ESE 650 Learning in Robotics, Spring 2016
Project 3: Gesture Recognition with Hidden Markov Model
Due: Thursday, Feb 25th

Yiren Lu

Introduction

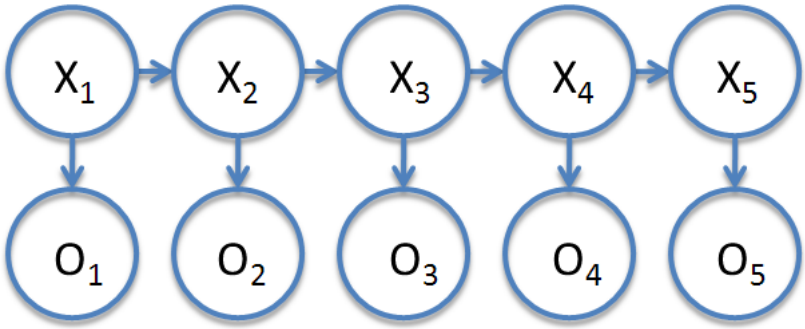
In ESE 650 learning in robotics third project, we are provided with raw data of the IMU's accelerometer's and gyroscope's data representing 6 different gestures. I used K-means unsupervised clustering algorithm to quantize the raw data. Also, I implemented Hidden Markov Model (HMM) to model the dynamic states of the gestures data sequences and built generative models to classify the gestures.

1 Data Quantization

I used kmeans algorithm to quantize the raw inputs of the 6 dimensional dataset and clustered them into a certain number of sets. I have tried different number of clusters ranging from 7 to 20 and picked the best performed models with 16 clusters.

2 Hidden Markov Model(HMM)

HMM is a standard method for modeling dynamic processes before deep learning came along. It uses dynamic Bayesian network to model the unobserved hidden states. The HMM structure is shown in the picture below:



Elements to be determined in HMM are as follows:

- N, the number of hidden states in the model.
- M, the number of distince observations symbols per state.
- $A_{N\times N}$, the transition probability distribution, where $a_{ij} = \mathbf{Pr}(q_{t+1} = S_j|q_t = S_i)$, $1 \leq i, j \leq N$.
- $B_{N\times M}$, the observation symbol probability distribution, where $b_j(k) = \mathbf{Pr}(O_t = v_k|q_t = S_j)$, $1 \leq j \leq N, 1 \leq k \leq M$
- $\pi_{N\times 1}$, the initial state distribution, where $\pi_i = \mathbf{Pr}(q_1 = S_i)$, $1 \leq i \leq N$.

The N and M are pre-determined, could be selected by cross-validation. The rest consist the model parameter $\lambda = (A, B, \pi)$.

There are generally three basic problems in HMM:

- Problem 1: Compute the probability of the observation sequence O, given a model with parameter λ, $\mathbf{Pr}(O|\lambda)$.
- Problem 2: Given the observation sequence O and the model λ, choose a corresponding state sequence Q which is optimal.
- Problem 3: Learn the model parameter $\lambda = (A, B, \pi)$, given the observation sequence O to maximize $\mathbf{Pr}(O|\lambda)$.

In this project, in order to do gesture classification, we only need to solve problem 1 and problem 3.

2.1 The forward-backward procedure

To solve problem 1, we use a dynamic algorithm called forward-backward procedure briefly stated as following:
Forward procedure

- Define $\alpha_t(i) = \mathbf{Pr}(O_1, O_2, ..., O_t, q_t = Si|\lambda)$.
- Initialize $\alpha_1(i) = \pi_i b_i(O_1)$, $1 \leq i \leq N$.
- Induction $\alpha_{t+1}(j) = \big[\sum_{j=1}^{N} \alpha_t(i)a_{ij}\big]b_j(O_{t+1})$, $1 \leq t \leq T-1, 1 \leq j \leq N$. (T is the length of the observation)

Backward procedure

- Define $\beta_t(i) = \mathbf{Pr}(O_{t+1}, O_{t+2}, ..., O_T, q_t = Si|\lambda)$.
- Initialize $\beta_T(i) = 1$, $1 \leq i \leq N$.
- Induction $\beta_t(j) = \sum_{j=1}^{N} a_{ij}b_j(O_{t+1})\beta_{i+1}(j)$, $t = T-1, T-2, .., 1, 1 \leq i \leq N$.

Outputs

- $\mathbf{Pr}(O|\lambda) = \sum_{t,i} \alpha_t(i)\beta_i(O_t)$

The forward-backward dynamic programming algorithm achieves time complexity of $O(N^2T)$ in comparison with the brutal force approach's $O(TN^T)$.

2.2 Baum-Welch method for training

To solve problem 3, we used Baum-Welch method, which is equivalent to EM algorithm in HMM.

Declaration

- Define $\xi_t(i,j) = \mathbf{Pr}(q_t = S_i, q_{t+1} = S_j|O, \lambda)$.
- Define $\gamma_t(i) = \mathbf{Pr}(q_t = S_i|O, \lambda)$.

E-step

- $\xi_t(i,j) = \frac{\alpha_t(i)a_{ij}b_j(O_{t+1})\beta_{t+1}(j)}{P(O|\lambda)}$.
- $\gamma_t(i) = \sum_{j=1}^{N} \xi_t(i,j)$.

M-step

- $\bar{\pi} = \gamma_1(i)$.
- $\bar{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)}$.
- $\bar{b}_j(k) = \frac{\sum_{t=1,s.t.O_t=v_k}^{T} \gamma_t(j)}{\sum_{t}^{T} \gamma_t(j)}$.

Iterative execute the above algorithm until it converges to a local optima.

2.3 Scaling to prevent underflow

HMM tends to underflow due to limited numerical precision of machine. We can handle this issue by scaling α and β.

$$\hat{\alpha}_t(i) = \frac{\alpha_t(i)}{\sum_{j=1}^{N} \alpha_t(j)}$$
$$\hat{\beta}_t(i) = \frac{\beta_t(i)}{\sum_{j=1}^{N} \beta_t(j)}$$

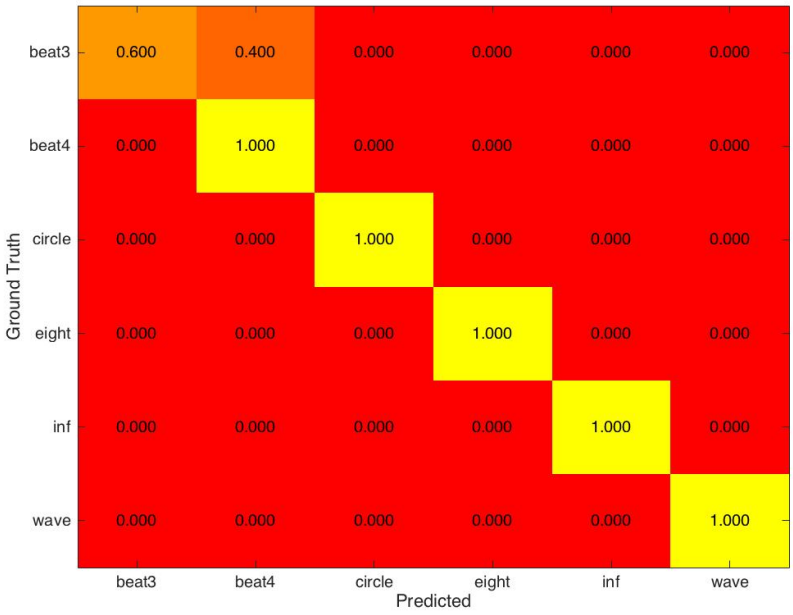
3 Training and Testing

3.1 Model selectoin and parameter initialization

I used $N = 10$, $M = 16$ for my submission, I determine N and M simply by trial and error. For the model parameter $\lambda = (A, B, \pi)$, I initialized them with random number and normalized each state.

3.2 Training set and test set

In the dataset, for each gesture, we are provided with 5 multiple gestures sequences, and 1 single gesture sequence. To train my models, *I only used one sequence of each gesture*. And test it on the remaining 4 multiple sequences and 1 single sequence.
The following are the confusion matrix on the rest of multiple gestures, 93.33%(28/30) of accuracy:



Here are the results over the single gesture sequences, 91.67%(11/12) accuracy:

#	Ground Truth	Prediction	Top Probability
1	beat3	beat4	0.4769
2	beat3	beat3	0.5290
3	beat4	beat4	0.6761
4	beat4	beat4	0.6308
5	circle	circle	0.8652
6	circle	circle	0.8580
7	eight	eight	0.9504
8	eight	eight	0.9586
9	inf	inf	0.6857
10	inf	inf	0.6790
11	wave	wave	0.4852
12	wave	wave	0.4378

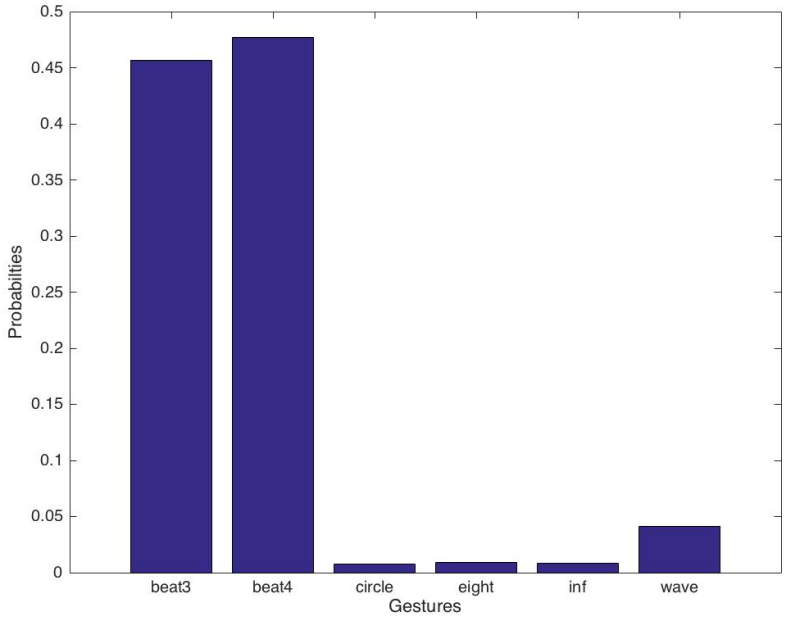
4 Calculate Probability

In order to visualize the prediction results and obtain a clear estimation of prediction confidence, I use the follow procedure to transform the raw outputs

of HMM prediction to probability.

$$
\begin{aligned}
y_i &= \frac{log(\mathbf{Pr}(O|\lambda_i))}{\sum_i log(\mathbf{Pr}(O|\lambda_i))} \\
y_{min} &= min\{y_i\} \\
\hat{y}_i &= y_{min}/y_i \\
\mathbf{Pr}(Y = i|O) &= \hat{y}_i / \sum_i \hat{y}_i
\end{aligned}
$$

Where, $\mathbf{Pr}(O|\lambda_i)$ could be computed by forward-backward procedure, λ_i is the model paramter for gesture i. $\mathbf{Pr}(Y = i|O)$ is the output probability that the gesture is i.
Below is an example visualization of the mis-classified single gesture #1. We can see that the probabilities of gesture beat3 and beat4 are quite near to each other.



5 Results and Discussion

5.1 Prediction on the validation test

Here are the prediction results along with the probability distribution. On the validation set, my model achieved *100%* accuracy.

#	beat3	beat4	circle	eight	inf	wave	Ground Truth	Prediction
1	0.4084	0.1469	0.0066	0.0120	0.0136	**0.4125**	wave	wave
2	0.1630	**0.7904**	0.0070	0.0087	0.0073	0.0236	beat4	beat4
3	0.0493	0.0260	0.0127	0.0867	**0.7992**	0.0261	inf	inf
4	**0.4971**	0.4160	0.0093	0.0125	0.0112	0.0538	beat3	beat3
5	0.0274	0.0269	**0.8807**	0.0288	0.0095	0.0268	circle	circle
6	0.0285	0.0142	0.0075	0.0489	**0.8866**	0.0143	inf	inf
7	0.0111	0.0081	0.0043	**0.9580**	0.0103	0.0082	eight	eight
8	0.1915	**0.7647**	0.0065	0.0080	0.0067	0.0226	beat4	beat4

5.2 Discussion

- Since in my project, the initialization of kmeans and parameter λ are all random, also the Baum-Welch method could only achieve the local optima, there are some probability that we get bad model. For the same configuration of parameters N and M, I tried training multiple times to obtain a relatively optimal model.

- I only used one sequence of each gesture for training with parameters $N = 10$ and $M = 16$, and it achieved 100% prediction accuracy on validation set. I believe more training data will contribute to even more accurate prediction, in which case we might also need a more complex model to reduce the training bias. Due to limited computation power and time, I have not yet tried more complex models.

6 Acknowledgement

Thanks Dr. Lee for designing and preparing this project and thanks TAs for timely responses to our questions.

References

[1] L.R.Rabiner, *A tutorial on hidden markov models and selected apllications in speech recognition*. In A. Waibel and K.-F. Lee, editors, Readings in Speech Recognition, pages 267-296. Kaufmann, San Mateo, CA, 1990.

[2] *Hidden Markov Models*, CIS520 Machine Learning, University of Pennsylvania, https://alliance.seas.upenn.edu/ cis520/dynamic/2014/wiki/index.php?n=Lectures.HMMs