EQ2341 Pattern Recognition and Machine Learning Project Activity Recognition

Hang Qin
hangq@kth.se

Chenting Zhang
chzha@kth.se

June 8, 2023

1 Introduction

In this project, we will implement a classifier for activity recognition based on data from the phone's sensors. We are given a set of training and testing acceleration data. We will firstly train our Hidden Markov Model parameters by using the EM algorithm. Then we will apply our trained model to test data and predict the state from three activities, standing still, walking, and running.

2 Acceleration data

We are given a set of training and testing acceleration data. Each component is a three-dimensional vector representing accelerated speed in x, y and z directions. The training and testing data are saved as "iphone_data_3.txt" and "iphone_test.txt". The data information is plotted in figure 1. The training data contains 3760 samples and the testing data contains 2742 samples.

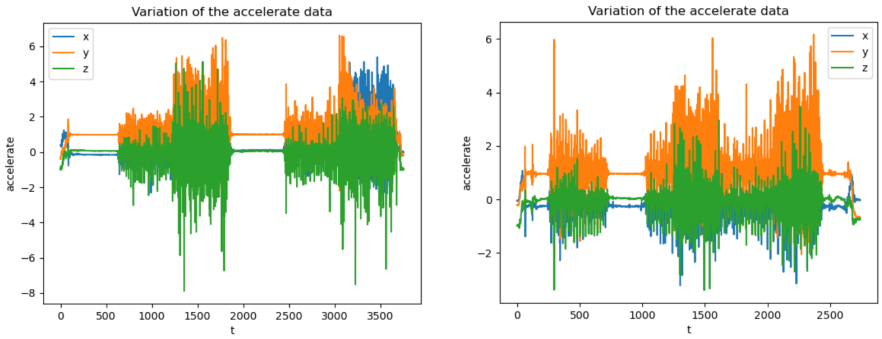

Figure 1: training samples(left) versus testing samples(right)

3 Model Training

We choose to use our already-designed HMM model $\lambda = (q, A, B)$ to fit data and do further prediction. There are three states in this model which represents walking, standing still and running respectively. In this particular problem, each output distribution corresponding each state is three-dimensional Gaussian distribution. Thus, the B matrix is denoted by the mean and covariance vector. In the model parameter initialization part, the initial values are given in figure 2:

```
# define distribution and observed data
g1 = GaussD(means=[0, 0, 0], cov=np.array([[1, 0, 0], [0, 1, 0], [0, 0, 1]]))
g2 = GaussD(means=[0, 0, 0], cov=np.array([[1, 0, 0], [0, 1, 0], [0, 0, 1]]))
g3 = GaussD(means=[0, 0, 0], cov=np.array([[1, 0, 0], [0, 1, 0], [0, 0, 1]]))
g = [g1, g2, g3]
q = np.array([1, 0, 0])
A = np.array([[0.8, 0.1, 0.1], [0.1, 0.8, 0.1], [0.1, 0.1, 0.8]])
```

Figure 2: Parameter Initialization

Then we apply the EM approach to the problem of HMM training. The algorithm derivation could be found in section 7.6[1]. As a result, the updating schemes for q, A, B are denoted in equation 7.55, 7.59 and 7.70 respectively. Then after iterating the updating schemes 20 times by using the pre-recorded training dataset, the trained model parameter values are printed in figure 3

```
updated q=
 [1.000000000000004, 0.0, 0.0]
 updated A matrix=
 [[8.54729622e-01 3.09319832e-05 1.45239446e-01]
 [1.18988094e-20 9.68307747e-01 3.16922534e-02]
 [1.91885177e-01 6.00280591e-02 7.48086764e-01]]
 updated mean vector=
 [array([ 0.48198173,  1.30585185, -0.30239864]), array([-0.01829341,  1.04006686,  0.01141591]), array([ 0.01216423,  0.68287028, -0.09508852])]
 updated Covariance matrix=
 [array([[ 1.54044622,  0.34210358, -0.2024915 ],
       [ 0.34210358,  2.21198895, -0.48270005],
       [-0.2024915 , -0.48270005,  1.86903824]]), array([[ 0.05212732, -0.00079223, -0.00849366],
       [-0.00079223,  0.04203925,  0.00033641],
       [-0.00849366,  0.00033641,  0.03131931]]), array([[0.13854149, 0.01975351, 0.00221554],
       [0.01975351, 0.28937607, 0.01836206],
       [0.00221554, 0.01836206, 0.21695133]])]
```

Figure 3: Updated model parameters

4 State prediction

After training the HMM model parameters, we predict the hidden state of each given sample in test data. We calculate the probability of each state given all the observations $P(S_t = i|\mathbf{x}, \lambda)$ and choose the state with maximum probability as the predicted state. The state prediction result is illustrated in figure 4(right).

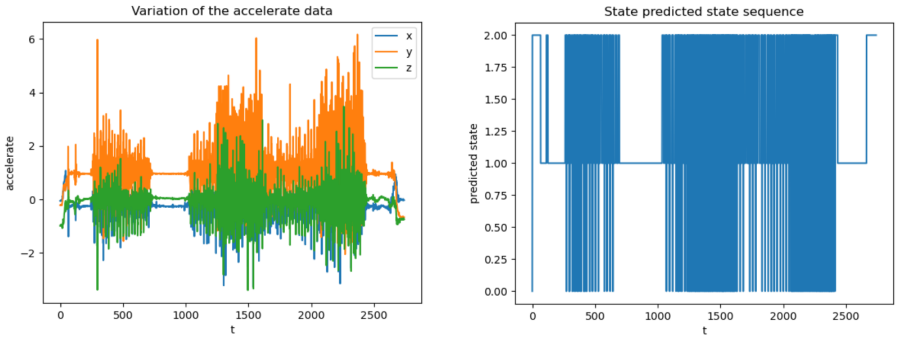

Figure 4: training samples(left) versus testing samples(right)

By comparing the two figures above, we could conclude that the state prediction in figure 4(right) aligns with the variation of acceleration data in figure 4(left), in which the state 1 represents standing still, state 0 represents waking and state 2 represents running.

5 Appendix

The repository of this activity recognition project is in EQ2341 project. The repository contains all the assignments of course EQ2341 from HMM design and implementation, music recognition, forward and backward algorithm implementation as well as this final activity recognition.

References

[1] EQ2341 PattRecKompPRINT_Marked, *Machine learning and Pattern recognition*