Stock Price Prediction Based on Hidden Markov Model with Change Point Detection

Runze Yan(ry4jr)

Abstract— **The stock market is an important indicator which reflects economic strengths and weaknesses. Stock trading will have great returns if the economy is strongly growing, but it may have negative returns if the economy is depressing. An accurate stock price prediction is a significant key to be successful in the stock trading. Hidden Markov Model(HMM) has the advantage of modeling the correlation between effect factors in the stock market and stock price. This paper presents Hidden Markov Model(HMM) approach for predicting the daily stock price of two stocks: Apple and Amazon. This approach is different from the previous studies. First, the observation is not uni-variate(closing price) but multi-variate. The daily trading volume is also included. Second, the Akaike information criterion(AIC) and Bayesian information criterion(BIC) are used to test the HMM's performances with a number of states from two to six and select the best number of states. Third, change point detection(CPD) are used in this problem to eliminate the change point. Then, this paper compares the approach of this HMM with the approach of the Long-short Term Memory(LSTM) based on the Root Mean Square Error(RMSE) and the Mean Absolute Percentage Error(MAPE).**

Key Words: HMM, stock price, CPD

I. INTRODUCTION

Stock market is an important part of the economy of a country. The stock market plays a pivotal role in the growth of the industry and the commerce of the country that eventually affects the economy of the country to a great extent. The successful prediction of a stock's future price could yield significant profit. The stock price can be affected by a lot of factors including internal and external factors. Internal factors may contain economics and market psychology and external factors may contain politics and market psychologies. In fact, these factors remain invisible to investors and the effects of these factors are also unknown.

The stock market prediction has been one of the more active research areas in the past, given the obvious interest of a lot of major companies. Historically, various machine learning algorithms-like Artificial Neural Networks (ANN), Fuzzy Logic and Support Vector Machines (SVM), have been applied with varying degrees of success. This paper highlights the application of the Hidden Markov Model (HMM) approach. Since the stock price is a time sequence and HMMs have been successful in analyzing and predicting time depending phenomena. What's more the HMM can model the internal factors which proposed before and can affect the stock price. Particularly, we don't need to figure out what these factors are and we just assume them as the states in the hidden layer of the HMM. In this way, the HMM can be assumed as a black box and the stock data is used as the observation sequence to get the estimates of parameters of the HMM. Then, the trained model can be used to predict the stock price in the future.

Since the internal change rule can be modeled by the HMM, we also need to consider the effects of external factors. Here, the change point detection was used to find change points. change point detection(CPD) is the problem of finding abrupt changes in data when a property of the time series changes. The stock price is a kind of time series and change points in stock price series can be assumed to be caused by external factors. After finding the change points, the stock price series can be split into several intervals according to the change points. In this way, for each sequence interval, a HHM can be build to fit the data.

This paper uses two different stocks for the evaluating the HMM - Apple Inc., and Amazon Inc. A separate HMM is trained for each stock. A baseline of the project is the Long Short-term Memory (LSTM). This paper will give a comparison of the performance of HMM, HMM with change point detection and LSTM at last. The remaining paper is organized as follows: in section II, this paper lists the fundamental definition of the HMM and the key algorithms of the HMM; in section III, this paper introduces the initialization of the parameters of the HMM; in section III, this paper shows the training process and analyzes the results.

II. RELEVANT RESEARCH

Aditya Gupta and Bhuwan Dhingra used a Hidden Markov Model(HMM) to predict the next day closing price. The stocks of four different companies viz. TATA Steel, Apple Inc., IBM Corporation and Dell Inc. were taken into consideration [1].

Abhishek, Khairwa, Pratap and Prakash [2] predicted the next day stock price using feed forward architecture. The stocks of Microsoft Corporation of the year 2011 were used. The features such as open, high, low and adjacent close were considered. Next day closing price was predicted using back propagation with multi-layer feed forward network. The author also presented a literature review on application of neural network to predict stock markets.

Luo, Wu, Yan presented a novel method for the study of Shanghai Stock Exchange (SSE) index time series based on SVM regression combination model (SVR-CM) linear regression with non linear regression. Linear regression was used to extract linear features while NN algorithms were used to extract non linear features. Finally, SVM regression was used to combine all output results. SVM-CM was found to have good learning ability and forecasting capability.

Wang and Leu [4] developed a prediction system useful in forecasting mid-term price trend in Taiwan stock exchange weighted stock index (TSEWSI). The system was based on a recurrent neural network trained by using features extracted from Auto Regressive Integrated Moving Average(ARIMA). The input is same as the standard input given to ARIMA while the output approximates conditional mean predictor.

Wang, Liu and Dou [6] presented the first work which shows a service oriented approach to facilitate the incorporation of multiple factors in stock market volatility prediction. Multi kernel learning approach was used. The system was a two tier architecture. The top tier was dedicated to prepare the datasets from multiple information while the second tier was dedicated to the market volatility analysis and prediction.

Li and Liu determined network structure and prediction precision. The sample data was divided into training and testing data. The network was trained in order to fix study sample time order as much as possible. Testing sample data was used to test the trained network. The network giving satisfactory result was used for prediction. Weekly closing price from Shanghai Stock Exchange Centre was used as the dataset [8].

III. Hidden Markov Model (HMM)

A. Definition of Hidden Markov Model

The HMM is based on augmenting the Markov chain. A Markov change is a model that tells us something about the probabilities of sequences of random variables, states, each of which have relations with other sets. A Markov chain is useful when we need to compute a probability for a sequence of observable events. In many cases, however, the events we are interested in are hidden: we don't observe them directly. A hidden Markov model(HMM) allows us to talk about both observed events and hidden events that we think of as causal factors in our probabilistic model. An HMM is specified by the following components [9]:

Symbol	Meaning
$Q = q_1 q_2 ... q_N$	a set of N states
$A = a_{11} ... a_{ij} ... a_{NN}$	a transition probability matrix A, each a_{ij} representing the probability of moving from state i to state j, s.t. $\sum_{j=1}^{N} a_{ij}$ $\forall i$
$O = o_1 o_2 ... o_T$	a sequence of T observations, each one drawn from a vocabulary $V = v_1, v_2, ..., v_V$
$B = b_i(o_t)$	a sequence of observation likelihoods, also called emission probabilities, each expressing the probability of an observation o_t being generated from a state i
$\pi = \pi_1, \pi_2, ... \pi_N$	an initial probability distribution over states. pi_i is the probability that the Markov chain will start in state i. Some states j may have $pi_j = 0$, meaning that they cannot be initial states. Also, $\sum_{i=1}^{n} \pi_i = 1$

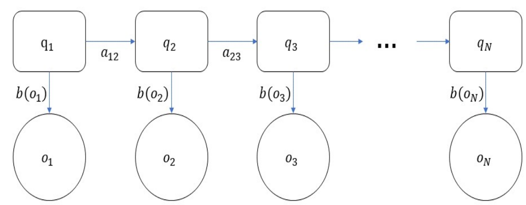

Fig. 1. Hidden Markov Model

From the graphical representation below, it is better to understand the meaning of each component in the HMM. What's more, an HMM can be assumed as a double stochastic process consisting of a hidden stochastic Markov process (of latent variables) that you cannot observe directly and another stochastic process that produces a sequence of the observation given the first process.

B. Three fundamental problems

An influential tutorial by Rabiner(1989) [10], based on tutorials by Jack Ferguson in the 1960s, introduced the idea that hidden Markov models should be characterized by three fundamental problems [9]:

- Problem 1(Likelihood): Given an HMM $\lambda = (A, B)$ and an observation sequence O, determine the likelihood $P(O|\lambda)$.
- Problem 2(Decoding): Given an observation sequence O and an HMM $\lambda = (A, B)$, discover the best hidden state sequence Q.
- Problem 3(Learning): Given an observation sequence O and the set of states in the HMM, learn the HMM parameters A and B.

C. Three according algorithms

- The first problem can be calculated by the Forward algorithm. The Forward algorithm computes the likelihood of a particular observation sequence and thus can be used in the model prediction of the HMM.
- The third problem for HMMs: learning the parameters of an HMM, that is, the A and B matrices. The input to such a learning algorithm would be an unlabeled sequence of observations O and a vocabulary of potential hidden states Q. The standard algorithm for HMM training is the Baum-Welch algorithm, a special case of the Expectation-Maximization or EM algorithm. The algorithm will train both the transition probabilities A and the emission probabilities B of the HMM. EM is an iterative algorithm, computing an initial estimate for the probabilities, then using those estimates to computing a better estimate, and so on, iteratively improving the probabilities that it learns.
- The second problem can be solved by the Viterbi algorithm and the paper only considers the algorithms

to solve the first and the last problem. Baum Welch algorithm can calibrate parameters for the HMM and forward algorithm can calculate the probability of observation to predict trending signals for stocks.

D. Change point detection

Change points are abrupt variations in time series. Such abrupt changes may represent transitions that occur between states. Detection of change points is useful in modelling and prediction of time series. That's because according to different time series' statistic property, the HMM will have different parameters. So, the combination of this theory with HMM can make the prediction more accurate. To make the stock price series more smooth, we choose to use 5 time series as a dot for us to detect the change point based on high dimensional change point theory. We choose mean, medium, second moment as three statistic property features to do the analysis [14].
The data assumption is :

$$X_i = \begin{cases} \mu_1 + \varepsilon_i, & i = 1,2,...,\lfloor n\tau \rfloor \\ \mu_2 + \varepsilon_i, & i = \lfloor n\tau \rfloor + 1,...,n \end{cases} \tag{1}$$

The hypothesis for the theory is like this:

$$H_0 : \tau = 0 \quad or \quad \tau = 1 \leftrightarrow H_1 : 0 < \tau < 1 \tag{2}$$

If H_1 holds, means no change point. If H_1 holds, there is a change point at τ. If k is the change point, $X_1,...,X_k$ and $X_{k+1},...,X_n$ are from dierent populations, the pooled sample covariance matrix is:

$$\begin{aligned} S_k = &\frac{1}{n-2}\sum_{i=1}^{k}(X_i - \bar{X}_{1k})(X_i - \bar{X}_{1k})' \\ &+ \frac{1}{n-2}\sum_{i=k+1}^{n}(X_i - \bar{X}_{(k+1)n})(X_i - \bar{X}_{(k+1)n})' \end{aligned} \tag{3}$$

We will choose to use the diagonal of S_k as D_k in the final model. And, our high-dimensional structural change model is as follows:

$$\begin{aligned} W(k) = &\frac{(n-k)^2k^2}{n^3\sqrt{p}}(\bar{X}_{1k} - \bar{X}_{(k+1)n})'D_k^{-1}(\bar{X}_{1k} - \bar{X}_{(k+1)n}) \\ &- \frac{k(n-k)\sqrt{p}}{n^2}(1+\frac{2}{n}) \\ &= \frac{1}{n\sqrt{p}}(\sum_{i=1}^{k}X_i - \frac{k}{n}\sum_{i=1}^{n}X_i)'D_k^{-1}(\sum_{i=1}^{k}X_i - \frac{k}{n}\sum_{i=1}^{n}X_i) \\ &- \frac{k(n-k)\sqrt{p}}{n^2}(1+\frac{2}{n}) \end{aligned} \tag{4}$$

IV. Initialization

A. Emission Probability

As the observations are a vector of continuous random variables, assume that the emission probability distribution is continuous. For simplicity, assume that it is a multinomial Gaussian distribution with parameters μ and σ. B is the emission matrix giving $b_j(O_t)$ the probability of observing O_t when in state j. Further, for a continuous HMM the emission probabilites are modelled as Gaussian Mixture Models (GMMs):

$$b_j(\vec{O}_t) = \sum_{m=1}^{M} c_{jm}N(\vec{O}_t, \vec{\mu_{jm}}, \Sigma_{jm}) \tag{5}$$

where:

- M is the number of Gaussian Mixture components
- c_{jm} is the weight of the m^{th} mixture component in state j.
- $\vec{\mu_{jm}}$ is the mean vector for the mth component in the j^{th} state.
- $N(\vec{O}_t, \vec{\mu_{jm}}, \Sigma_{jm})$ is the probability of observing $\vec{O}_t$ in the multi-dimensional Gaussian distribution.

B. Number of states

Choosing a number of hidden states for the HMM is critical task. We first use two common criteria: the AIC and the BIC to evaluate the performances of HMM with two different numbers of states. The two criteria are suitable for HMM because in the model training algorithm, the Baum-Welch algorithm, the EM method was used to maximize the log likelihood of the model. We limit numbers of states from two to four to keep the model simple and feasible to stock prediction. The AIC and BIC are calculated using the following formulas respectively:

$$\begin{aligned} AIC &= -2ln(L) + 2k \\ BIC &= -2ln(L) + kln(M) \end{aligned} \tag{6}$$

where L is the likelihood function for the model, M is number of observation points, and k is the number of estimated parameters in the model. In this paper, we assume that the distribution corresponding with each hidden state is Gaussian distribution, therefor the number of parameters, k is formulated as $k = N^2 + 2N - 1$, where N is the numbers of states used in the HMM.

C. Transition probability and initial probability distribution

The transition probabilities A and the initial probabilities π are assumed to be uniform across all states. To initialize the mean, variance and weights of the Gaussian mixture components we use a k-means algorithm. Each cluster found from k-means is assumed to a separate mixture component from k-means is assumed to be a separate mixture component from which the mean and variance are computed. Weights of the components are assumed to be weights of the clusters, which are divided equally between the states to obtain the initial emission probabilities.

V. Implementation

A. Dataset

This paper chooses the stock datasets of Apple and Amazon. The two datasets have the same length and are both from January 1st, 2014 to December 31th, 2018, 1259 days in total. There are limited observable features for each day, namely the opening price, the closing price, the highest price of the stock and the lowest price of of the stock for that day.

In machine learning, the entire dataset needs to be divided into two categories. The first set, the training dataset, is used to train the model. The second set, the testing dataset, is used to provide an unbiased evaluation of a final model fit on the training dataset. Separating the training dataset from the testing dataset prevents from overfitting the data into the model. So, in this case, split the dataset into two categories, training dataset for training the model and testing dataset for evaluating the model.The description of the training dataset and the testing dataset are shown in the tables below:

TABLE I
Training and Testing Dataset

Observation	Trainset	Testset
Apple	1/2/2014-5/8/2017	5/8/2017-12/31/2018
Amazon	1/2/2014-5/8/2017	5/8/2017-12/31/2018

After using the changing point detection to the close price for each day, we found two points: the 800th day and the 1100th day. Then, we split the entire dataset into three intervals as follows:

TABLE II
Three intervals after CPD

Observation	IntervalI	IntervalII	IntervalII
Apple	1/2/2014-3/6/2017	3/7/2017-5/29/2018	6/1/2018-12/31/2018
Amazon	1/2/2014-3/6/2017	3/7/2017-5/29/2018	6/1/2018-12/31/2018

B. New features

Instead of directly using the opening, closing, low, and high prices of a stock, extract the fractional changes in each of them that would be used to train our HMM. Define these parameters as follows:

$$\begin{aligned} frac_{change} &= \frac{(close - open)}{open} \\ frac_{high} &= \frac{(high - open)}{open} \\ frac_{low} &= \frac{(open - low)}{open} \\ frac_{volume} &= \frac{(close - open)}{volume} \end{aligned} \tag{7}$$

For the stock price predictor HMM, we can represent a single observation as a vector for these parameters, namely $O_t =< frac_{change}, frac_{high}, frac_{low}, frac_{volume} >$.

C. Training

The essential step in predicting the price is to train an HMM to compute the parameters from a given sequence of observations. As mentioned above, training the HMM needs the Baum-Welch algorithm which uses Expectation-Maximization (EM) to arrive at the optimal parameters for the HMM. With the help of GaussianHMM class provided by the hmmlearn package [14] in Python 3.7, the implementation of the function of the Baum-Welch algorithm can be achieved.

The number of states is an important parameters. Since the input number off states is from two to six, we get four different HMMs after training process. Based on AIC and BIC, the HMM with four states is the best candidate for Apple stock and HMM with five states is suitable for Amazon. Then, the HMM with four hidden states will be used for the stock price prediction of Apple and the HMM with five hidden states will be used for the stock price prediction of Amazon.

As for the LSTM, the number of layer is 3, the maximum number of timestaps to learn from is 64 and the number of neurons in each layer is 264.

D. Prediction

Once the HMM is trained, the closing stock price for a day can be calculated, given the opening stock price for that day and previous some d days' data. This means that if you are able to predict $frac_{change}$ for a given day, you can compute the closing price as follows:

$$clsoing = open * (1 + frac_{change}) \tag{8}$$

Thus, the problem boils down to computing the $o_{d+1} =< frac_{change}, frac_{high}, frac_{low}, frac_{volume} >$ observation vector for a day given the observation data for d days, $o_1, ..., o_d$, and the parameters of the HMM,

$$\theta = (A, B, \pi) \tag{9}$$

which is finding the value of o_{t+1} that maximizes the posterior probability of $P(o_{t+1}|o_1, ..., o_t, \theta)$:

$$\begin{aligned} o_{d+1} &= argmax_{o_{d+1}} P(o_{d+1}|o_1, ..., o_d, \theta) \\ &= argmax_{o_{d+1}} \frac{P(o_1, ..., o_d, o_{d+1}, \theta)}{P(o_1, ..., o_t, \theta)} \\ &= argmax_{o_{d+1}} P(o_1, ..., o_d, o_{d+1}|\theta) \end{aligned} \tag{10}$$

The joint probability value $P(o_1, ..., o_d, o_{d+1}|\theta)$ can be computed using the forward algorithm mentioned above.

If we assume that $frac_{change}$ is a continuous variable, the optimization of the problem would be computationally difficult. So, divide these fractional changes into some discrete values ranging between two finite variables (as stated in the following table) and find a set of fractional changes, $< frac_{change}, frac_{high}, frac_{low}, frac_{volume} >$ that would maximize the probability, $P(o_1, ..., o_{t+1}|\theta)$:

TABLE III
FEATURES DISCRETIZATION

Observation	Minimum value	Maximum value	Number of points
$frac_{high}$	0	0.1	10
$frac_{clow}$	0	0.1	10
$frac_{change}$	-0.1	0.1	20
$frac_{volume}$	-0.1	0.1	20

So, with the preceding discrete set of values, run $20 \times 10 \times 10 = 2000$ operations.

E. Results

1) HMM: Figure 2 and 3 show the actual stock values along with the prediction values for Apple and Amazon respectively. The blue line is the actual values and the red line is the prediction values.

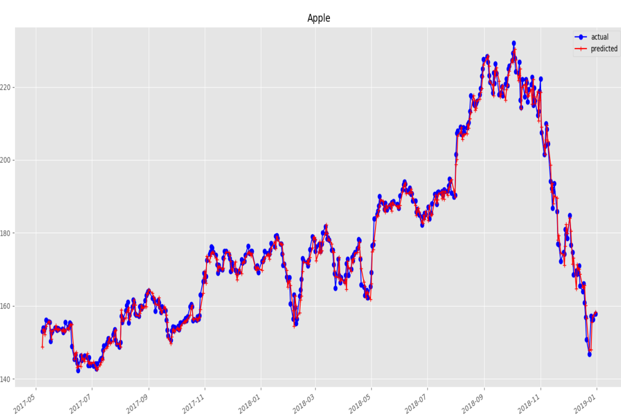

Fig. 2. Apple Stock Price Using HMM without CPD

2) HMM with CPD: Figure 4-6 show the actual stock values along the prediction values of Apple for the three intervals respectively. The blue line is the actual values and the red line is the prediction values.

F. Testing

To describe the performance of the HMM model specifically and compare the performance of the HMM and the baseline LSTM, this paper uses the Root Mean Square Error(RMSE) and the Mean Absolute Percentage Error(MAPE) to calculate the prediction error. For the HMM with CPD, we first get the RMSR and MAPE of the three intervals separately and calcualte the mean value of the three intervals.

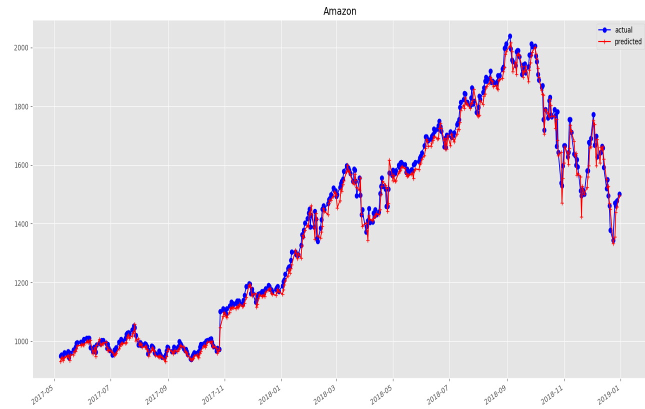

Fig. 3. Amazon Stock Price Using HMM without CPD

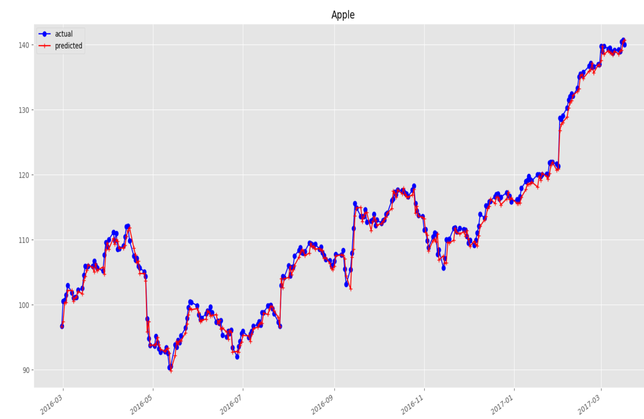

Fig. 4. Apple Stock Price Using HMM with CPD for intervalI

$$
\begin{aligned}
RSME &= \sqrt{\frac{1}{n}\sum_{i=1}^{n}(p_i - a_i)^2} \\
MAPE &= \frac{1}{n}\sum_{i=1}^{n}\frac{|p_i - a_i|}{a_i} \times 100\%
\end{aligned}
\tag{11}
$$

where a_i is the actual stock value, p_i is the predicted stock value on day i and n is the number of days for which the data is tested. LSTM is a kind of advanced neural network which suits for the time series prediction well [13]. The results of the RMSE and MAPE for the two stocks using the only HMM, the HMM with CPD and the LSTM are listed in the table below:

TABLE IV
TESTING RESULTS FOR APPLE STOCK PRICE

Metric	HMM	CPD	LSTM
RMSE	0.134	0.109	0.145
MAPE	1.1%	0.9%	1.2%

VI. CONCLUSIONS

In this paper, the Hidden Markov Model(HMM) is used in the stock price predicting and the comparison of the

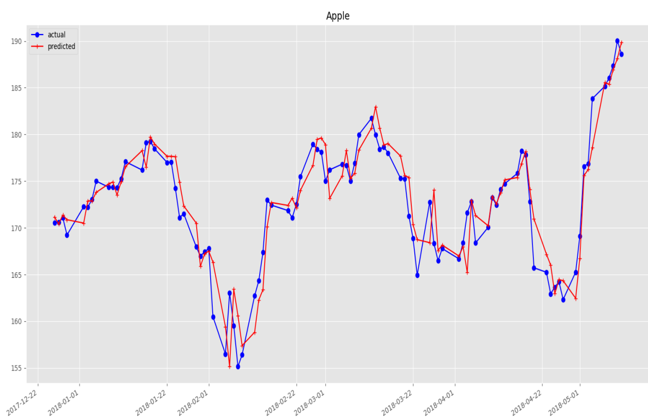

Fig. 5. Apple stock price using HMM with CPD for intervalII

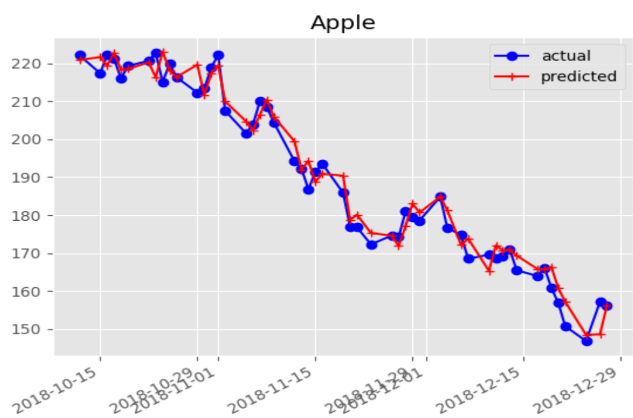

Fig. 6. Apple stock price using HMM with CPD for intervalIII

performance between the HMM and LSTM was shown in the table. What's more, the changing point detection is used to improve the performance of the HMM. According to the RSME and MAPE values from the table above, we can get that the HMM performs well compared with LSTM. In the prediction of the Apple stock, the HMM is better than the LSTM; and in the prediction of the Amazon stock, the LSTM is better. What's more, after using CPD, the average RSME and MAPE values both decrease.

VII. FUTURE WORK

Although the performance of the HMM model is good, it still has some inefficiency. For example, first, the emission probabilities are modelled by Gaussian Mixture Models (GMMs) and neural networks may have better performance. Second, from the formular, we can see that using AIC and BIC to select the number of states N can lead to high algorithm complexity. What's more, AIC and BIC may get different results.

$$O(TN^2t) \tag{12}$$

where:

- T is the number of observation sequences
- N is the number of states
- t is the number of iterations

TABLE V

TESTING RESULTS FOR AMAZON STOCK PRICE

Metric	HMM	CPD	LSTM
RMSE	1.401	0.902	0.732
MAPE	1.4%	0.9%	0.7%

REFERENCES

[1] Aditya Gupta and Bhuwan Dhingra, Stock Market Prediction Using Hidden Markov Models, IEEE 2012.

[2] Kumar Abhishek and , Anshul Khairwa, Tej Pratap and Surya Prakash, A stock market prediction model using artificial neural network, ICCCNT12, 26 th 28th July 2012, Coimbatore, India.

[3] F.Luo, J. Wu and K. Yan, A Novel Nonlinear Combination Model Based on Support Vector Machine for Stock Market Prediction, Proc. Of the 8 th World Congress on Intelligent Control and Automation, July 6-9 2010, Jinan, China.

[4] J.H.Wang and J.Y.Leu, stock market trend prediction using ARIMAbased neural network,Proc. Of IEEE c onference on neural networks, volA, pp.2 160 -2 165, 1996.

[5] Akinwale Adio T, Arogundade O.T and Adekoya Adebayo F, Translated Nigeria stock market price using artificial neural network for effective prediction. Journal of theoretical and Applied Information technology, 2009.

[6] F. Wang, L. Liu and C. Dou, Stock Market Volatility Prediction: A Service-Oriented Multi-Kernel Learning Approach, IEEE 9th International Conference on Services Computing, 2012.

[7] M. Naeini, H. Taremian and H. Hashemi, Stock Market Value Prediction Using Neural Networks, International Conference on Computer Information Systems and Industrial Management Applications (CISIM) 2010.

[8] F. Li and C. Liu, Application Study of BP Neural Network on Stock Market Prediction, 9th International Conference on Hybrid Intelligent Systems 2009.

[9] Jurafsky, Daniel.Martin, James. (2008). Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition.

[10] Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. Proceedings of the IEEE, 77(2), 257286.

[11] M. R. Hassan and B. Nath, "Stock market forecasting using hidden Markov model: a new approach," 5th International Conference on Intelligent Systems Design and Applications (ISDA'05), Warsaw, 2005, pp. 192-196.

[12] Gupta, Aditya.Dhingra, Bhuwan. (2012). Stock market prediction using Hidden Markov Models. 2012 Students Conference on Engineering and Systems, SCES 2012. 1-4. 10.1109/SCES.2012.6199099.

[13] D. M. Q. Nelson, A. C. M. Pereira and R. A. de Oliveira, "Stock market's price movement prediction with LSTM neural networks," 2017 International Joint Conference on Neural Networks (IJCNN), Anchorage, AK, 2017, pp. 1419-1426.

[14] JIN BaiSuo, PAN GuangMing,, YANG Qing ZHOU Wang. On high-dimensional change point problem .*Science China Mathematics, pages 3–10*, 2016.

[15] hmmlearn hmmlearn 0.2.1.post22+g7451142 documentation. [online] Available at: `https://hmmlearn.readthedocs.io/en/latest/`.