Stock Modeling using Continuous Hidden Markov Model: Experiment of CSI 300 Index

Biancheng Wang[1]

Abstract

Stock modeling has long been a difficult problem. There are two main tasks, including market regime recognition and stock price trend prediction. In this paper, we use continuous Hidden Markov Model (HMM) as a method of technical analysis to deal with these two tasks. Based on the data of CSI 300 Index, we find that continuous HMM performs better than K-means Clustering in both tasks. Continuous HMM might be useful to develop investment strategy in the Chinese financial market.

Keywords: Continuous Hidden Markov Model, K-means clustering, Stock Modeling

1. Introduction

Over the past half-decade, there has been a large amount of intensive research on the stock market. The main goal of financial analysts is to beat the market, which means to have a rate of return consistently higher than the one of market with at least the same risk of market. There are mainly two methods in investment or financial analysis, including fundamental analysis and technical analysis. For fundamental analysis, analysts make investment decision based on information from macroeconomics, industry and company analysis. They try to compare the current value of stock with the fair value to see whether it is over or under valued and they believe the future price will go toward the fair value. However, it is time-consuming and will be influenced by subjectivity. For technical analysis, this method forecasts the future price based on the past price movement. The most important part of this framework is stock market modeling.

In academia, many researchers have found that there exist several stages in the finance market. Also, many machine learning techniques have been tried to apply in the stock market prediction, including K-means Clustering, Support Vector Machine, Neural Network, and Deep Learning.

In practice, there was a famous fund among all those quantitative hedge funds, called The Medallion Fund, established by Renaissance Technologies LLC. It was also regarded as one of the blackest strategies in computational finance field. From 2001 through 2013, the fund's worst year was a 21 percent gain, after subtracting fees. Medallion reaped a 98.2 percent gain in 2008, the year the Standard & Poor's 500 Index lost 38.5 percent. At the beginning of Renaissance Technologies, three great mathematicians developed trading models and contributed a lot the long-run development of this company. Leonard Baum, the co-author of the Baum-Welch algorithm

[1] Department of Statistics, UCLA (email: wangbcbill@ucla.edu)

which will be discussed in the section 2, was one of them. Therefore, most people believe that Medallion Fund was developed based on Hidden Markov Model (HMM).

However, it is always hard to predicate the actual price in stock market precisely. Therefore, in this paper, we mainly focus on modeling the trend, or movement, of stock instead based on continuous HMM.

The remaining paper is organized as follows. In the next section, we introduce continuous HMM, including model settings and parameter estimation. Also, stock modeling process which might be used in the experiment will also be briefly described in this Section. Section 3 is dedicated to comparison between different models using a simulated data as a toy example. In Section 4, based on CSI 300 data, we try to use HMM to recognize the regime switching pattern in the stock market and predict the future movement of CSI 300 Index. Besides, we will compare the predicating power between this HMM based method and K-means Clustering based method. Finally, we conclude and give more discussion in Section 5.

2. Approach

2.1 Hidden Markov Model

The HMMs have been extensively used in the area like speech recognition, DNA sequencing, electrical signal prediction and image processing, etc., due to its strong mathematical structure and theoretical basis.

2.1.1 Model Setting

There are five basic elements in an HMM:

a. $\mathrm{S} = \{1, .., \mathrm{N}\}$ is the set of hidden states. The hidden state at time t is S_t.
b. M is the number of observed symbols in a discrete HMM while it is the number of mixture components. The observed symbol at time t is O_t.
c. Initial state distribution $\Pi = \{\pi_\mathrm{i}\}, \mathrm{i} \in \mathrm{S}$. π_i is defined as $\pi_i = \mathrm{P}(\mathrm{S}_1 = \mathrm{i})$.
d. State transition probability matrix $A = \{a_{ij}\}, i, j \in S$. a_{ij} is defined as $a_{ij} = P(S_{t+1}|S_t)$.
e. Emission probability $B = \{b_j(O_t)\}$. $b_j(O_t)$ is defined as $b_j(O_t) = P(O_t|S_t = j)$. In a continuous HMM, the function $b_j(O_t)$ is in the form of a mixture of probability density functions as: $b_j(O_t) = \sum_{\mathrm{m}=1}^{\mathrm{M}} \omega_{\mathrm{jm}} \mathrm{N}(\mathrm{O_t}, \mu_{\mathrm{jm}}, \Sigma_{\mathrm{jm}})$, where $\sum_{\mathrm{m}=1}^{\mathrm{M}} \omega_{\mathrm{jm}} = 1$.

2.1.2 Parameter Estimation

In an HMM, there are several problems we care about. Give an observation sequence, how to estimate parameters? Given the model parameter and an observation sequence, what is the underlying state sequence?

There are three algorithms dealing with these problems, including the forward-backward algorithm, the Viterbi algorithm and the Baum-Welch algorithm. The Python code for implementing these algorithms are attached.

a. Forward-backward algorithm

Forward procedure:

The forward variable $\alpha_t(i)$ is defined as $\alpha_t(i) = P(O_{1:t}, S_t = i)$.

Initialization: $\alpha_1(i) = \pi_i b_i(O_1)$ for $1 \leq i \leq N$;

Recursion: For $t = 1, \dots, T-1$, $\alpha_{t+1}(j) = b_j(O_{t+1}) \sum_{i=1}^{N} a_{ij}\, \alpha_t(i)$ for $1 \leq j \leq N$.

Backward procedure:

The backward variable $\beta_t(i)$ is defined as $\beta_t(i) = P(O_{t+1:T} | S_t = i)$.

Initialization: $\beta_T(i) = 1$ for $1 \leq i \leq N$;

Recursion: For $t = T-1. T-2, \dots, 1$, $\beta_t(i) = \sum_{j=1}^{N} a_{ij}\, b_j(O_{t+1}) \beta_{t+1}(j)$ for $1 \leq i \leq N$.

b. Viterbi algorithm

This algorithm is used to predict hidden states. We define $\delta_t(i)$ to be

$$\max_{S_1, \dots, S_{t-1}} P(O_{1:t}, S_1, \dots, S_{t-1}, S_t = i).$$

Initialization: $\delta_1(i) = \pi_i b_i(O_1)$ for $1 \leq i \leq N$;

Forward Maximization: For $t = 2, \dots, T$,

$$\delta_t(j) = \max_{1 \leq i \leq N} \left[\delta_{t-1}(i) a_{ij}\right] b_j(O_t), \text{ for } 1 \leq j \leq N.$$

$$\varphi_t(j) = \operatorname*{argmax}_{1 \leq i \leq N} \left[\delta_{t-1}(i) a_{ij}\right], \text{ for } 1 \leq j \leq N.$$

Backward tracking to find the most likely hidden state: $\hat{Z}_T = \operatorname*{argmax}_{1 \leq i \leq N} [\delta_T(i)]$; For $t = T-1. T-2, \dots, 1$, $\hat{Z}_t = \varphi_{t+1}(\hat{Z}_{t+1})$.

c. Baum-Welch algorithm

This algorithm is used to estimate parameters.

For a discrete HMM:

In the E-step, we define $\xi_t(i, j) = P(S_t = i, S_{t+1} = j | O)$ and $\gamma_i(t) = P(S_t = i | O)$. Based on the forward and backward variables, it is easy to get that

$$\xi_t(i, j) = \frac{a_{ij} b_j(O_{t+1}) \alpha_t(i) \beta_{t+1}(j)}{\sum_i \sum_j a_{ij} b_j(O_{t+1}) \alpha_t(i) \beta_{t+1}(j)} \text{ and } \gamma_t(i) = \frac{\alpha_t(i) \beta_t(i)}{\sum_{i=1}^{N} \alpha_t(i) \beta_t(i)}.$$

In the M-step, then we can get the re-estimation formula:

$$\tilde{\pi}_i = \gamma_1(i),\ \tilde{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)}, \text{ and } \tilde{b}_j(k) = \frac{\sum_{t=1:\, O_t = k}^{T} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)}$$

Then we can get the estimation by iterating between E-step and M-step until converge.

For a mixture Gaussian HMM:

In the E-step, we define $\gamma_i(j,k)$ as the probability of being in state j at time i with the kth mixture component accounting for O_t. Based on the forward and backward variables, it is easy to get that

$$\gamma_t(j,k) = \frac{\alpha_t(j)\beta_t(j)}{\sum_{j=1}^{N}\alpha_t(j)\beta_t(j)} \frac{\omega_{jk}N(O_t,\mu_{jk},\Sigma_{jk})}{\sum_{m=1}^{M}\omega_{jm}N(O_t,\mu_{jm},\Sigma_{jm})}$$

In the M-step, like the M-step in the EM algorithm when we deal with mixture Gaussian, we can get the re-estimation formula:

$$\widetilde{\omega}_{jk} = \frac{\sum_{t=1}^{T}\gamma_t(j,k)}{\sum_{t=1}^{T}\sum_{k=1}^{M}\gamma_t(j,k)}$$

$$\tilde{\mu}_{jk} = \frac{\sum_{t=1}^{T}\gamma_t(j,k)\,O_t}{\sum_{t=1}^{T}\gamma_t(j,k)}$$

$$\text{and } \tilde{\Sigma}_{jk} = \frac{\sum_{t=1}^{T}\gamma_t(j,k)(O_t-\mu_{jk})(O_t-\mu_{jk})'}{\sum_{t=1}^{T}\gamma_t(j,k)}.$$

Then we can get the estimation by iterating between E-step and M-step until converge.

In this paper, we will use continuous HMM in our analysis part.

2.2 Stock Modeling Method

There are three fundamental assumptions made in technical analysis: the market discounts everything, price moves in trends, and history tends to repeat itself. Under this framework, the movement of stock price is a dynamic process determined by the supply and demand in the stock market which is driven by some unobserved variables or even unknown variables. These unobserved variables together construct the hidden state of the market. Therefore, we think it is appropriate to use continuous HMM to model the financial market. By predicating the hidden state of each trading day, we may identify the regime switching pattern in the market.

Another task continuous HMM can do is to help predict the trend of price. The basic idea of the method in this paper is that history tends to repeat itself. Therefore, we will fit continuous HMM in the training sample first and then find the training sample which has the closest log-likelihood to each testing sample, which is like the work of Hassan R (2005). Then we can predict the trend of the testing sample by the moving direction of this training sample. Another method is to predict the state it belongs to and then predict the stock price trend by the history trend of this certain state.

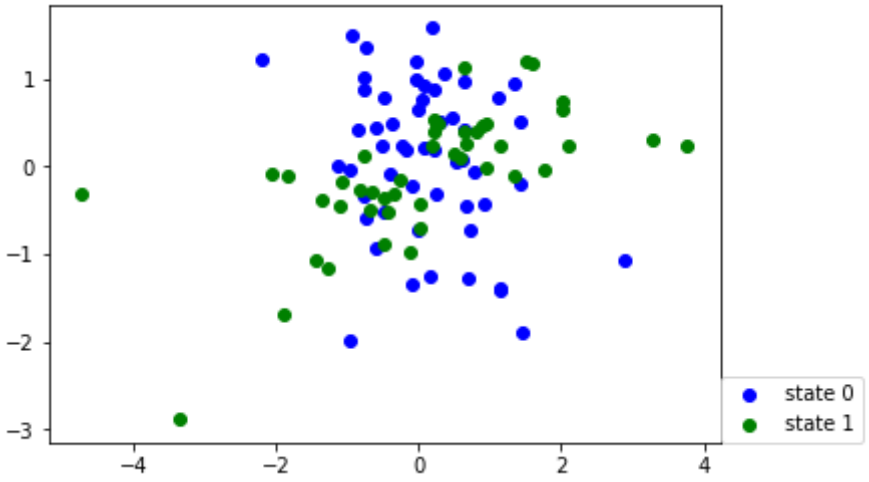
As for feature engineering, some researchers directly use the opening price, the highest price, the lowest price and the closing price as the observed variables. However, it might be not enough. Some other features, including the trading volume, will be introduced.

3. Simulation

In this section, we want to use a toy example to give an intuitive idea why K-means Clustering or other methods will not perform well especially when we have limited stock information. As discussed above, the financial market may behave like a continuous HMM. When simply using K-means clustering, the transition information within the financial time series is lost.

Therefore, we generate data based on a simple two-state two-dimensional mixture Gaussian HMM and fit this toy example with continuous HMM and K-means clustering. From Figure 1, we can find that points representing two different states are not separated, which is also the problem we might have when modeling with limited features.

Figure 1: Data generated by a two-state two-dimensional mixture Gaussian HMM



Based on this data, if we fit them with continuous HMM, we can get a 63% accuracy rate. However, for K-means Clustering, the accuracy rate is 52%, which is just a little bit more than a random guess and lower than the one we get from continuous HMM. In stock market modeling, a small increase of accuracy rate really matters as it might be enough for you to beat the market. Therefore, it might be useful to model with continuous HMM by taking the transition information into consideration while modeling.

4. Experiment

4.1 Data Description

The data we work with in this experiment comes from the NetEase website[2]. It contains the daily data of CSI 300 Index, including the closing price, the highest price, the lowest price and the stock volume, from January 4th, 2005 to May 31st, 2018, with totally 3258 days. The CSI 300 is a capitalization-weighted stock market index designed to replicate the performance of top 300 stocks

[2] http://quotes.money.163.com/stock

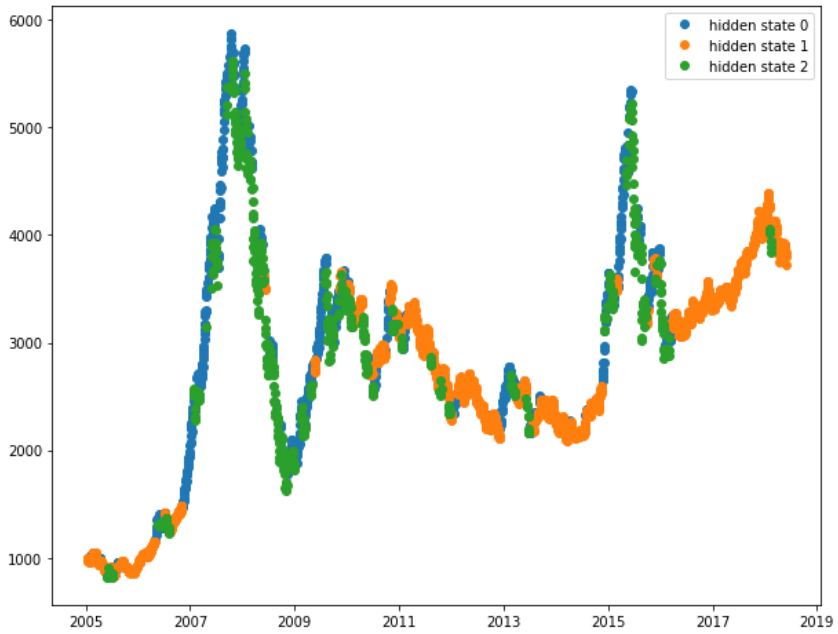
traded in the Shanghai and Shenzhen stock exchanges. The HMM has been examined to be useful in the US stock market. In this section, we try to model Chinese stock market using the HMM.

To better capture the characteristics of stock market, three features are made by transforming both price and volume information as our observation sequences in the HMM. To avoid from the weekday effect, we collect the log difference of the stock volume and the log difference of the closing price between one trading day and the day one week ago. Also, the log difference between the highest price and the lowest price has been used as the third feature. Thus, we get a 3-dimension observation variable.

4.2 Regime Recognition

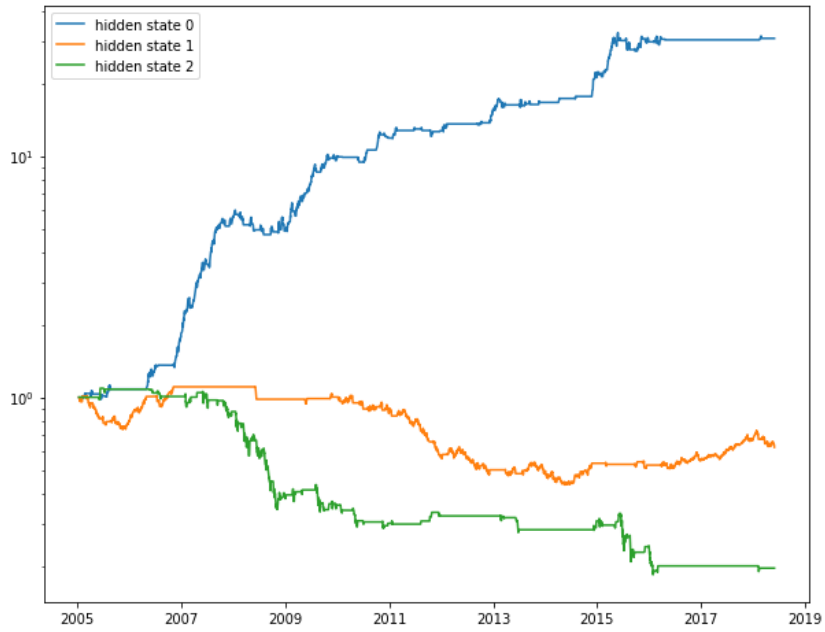
To regime-switching pattern of CSI 300 Index, we use continuous HMM to fit our data and find out the hidden state of each trading day. Here, we assume there are 3 different regimes or hidden states first. Figure 2 presents the result we get by fitting 3-state HMM. As we can see, almost all the blue points which stands for state 0 are in a bull market while a large proportion of the green points are in a bear market, e.g., the financial crisis of 2008. As for the orange points, it is hard to simply classify those points to a bull market or a bear market and it looks like a bumpy ride.

Figure 2: CSI 300 Index with States Predicated by 3-state Continuous HMM



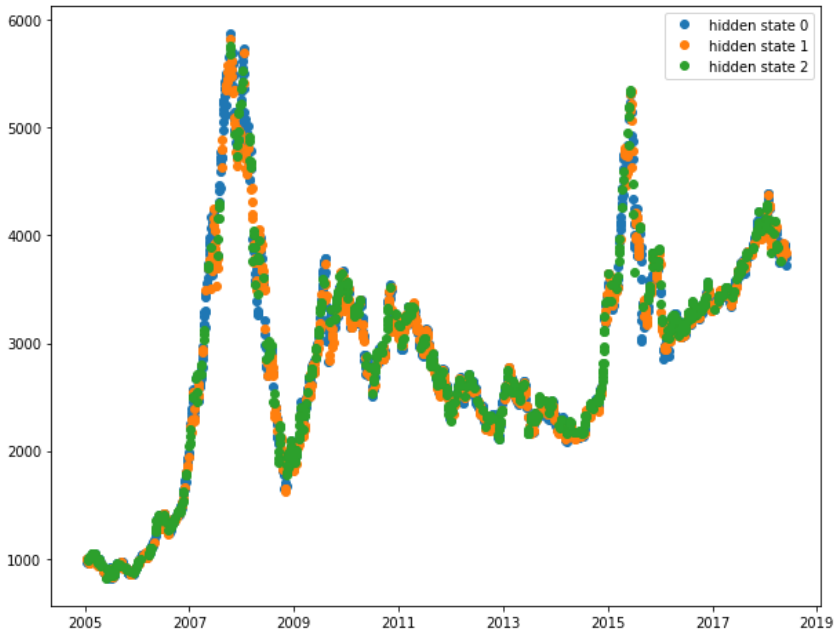
To add more credibility, for each state, we assume that we hold the stock in the next day of this state and calculate the cumulative rate of return of different states. Then we plot them in Figure 3. It seems to accord well with our analysis above, which suggests that we can expect a rise of price after state 0 and a fall after state 2. Therefore, we think this HMM has already well captured the regime switching pattern of CSI 300 Index. But we cannot regard this rate of returns as an evidence of the power of the strategy based on HMM.

Figure 3: Rates of Return in the Next Day of Different States by 3-state Continuous HMM



For comparison with other unsupervised learning methods, we also try to use K-means Clustering here. To make it more comparable, we choose a K equal to 3.

Figure 4: CSI 300 Index with States Predicated by 3-means Clustering

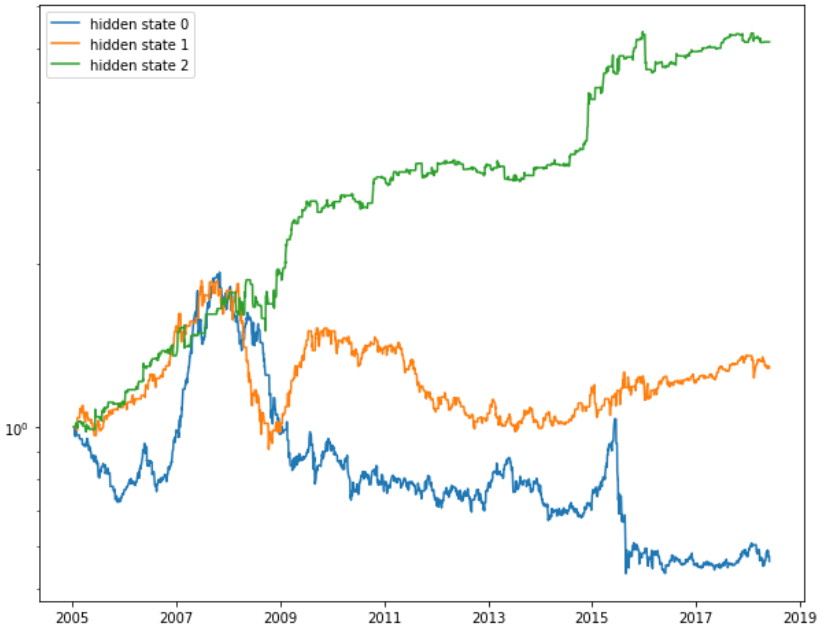


We also plot CSI 300 Index with predicated states in Figure 4 and rates of return in the next day of different states in Figure 5. From Figure 4, we can find that the distribution of the same state is more scattered than the results from continuous HMM. However, this violates the

assumption we made for the financial market, in which we believe that there are different regimes and these regimes can mutually switch after a period.

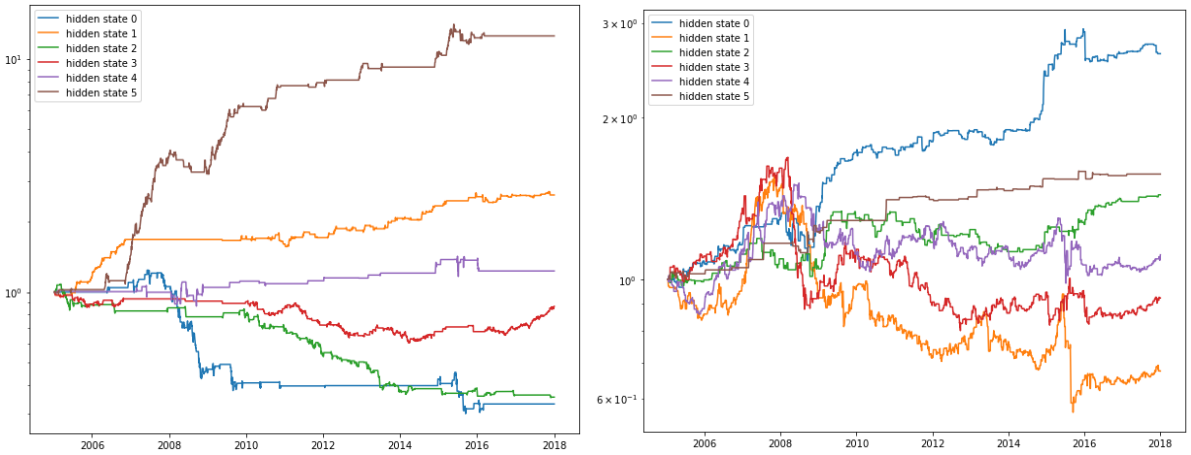
More evidence can be found in Figure 5. Both state 1 and state 2 are hard to classify to meaningful regimes as what we have done for the results from continuous HMM. Although state 0 has a positive cumulative rate of return, it seems to be much smaller than the counterpart in Figure 3. All of these have indicated that K-means Clustering does not preform as well as continuous HMM in this regime recognition task.

Figure 5: Rates of Return in the Next Day of Different States by 3-means Clustering



We also fit data with different number of states. Figure 6 shows the result from 6-state models. It seems that HMM performs better in recognize the different state and K-means clustering does a bad job especially during the financial crisis.

Figure 6: Rates of Return by 6-state continuous HMM (left) and 6-means Clustering (right)



4.3 Trend Prediction

In this part, to test whether continuous HMM is useful in predicting the price moving direction, we implement the following experiments.

We use the data of CSI 300 Index from January 4th, 2005 to December 29th, 2017 as our training data and the rest are testing data. After fitting continuous HMM for the training data and then predicate the price trend based on the history sample with the closest log-likelihood. The accuracy rates are showed in the second column of Table 1. We also make the prediction based on the history sample with the shortest Euclidian distance which is the core idea behind K-means Clustering. However, the accuracy rate is only 50.4%, which seems to be just a random guess.

Table 1: Accuracy rate of Price Trend Prediction

state	HMM (closest log-likelihood)	HMM (state prediction)	K-means Clustering (state prediction)
3	56.2%	52.3%	50.5%
4	57.2%	54.3%	53.4%
5	60.0%	54.3%	58.2%
6	53.0%	54.3%	58.2%

Instead of finding the most similar trade day in the training data, we predict the state of testing data first and then predict the trend with the trend of state in the training data. The third column and the fourth column of Table 1 show that K-means Clustering outperforms continuous HMM when assuming a large number of hidden states. Besides, as the number of states increases, K-means Clustering seems to have a stronger power. However, it will be hard to interpret the state. Therefore, it might be better to predict by finding closest log-likelihood using continuous HMM and we get the largest accuracy rate when we assume there are 5 hidden states. This rate might be enough for us to beat the market.

5. Discussion

In this research, we apply continuous HMM to model the Chinese financial market using CSI 300 Index as an example. Continuous HMM outperforms K-means Clustering in regime switching recognition and we get the best result of trend prediction by using 5-state continuous HMM. Therefore, we think HMM can help financial analyst to make decisions.

However, there are more work can be done in the future. First, in this paper, we only compare HMM with K-means Clustering. More methods, including time series methods, e.g., GARCH model, and machine learning methods, can be added into comparison. Second, more financial market can be tested as here we only choose CSI 300 Index as an example. Last but not least, it might be more reasonable to fit model for data from a shorter period of time since there still exists some short run effects which is hard to be found using our current methods. We may predict the

price trend in a dynamic way to test whether HMM is useful. Specifically, we can choose a window, e.g., 50 trading days, as the training data, predict the trend in the next day. Then continuously move the window and make the prediction.

6. References

[1] De Angelis L, Paas L J. (2013). A dynamic analysis of stock markets using a hidden Markov model. Journal of Applied Statistics, 40(8):1682-1700.

[2] Hassan R, Nath B. (2005). Stock market forecasting using hidden Markov model: a new approach. 5th International Conference on Intelligent Systems Design and Applications. Los Alamitos: IEEE,2005:192-196.

[3] Rabiner, L.R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. Proceedings of the IEEE, 77: 257-286.