Hidden Marcov Model Based Gesture Recognition

Qiong Wang
University of Pennsylvania
3401 Chestnut St., Philadelphia
qiong@seas.upenn.edu

Daniel Lee
University of Pennsylvania
3401 Chestnut St., Philadelphia
ddlee@seas.upenn.edu

Abstract

To identify variations in the signal that fit some known model given a signal varying in time, the most successful solutions tends to involve some kind of feature-based, bottom-up statistical learning, usually Hidden Markov Models (HMM), since stochastic model is appropriate for non-stationary stochastic sequences. In this paper, one Hidden Markov Model (HMM) algorithm was developed to train the data from the Inertial Measurement Unit (IMU) to recognize the gesture.

1. Introduction

This project was accomplished for coursework in ESE 650: Learning in Robotics at University of Pennsylvania. The objective is to implement one recognition algorithm to predict the gesture information based on the data provided by Inertial Measurement Unit (IMU). As one way to build sequence stochastic model, Hidden Markov Model was used to train the data and to estimate new test datasets here.

A Hidden Markov Model (HMM) is a type of stochastic model appropriate for non-stationary stochastic sequences, with statistical properties that undergo distinct random transitions among a set of different stationary processes. In other words, the HMM models a sequence of observations as a piecewise stationary process. Over the past years, Hidden Markov Models have been widely applied in several models like pattern, pathologies or speech recognition and so on. The HMMs are suitable for the classification from one or two dimensional signals and can be used when the information is incomplete or uncertain. To use a HMM, we need a training phase and a test phase. For the training stage in this project, the Baum-Welch algorithm was implemented to estimate the parameters (Π, A, B) for the HMM. This method is based on the maximum likelihood criterion in learning.

The tracking information data were provided from the accelerometer and gyroscope in the IMU as in last project. In this project, a HMM algorithm was implemented to recognize the gesture given the data from IMU. The algorithm, the results and even the recognition precision table are presented in details below.

2. Clustering and Labelling

At an input, the training data were taken from an IMU at different time stamps. In this project, there is no complex preprocessing of the data, but we have to discretize the data so as to label them in numerical space [1]. To reduce the real gesture IMU data to a workable number of discrete output symbols and states, one unsupervised clustering algorithm – K-means was implemented to cluster the 3D or 6D points of the training data for some specific gesture sequences (such as the circle) into N clusters. Then every point in the training data represents an output symbol that is closely tied to one of the N true states of the model. Then the centroids of the clusters were used to label the test data. Since the uncertainty of the test data, we expand the available output symbols to larger dimension so that they are not automatically constrained to the previous number.

3. HMM Algorithm

The typical model for a stochastic sequence of a finite number of states is called a Markov Chain or Markov Model, and when the true states of the model $S = \{s_1, s_2, \ldots, s_N\}$ are hidden that cannot be directly observed is a Hidden Markov Model (HMM). At each state an output symbol $O = \{o_1, o_2, \ldots, o_M\}$ is emitted with some probability(Log Likelihood), and one state transition matrix generated. Here, we can learn the emission B and transition A probabilities after we throw the known training data into the HMM and store these values in the emission and transition matrices. Each trained model can then be used to determine with what probability a given gesture appears in test data, where the presence or absence of the gesture is unknown, then the trained HMMs can be used to recognize gestures. HMMs are decently scale-invariant, meaning recognition should work regardless of the size of the person performing the gesture. The followings are some of the main equations[2].

1. To determine whether a sequence of observations fits the model is to compute the probability of a certain true state i at time step t given the data:

$$P_t(i) = P(s_t = i|O)$$

2. After computing this marginal probability at each round in a sequence, we can predict the hidden states and determine to what degree the data fit our model of a gesture. Then the marginal probabilities of the state were computed to find the change in emission and transition at each time step by forward-backward algorithm:

$$\alpha_{t+1} = \sum_{i=1}^{N} [\alpha_t(i) a_{ij}] b_j(o_{t+1})$$

$$\beta_{t+1} = \sum_{i=1}^{N} a_{ij} b_j [o_{t+1} \beta_{t+1}(j)]$$

3. The probability in a specific state i at time t and the probability of transition from state i to state j at time t was estimated as

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{P(O|\lambda)}$$

4. Finally, these probabilities were then encoded into the transition matrix A and the emission matrix B. After iterative optimization algorithm in the HMM Baum-Welch algorithm, we can update the parameters α and β.

4. Results

In this project, I implemented two methods: one is the HMM training by my own and the other is based on matlab build-in function `hmmtrain` and `hmmdecode`. Here are the results of several train datasets as Table 1. The recognition for the type "fish" is bad others seems fine when using my own HMM. Besides, the results from the MATLAB HMM sometimes comes out NaN. Here the NaN was treated as the detected Log Likelihood to be too small. We can also have the precision table as Table 2.

5. Conclusion

In this project, one HMM algorithm was implemented to recognize the gesture based on the data from the IMU. The two main methods worked a little bit differently sometimes. Most of them seems good, some are still not that precise. Probably, the boosting method is one solution to this scenario to combine the two training model together smartly, and it can be one of the improvement in the future work.

Gesture Type	**Recognition Rate**	**Recognized to be**
Circle	80%	Circle
Figure8	75%	Figure8
Fish	0 %	Figure8
Hammer	80%	Hammer
Pend	75%	Pend
Wave	60%	Wave

Table 1. Results of Gesture Recognition using my own HMM

Gesture Type	**Log Likehood**	**Recognized to be**
Circle	-11293	Circle
Figure8	-14902	Figure8
Fish	-16923	Fish
Hammer	-13130	Wave
Pend	-10459	Figure8
Wave	-12806	Wave

Table 2. Results of Gesture Recognition using MATLAB HMM

6. Acknowledgement

Thanks to Professor Lee, Alex and Zhuo. Thank you so much to prepare this great project.

References

[1] D. Lee. Lecture notes for ese 650: Learning in robotics, 2014.

[2] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition, 1989. IEEE.