**CS405 Machine Learning**

**Pre Lab #8 HMM**

**Pre-Lab (25 points)**:

In this pre-lab, we provide the dataset of the China stock market. This dataset records the daily data from 1990 to the 2013. The following table explains the structure of this dataset. The hidden states of this pre-lab include “Up”,“Down” and so on.

|  |  |
| --- | --- |
| Symbol | The ID of each share |
| Open | The opening price of each share |
| Close | The closing price of each share |
| High | The highest price of each share |
| Low | The lowest price of each share |
| Volume | The daily trading volume of each share;  Volume = Amount\*Price |
| Amount | The total volume traded of each share |

Exercise 1:

1. Extract the data of “SH000001” from 1997-01-01 to 2005-12-31;

def get\_price(filename,name,start\_date,end\_date):  
 data = pd.read\_csv(filename)  
 data[["Date"]] = data[["Date"]].astype('<M8[ns]')  
 data = data[data.Name == name]  
 tmp = data[data.Date > start\_date]  
 data2 = tmp[tmp.Date < end\_date]  
 return data2

1. According to the “Open”, “Close”, “High”, “Low” and “Volume”, you can arbitratly pick some of these indicators as the features to train the hidden markov model(HMM); you can use the third party toolbox, such as “hmmlearn”;

def get\_features(filename,name,start\_date,end\_date):  
 data = get\_price(filename, name, start\_date, end\_date)  
 volume = data['Volume']  
 close = data['Close']  
 logDel = np.log(np.array(data['High'])) - np.log(np.array(data['Low']))  
 logRet\_1 = np.array(np.diff(np.log(close))) # 这个作为后面计算收益使用  
 logRet\_5 = np.log(np.array(close[5:])) - np.log(np.array(close[:-5]))  
 logVol\_5 = np.log(np.array(volume[5:])) - np.log(np.array(volume[:-5]))  
 logDel = logDel[5:]  
 logRet\_1 = logRet\_1[4:]  
 close = close[5:]  
 Date = data.Date[5:]  
 A = np.column\_stack([logDel, logRet\_5, logVol\_5])  
 return A, Date, close, logRet\_1

1. Please test your model on the data of “SH000001” from 2005-02-01 to 2007-12-31. You should give the figures including the prdictions of your model and the hidden states, as shown in the following figure (Fig.1).
2. Please set another set of prior distributions, transition probability matrix of your model, and compare the results of these 2 models, as shown in the following figure 3. And you can plot the Fig.3 referring to Fig.2.

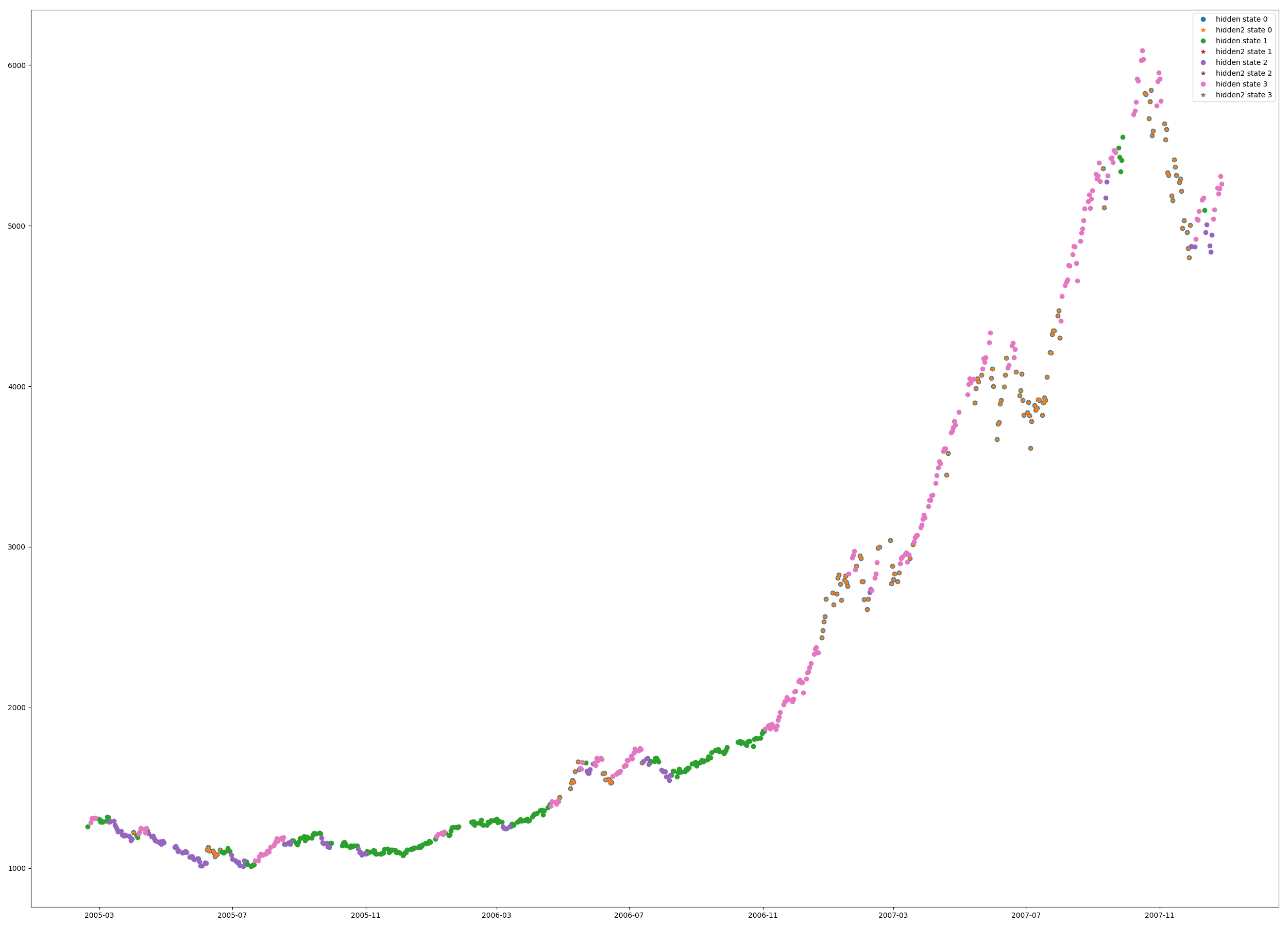


Fig.1 Prdictions

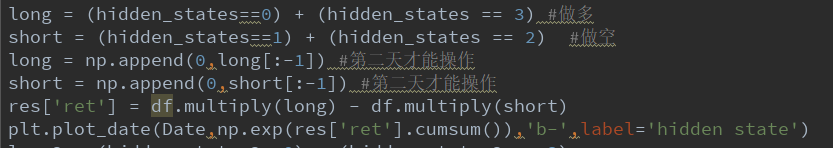


Fig.2

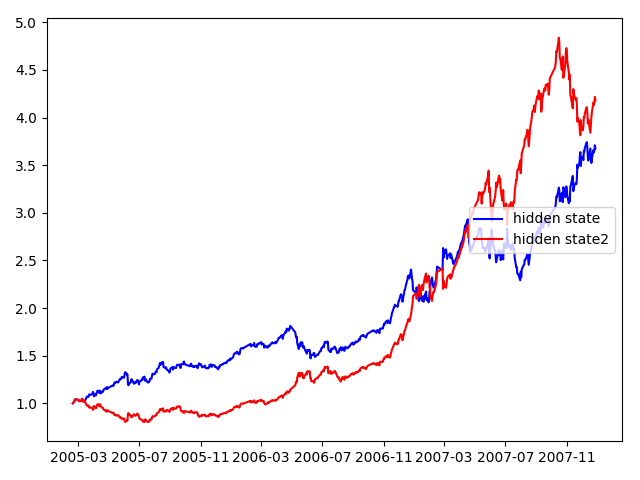


Fig.3

**Lab (75 points):**

In this lab, you should implement the HMM method by yourself, and should not use 3rd party libraries.

Exercise 2:

There are 3 coins, denoted by coin1, coin2 and coin3. The observations of 3 coins flipping are shown in the following:

seq0 = ('Heads', 'Heads', 'Heads')

seq1 = ('Heads', 'Heads', 'Tails')

seq2 = ('Heads', 'Tails', 'Heads')

seq3 = ('Heads', 'Tails', 'Tails')

seq4 = ('Tails', 'Heads', 'Heads')

seq5 = ('Tails', 'Heads', 'Tails')

seq6 = ('Tails', 'Tails', 'Heads')

seq7 = ('Tails', 'Tails', 'Tails')

Please arbitrarily select five sets of sequences as the observation sequence, denoted by *Seq.*

For example:

Seq = seq1 + seq2 + seq3 + seq4 + seq5

1. According to the following initial models (Tab-A1, B1 and C1), please use the HMM algorithm (Viterbi) to decode the order of the coin flipping (hidden state) for *Seq*, and give/compare the predictions under different priori distributions (Tab-C1). The model is (A, B, pi) where A = Transition probs, B = Observation Probs, pi = initial distribution.

Tab-A1

|  |  |  |  |
| --- | --- | --- | --- |
|  | Coin1 | Coin2 | Coin3 |
| Coin1 | 0.6 | 0.2 | 0.2 |
| Coin2 | 0.3 | 0.5 | 0.2 |
| Coin3 | 0.5 | 0.2 | 0.3 |

Tab-B1

|  |  |  |
| --- | --- | --- |
|  | Head | Tail |
| Coin1 | 0.7 | 0.3 |
| Coin1 | 0.4 | 0.6 |
| Coin1 | 0.5 | 0.5 |

Tab-C1

|  |  |  |
| --- | --- | --- |
| Coin1 | Coin2 | Coin3 |
| 0.3 | 0.4 | 0.3 |
| 0.34 | 0.33 | 0.33 |
| 0.2 | 0.4 | 0.4 |

1. Given a set of randomly initial transition probabilities (Tab-A2), initial observation probabilities (Tab-B2) and the probabilities of picking each coin (Tab-C2), please give the new model parameters after 2 iterations for *Seq*;

Tab-A2

|  |  |  |  |
| --- | --- | --- | --- |
|  | Coin1 | Coin2 | Coin3 |
| Coin1 | 0.1 | 0.3 | 0.6 |
| Coin2 | 0.5 | 0.3 | 0.2 |
| Coin3 | 0.4 | 0.3 | 0.3 |

Tab-B2

|  |  |  |
| --- | --- | --- |
|  | Head | Tail |
| Coin1 | 0.2 | 0.8 |
| Coin1 | 0.4 | 0.6 |
| Coin1 | 0.5 | 0.5 |

Tab-C2

|  |  |  |
| --- | --- | --- |
| Coin1 | Coin2 | Coin3 |
| 0.34 | 0.33 | 0.33 |