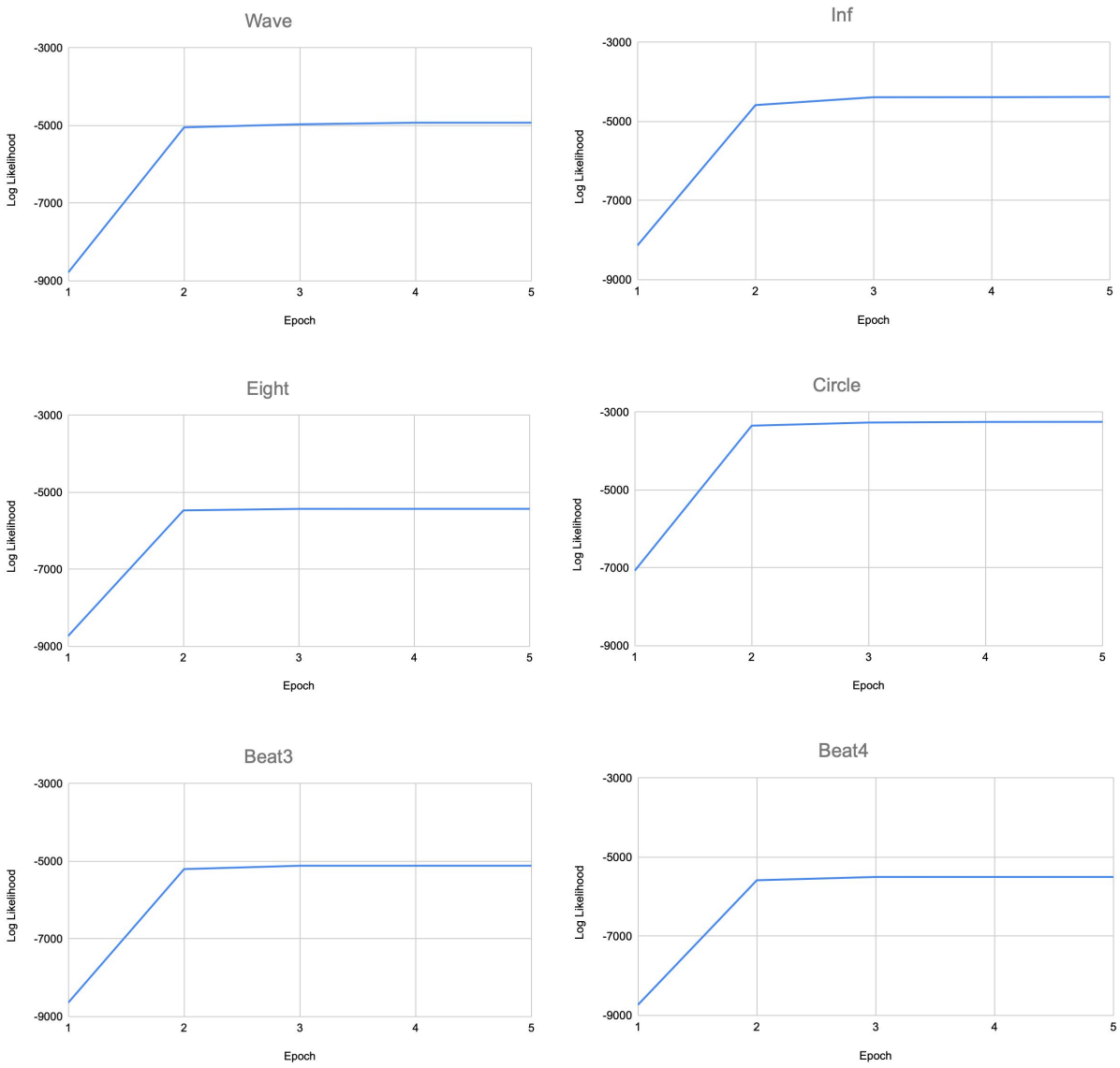
$$\pi_i = \frac{\gamma_1(i)}{\sum_{j=1}^{K} \gamma_1(j)}, \ a_{ij} = \frac{\sum_{t=1}^{T} \varepsilon_t(i,j)}{\sum_{k=1}^{K}\sum_{t=1}^{T} \varepsilon_t(i,k)}, \ w_{kd} = \frac{\sum_{t=1}^{T} \gamma_t(k,d)}{\sum_{t=1}^{T}\sum_{r=1}^{D} \gamma_t(k,r)}$$

$$\text{for } 1 \le i \le K, \ 1 \le j \le K, \ 1 \le k \le K, 1 \le d \le D$$

end

Result: $\{\Theta_l\}_{l=0}^{l_{\max}}$

The training stops when the Log Likelihood stops improving by more than 0.1. Below are the Log Likelihoods during training of different gestures:



2.4 Prediction

After training, each gesture will have an unique set of A, B and π. Then given a new observation, I can calculate the Log Likelihood that such observation belongs to each of the gestures. The gesture with highest Log Likelihood will be the predicted gesture.

IV/ Results and Discussion

The following are Test results:

File	Prediction	Log Likelihood
test1.txt	['wave', 'inf', 'eight']	-638.713328271883
test2.txt	['beat3', 'beat4', 'wave']	-1453.6719211197042
test3.txt	['inf', 'eight', 'wave']	-1308.6238860354722
test4.txt	['beat3', 'beat4', 'wave']	-1279.3916558201722
test5.txt	['circle', 'wave', 'inf']	-726.8471750814153
test6.txt	['inf', 'wave', 'eight']	-1397.7984088045139
test7.txt	['eight', 'inf', 'wave']	-1118.9487478864692
test8.txt	['beat4', 'beat3', 'wave']	-1271.0288119830504

During development, the algorithm correctly predicts most gestures except for ‘beat3’ and ‘beat4’, which are often confused with each other. I expect most errors from the Test results will come from these ‘beat’ gestures.