ESE 650 Learning in Robotics, Spring 2018
Project 3 Gesture Recognition

Hantian Liu

Abstract— **The report presents an approach for gesture recognition based on Hidden Makov Models. With angular velocity and acceleration data from onboard IMU, the raw data under each motion gesture is discretized and used to train the models. For new data, same discretization would be obtained and recognized as the motion with the highest probability according to the models.**

I. INTRODUCTION

In this project, the Hidden Markov Models was trained via Baum-Welch method, which was equivalently the EM method. Model parameters were learned with discretized training data, until the log likelihood converges. Then the models were applied to classify unknown arm motions in real-time.

II. METHODOLOGY

A. Data Pre-process

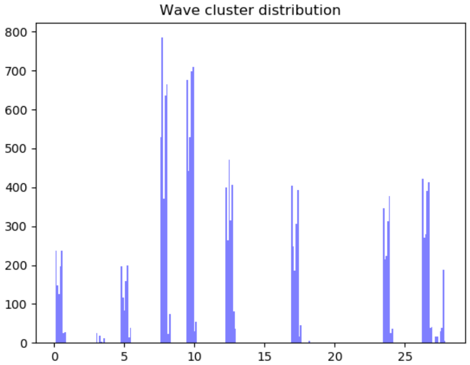
Filter and Standerdization

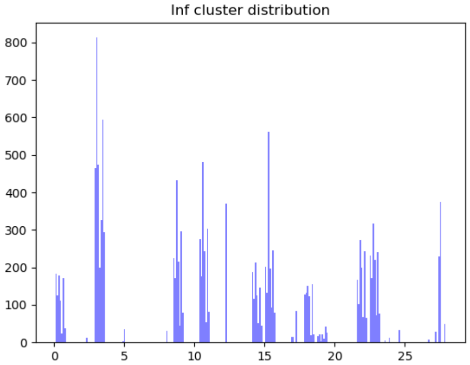
I tried with Kalman Filters to convert the raw data in acceleration and angular velocities, into Euler Angles, in representation of the orientation. However, due to the lack of ground truth orientation, it was hard to tune the parameters of the process and measurement noise. What's more, if zero-noise was chosen, where raw data was one-hundred percent trusted, the prediction accuracy would not differ much from what I could obtain from the raw data set.

Standardization was another possible method to pre-process data, where I simply unified each item in the data to have zero mean and one standard deviation. But I noted that for models with large training set, such as motion wave and infinite, the prediction confidence does not change much. While for models with small training set, it is not strong enough to prove its better performance. Therefore, it was not applied to the final training process.

Disceretization

To get discrete observation classes, I first extracted features for all data, where all the data was clustered by k means, i.e. $sklearn.cluster.KMeans$, and assigned a corresponding cluster center. Thus their labels, in replacement of the raw data, were seen as the observation classes. The observations under each model varied, as distributions shown in following figures (also shown on the next page), where the number of clusters, i.e. the total number of observation classes was set to 30. Therefore, it validated HMMs to distinguish them.





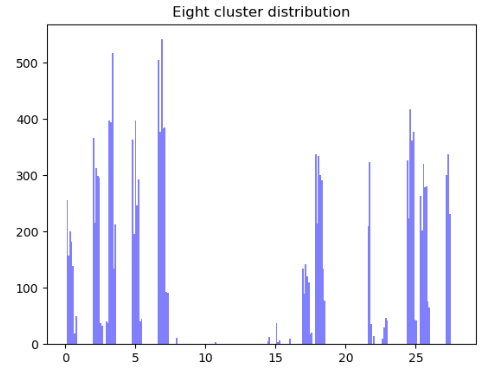
B. Hidden Markov Models

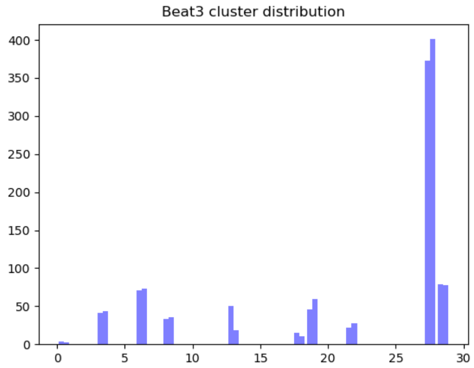
HMM is a type of stochastic signal model, following the assumption that the signal can be well characterized as a parametric random process. In our case, in oder to evaluate the probability of a sequence of observations, a best sequence of hidden states model should be determined, and then model parameters would be adjusted to work best for the observed signal.

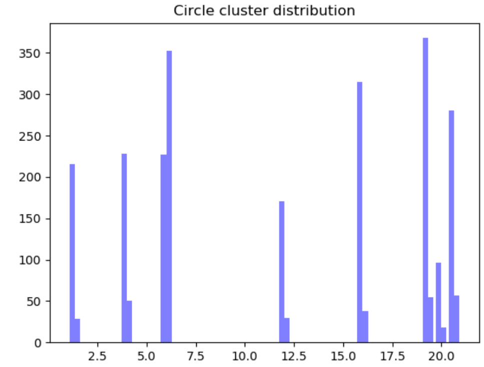
Initialization

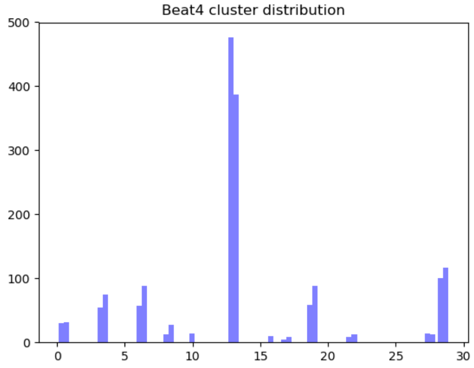
Elements of an HMM includes:

- N, the number of hidden states in the model
- M, the number of distinct observation symbols per state, here the cluster labels worked as the symbols


Eight cluster distribution


Beat3 cluster distribution


Circle cluster distribution


Beat4 cluster distribution

I started with $N = 10$, and $M = 30$. Instead of doing cross validation for different combinations of N and M, which was too slow for training process, I first tried with fixed N and different M in range (30, 90), I found that the confidence did not monotonically increase with the increase of number of states. The prediction confidence reached the peak at $M = 60$, increased by 0.025 than the lowest model at which $M = 30$, and by around 0.03 over $M = 90$. Then with $M = 60$, I tried different N in range (10,20), the confidence did not change much. Therefore, I finalized my training model with $N = 10$, $M = 60$.

- $A = a_{ij}$, the state transition probability distribution, $N \times N$ matrix

$$a_{ij} = P[q_{t+1} = S_j | q_t = S_i] \qquad 1 \leq i, j \leq N$$

Initially I assumed a left-to-right model, which could readily model signals whose properties change over time.

$$a_{ij} = 0 \qquad j > i + \Delta$$

Further, there are some properties of motion to notice. For example, circle could be realized repeatedly, where the end reconnects with the beginning. However, for beat-3 and beat-4 motions, the end could not easily transfer into the beginning. Therefore,

$$a_{N,1} \neq 0 \text{ for models except beat-3 and beat-4}$$

Also, I noticed that most of the observance sequences would consist of small number of states in a long time, i.e. the likelihood that the state stayed the same was the largest. So instead of averaging the A matrix row-wise. I put the most weight on the diagonal, and the weights on neighbors sharply decreased to zero. By taking the average of the length of the observance sequences as T, the initial A was obtained as

$$a_{i,i} = 1 - \frac{N}{T}$$

$$a_{i,i+1} = \frac{2}{3}(1 - (1 - \frac{N}{T}))$$

$$a_{i,i+2} = \frac{1}{3}(1 - (1 - \frac{N}{T}))$$

- $B = b_{ij}$, the observation symbol probability distribution in states, $M \times N$ matrix

$$b_{ij} = P[v_t = i | q_t = S_j] \qquad 1 \leq i \leq M, 1 \leq j \leq N$$

- $\pi = \pi_i$, the initial state distribution, $N \times 1$ matrix

$$\pi_i = P[q_1 = S_i] \qquad 1 \leq i \leq N$$

Both B and π matrix was initialized evenly,

$$b_{ij} = \frac{1}{M \times N} \qquad \pi_i = \frac{1}{N}$$

C. Baum-Welch Method

Since there is no optimal way of estimating the model parameters so as to maximize the probability of the observation sequecne, here local maximal was chosen using Baum-Welch method.

Forward-backward Procedure

The forward variable is the probability of the partial observation sequence until time t, and the state S_i at time t, given the model λ. Here I set it as a $T \times N$ matrix,

$$\alpha_{t,i} = P(O_1, O_2, \cdots, O_t, q_t = S_i|\lambda)$$

1) Initialization

$$\alpha_{1,i} = \pi_i b_{O_1,i} \qquad 1 \leq i \leq N$$

2) Induction

$$\alpha_{t+1,j} = [\sum_{i=1}^{N} \alpha_{t,i} a_{i,j}] b_{O_{t+1},j}$$

$$1 \leq t \leq T-1 \quad 1 \leq j \leq N$$

3) Termination

$$P(O|\lambda) = \sum_{i=1}^{N} \alpha_{T,i}$$

Similarly, the backward variable is the probability of the partial observation sequence from time $t+1$ to the end, and the state S_i at time t, given the model λ, also a $T \times N$ matrix,

$$\beta_{t,i} = P(O_{t+1}, O_{t+2}, \cdots, O_T, q_t = S_i|\lambda)$$

1) Initialization

$$\beta_{T,i} = 1 \qquad 1 \leq i \leq N$$

2) Induction

$$\beta_{t,j} = \sum_{j=1}^{N} a_{i,j} b_{O_{t+1},j} \beta_{t+1,j}$$

$$t = T-1, T-2, \cdots, 1 \qquad 1 \leq i \leq N$$

Reestimation Procedure

The probability of being in state S_i at time t and state S_j at time $t+1$, given the model and the observation sequence, an $N \times N \times T$ matrix, was defined and obtained from the forward and backward variables,

$$\begin{aligned}\xi(i,j,t) &= P(q_t = S_i, q_{t+1} = S_j|O, \lambda) \\ &= \frac{\alpha_{t,i} a_{i,j} b_{O_{t+1},j} \beta_{t+1,j}}{\sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_{t,i} a_{i,j} b_{O_{t+1},j} \beta_{t+1,j}}\end{aligned}$$

Then the probability of being in state S_i at time t, given the model and the observation sequence, an $N \times T$ matrix, could be calculated as

$$\begin{aligned}\gamma(i,t) &= P(q_t = S_i|O, \lambda) \\ &= \sum_{j=1}^{N} \xi_{i,j,t}\end{aligned}$$

Based on ξ and γ, the reestimation of the parameters of the HMM was performed,

$$\overline{a_{i,j}} = \frac{\sum_{t=1}^{T-1} \xi_{i,j,t}}{\sum_{t=1}^{T-1} \gamma_{i,t}}$$

$$\overline{b_{i,j}} = \frac{\sum_{\substack{t=1 \\ s.t. O_t = i}}^{T} \gamma_{j,t}}{\sum_{t=1}^{T-1} \gamma_{j,t}}$$

During the training process, it often occurs that some rows of B matrix equals to nearly zero, which means the corresponding observation classes never appear in the obsevation sequences of the motion. In order to maintain numerical stability, I forced the extremely small values in B matrix to be a certain computable value, i.e.

$$b_{i,j} \leq 1e-12 = 1e-12$$

Multiple Observance Sequences

For multiple observance sequences, where $\mathcal{O}^k = [O_1^k, O_2^k, \cdots, O_{T_k}^k]$,

$$\mathcal{O} = [\mathcal{O}^1, \mathcal{O}^2, \cdots, \mathcal{O}^k]$$

The reesitimation formulas were modified as follows,

$$\overline{a_{i,j}} = \frac{\sum_{k=1}^{K} \sum_{t=1}^{T_k-1} \xi_{i,j,t}^k}{\sum_{k=1}^{K} \sum_{t=1}^{T_k-1} \gamma_{i,t}^k}$$

$$\overline{b_{i,j}} = \frac{\sum_{k=1}^{K} \sum_{\substack{t=1 \\ s.t. O_t = i}}^{T_k} \gamma_{j,t}^k}{\sum_{k=1}^{K} \sum_{t=1}^{T_k} \gamma_{j,t}^k}$$

Scaling

To avoid underflow during training, the computation was performed by incorporating a scaling procedure. For each $\alpha_{t,i}$, I multiplied it by a scaling coefficient c_t,

$$c_t = \frac{1}{\sum_{i=1}^{N} \alpha_{t,i}}$$

The scaled forward variable became

$$\hat{\alpha_{t,i}} = c_t \alpha_{t,i}$$

Since the magintudes of α and β are comparable, same scaling factors were used,

$$\hat{\beta_{t,i}} = c_t \beta_{t,i}$$

Note that the scaled variables had their weights canceled out in both the numerator and the denominator. Thus the reestimation parameters would still achieve at the same values as it was calculated using unscaled variables. However, it changed for computing $P(O|\lambda)$, instead of summing up α,

$$P(O|\lambda) = \frac{1}{\prod_{t=1}^{T} c_t}$$

Again, to avoid underflow, log likelihood was calculated,

$$log[P(O|\lambda)] = -\sum_{t=1}^{T} log(c_t)$$

Convergence

I vetorized my code and a single training epoch takes around 2-3 seconds for a model consisting of 7 training data files. A limit of 200 epochs is set to halt the loop. And I treated the smaller-than-000005 difference of log likelihood between previous epoch and current epoch as convergence.

Prediction

After an HMM was trained for each type of motion, new data was discretized by the same clusters, and its labels became the new observance sequence. By calculating $P(O|\lambda)$ as mentioned above,

$$\text{recognized motion} = \text{arg}_{\text{motion}} \max(P(O|\lambda_{\text{motion}}))$$

Also, confidence for the recognition was calculated as

$$\text{confidence} = \frac{\frac{1}{P_{max}} - \frac{1}{P_{second}}}{\frac{1}{P_{max}}}$$

where

$$P_{max} = \max(P(O|\lambda_{\text{motion}})$$

$$P_{second} = \max(P(O|\lambda_{\text{motion except recognized motion}})$$

The closer the confidence came to 1, the larger the difference between the maximum probability and the sencond maximum probability in models is. It refers to more confidence in recognizing the data as the motion.

III. RESULTS

For training data, the recognition results are as follows, which correctly recognized all data. Please refer to the Appendix for details.

For test data, the recognition results are as follows

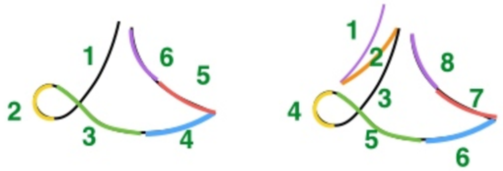
- test11.txt belongs to the motion of beat3
  with maximum log likelihood of -2555.46475819
  with confidence of 0.154403376653
- test12.txt belongs to the motion of inf
  with maximum log likelihood of -754.003132005
  with confidence of 0.878668314822
- test13.txt belongs to the motion of beat3
  with maximum log likelihood of -2263.23390835
  with confidence of 0.708816035229
- test14.txt belongs to the motion of eight
  with maximum log likelihood of -515.294575119
  with confidence of 0.928250154778
- test15.txt belongs to the motion of inf
  with maximum log likelihood of -1026.6032994
  with confidence of 0.795682683009
- test16.txt belongs to the motion of circle
  with maximum log likelihood of -355.242166315
  with confidence of 0.951419597937
- test17.txt belongs to the motion of wave
  with maximum log likelihood of -273.133531854
  with confidence of 0.953823493178
- test18.txt belongs to the motion of beat4
  with maximum log likelihood of -2180.86447713
  with confidence of 0.101695043655

IV. DISCUSSION

My HMMs achieved one-hundred percent accuracy on the training data, and seems to also have good estimation on the test data. However, there were still some problems and further improvements.

First of all, it is obvious that, both in the results of training set and test set, the confidence in recognizing beat3 and beat4 is relatively low compared to other models. In training sets, the confidence was around 0.7 compared to 0.96 for other models, while in test set it went as low as 0.15.

One possible reason would be the lack of training data set for these complex models. Among the six motions, beat motion were the most complex. Therefore, larger training data would be necessary for their higher accuracy. Also, they have really small differences between each other, namely only a short part labeling 2 in the following graph.



Therefore, if the test motion belongs to beat, the log likelihood for both beat-3 and beat-4 models would be large and causing a low confidence in the final choice.

Also, from the training process, I realized the vital significance of initialization for the model parameters in Baum-Welch procedure. Because it aimed at a local maxima, the initialized model pointed to the final model already. I did use different A matrix, which increased the cofidence of prediction much. But I did not use different B matrix. I believe B matrix also needs a wiser initialization as mentioned in the paper, so that

the accuracy would be further increased. For example, the observation sequences could be manually segmented into states. If time allows, I would try with different initialization patterns of B and analyze their influence on results.
Thirdly, feature pre-processing is another important factor for learning efficiency as well as model accuracy. If given larger training datasets, I would further discover the effcts on the filter or standardization of raw data.

APPENDIX

Results on Training data recognition

- wave01.txt belongs to the motion of wave
  with maximum log likelihood of -1748.67698587
  with confidence of 0.947958938832
- wave02.txt belongs to the motion of wave
  with maximum log likelihood of -1758.62886563
  with confidence of 0.951961680936
- wave03.txt belongs to the motion of wave
  with maximum log likelihood of -2567.46812848
  with confidence of 0.951710305928
- wave05.txt belongs to the motion of wave
  with maximum log likelihood of -2626.71679102
  with confidence of 0.945402439196
- wave07.txt belongs to the motion of wave
  with maximum log likelihood of -2682.02227351
  with confidence of 0.951760549526
- wave31.txt belongs to the motion of wave
  with maximum log likelihood of -570.196370919
  with confidence of 0.826924550617
- wave32.txt belongs to the motion of wave
  with maximum log likelihood of -282.454002894
  with confidence of 0.950025209531
- inf11.txt belongs to the motion of inf
  with maximum log likelihood of -1899.9046473
  with confidence of 0.859370838579
- inf112.txt belongs to the motion of inf
  with maximum log likelihood of -3843.62806417
  with confidence of 0.878158646032
- inf13.txt belongs to the motion of inf
  with maximum log likelihood of -2540.70106441
  with confidence of 0.917933272496
- inf16.txt belongs to the motion of inf
  with maximum log likelihood of -2281.95197272
  with confidence of 0.880548792871
- inf18.txt belongs to the motion of inf
  with maximum log likelihood of -2391.71562323
  with confidence of 0.88671449042
- inf31.txt belongs to the motion of inf
  with maximum log likelihood of -666.827007621
  with confidence of 0.898617295986
- inf32.txt belongs to the motion of inf
  with maximum log likelihood of -738.810025469
  with confidence of 0.871491959068
- eight01.txt belongs to the motion of eight
  with maximum log likelihood of -2268.68519793
  with confidence of 0.936967752495
- eight02.txt belongs to the motion of eight
  with maximum log likelihood of -3748.57264649
  with confidence of 0.920095796278
- eight04.txt belongs to the motion of eight
  with maximum log likelihood of -3668.53797115
  with confidence of 0.916474261466
- eight07.txt belongs to the motion of eight
  with maximum log likelihood of -3143.01954615
  with confidence of 0.909763205106
- eight08.txt belongs to the motion of eight
  with maximum log likelihood of -3124.37703253
  with confidence of 0.911742010719
- eight31.txt belongs to the motion of eight
  with maximum log likelihood of -528.979316626
  with confidence of 0.92025120517
- eight32.txt belongs to the motion of eight
  with maximum log likelihood of -469.771011668
  with confidence of 0.930640357956
- circle12.txt belongs to the motion of circle
  with maximum log likelihood of -1377.0999375
  with confidence of 0.96552802343
- circle13.txt belongs to the motion of circle
  with maximum log likelihood of -1403.86489367
  with confidence of 0.96146314076
- circle14.txt belongs to the motion of circle
  with maximum log likelihood of -1562.19139455
  with confidence of 0.954841510743
- circle17.txt belongs to the motion of circle
  with maximum log likelihood of -1413.94499728
  with confidence of 0.963864856764
- circle18.txt belongs to the motion of circle
  with maximum log likelihood of -1233.53652331
  with confidence of 0.964432911289
- circle31.txt belongs to the motion of circle
  with maximum log likelihood of -238.945708206
  with confidence of 0.962685309101
- circle32.txt belongs to the motion of circle
  with maximum log likelihood of -338.340927689
  with confidence of 0.958519954098
- beat3-31.txt belongs to the motion of beat3
  with maximum log likelihood of -403.985730197
  with confidence of 0.759895531409
- beat3-32.txt belongs to the motion of beat3
  with maximum log likelihood of -353.409060757
  with confidence of 0.602937199599
- beat4-32.txt belongs to the motion of beat4
  with maximum log likelihood of -479.889055417
  with confidence of 0.715417997858
- beat4-31.txt belongs to the motion of beat4
  with maximum log likelihood of -609.967778681
  with confidence of 0.713970275839

REFERENCES

[1] Lawrence R. Rabiner, A tutorial on hidden Markov models and selected applications in speech recognition, Proceedings of the IEEE Vol. 77, No. 2, pp. 257?286, 1989.