Physica A 517 (2019) 1–12

Contents lists available at ScienceDirect

Physica A

journal homepage: www.elsevier.com/locate/physa

# High-order Hidden Markov Model for trend prediction in financial time series

Mengqi Zhang [a,b], Xin Jiang [a,b,*], Zehua Fang [b], Yue Zeng [b], Ke Xu [a,c]

[a] *Beijing Advanced Innovation Center for Big Data and Brain Computing (BDBC), China*
[b] *LMIB & School of Mathematics and Systems Science, Beihang University, 100191, Beijing, China*
[c] *School of Computer Science and Engineering, Beihang University, 100191, Beijing, China*

## HIGHLIGHTS

- We give a detailed description of some financial indicators used in evaluating the performance of the trading strategies.
- We demonstrate how the dynamic trading strategy works in each time window and generate the trading signals according to the observation sequence.

## ARTICLE INFO

*Article history:*
Received 22 May 2018
Received in revised form 29 September 2018
Available online 1 November 2018

*Keywords:*
High-order HMM
Trend prediction
Trading algorithm

## ABSTRACT

Financial price series trend prediction is an essential problem which has been discussed extensively using tools and techniques of economic physics and machine learning. Time dependence and volatility issues in this problem have made Hidden Markov Model (HMM) a useful tool in predicting the states of stock market. In this paper, we present an approach to predict the stock market price trend based on high-order HMM. Different from the commonly used first-order HMM, short and long-term time dependence are both considered in the high order HMM. By introducing a dimension reduction method which could transform the high-dimensional state vector of high-order HMM into a single one, we present a dynamic high-order HMM trading strategy to predict and trade CSI 300 and S&P 500 stock index for the next day given historical data. In our approach, we make a statistic of the daily returns in the history to demonstrate the relationship between hidden states and the price change trend. Experiments on CSI 300 and S&P 500 index illustrate that high-order HMM has preferable ability to identify market price trend than first-order one. Thus, the high-order HMM has higher accuracy and lower risk than the first-order model in predicting the index price trend.



## 1. Introduction

Financial time series trend prediction is one of the most active research areas for economics and investments [1–4]. Specifically, the trend of stock market index price refers to the movement of the price index or the direction of fluctuation in the stock market index in the future. The prediction of price trend is a valuable issue which heavily influences the correctness of the financial participants' decision making. Leung, Daouk [5] believed that trading could be made profitable by an accurate prediction of the trend of stock index price. However, prediction of financial time series is tough due to uncertainties and nonlinear factors involved in the data. In fact, a stock market is a highly complex system, which consists of many components

* Correspondence to: School of Mathematics and Systems Science, Beihang University, China.
*E-mail address:* jiangxin@buaa.edu.cn (X. Jiang).

https://doi.org/10.1016/j.physa.2018.10.053
0378-4371/

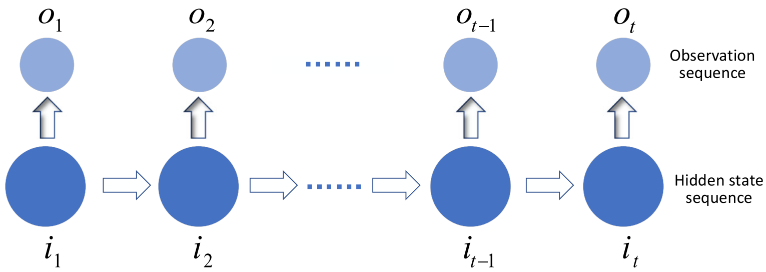



**Fig. 1.** The graph structure of HMM.

whose price move up and down without having significant patterns. Moreover, the behavior of stock markets also depends on various qualitative factors such as political, economic, natural factors and so on, which makes the stock market highly nonlinear and complex dimensionality. The complex nature of stock market challenges us on making a reliable prediction of its future trend.

During the past decades, researches have been constantly seeking for an efficient and reliable way to predict trend in financial time series [6–9]. In recent years, the machine learning methods have been applied to the areas of financial time series prediction. There are various forecasting models of financial time series using machine learning tools such as Neural Networks [10], Support Vector Machines [11], Ensemble Learning [12], Hidden Markov model(HMM) et al. Among these models, HMM is a very popular approach for modeling sequential data, such as time series, typically based on the assumption of a first-order Markov chain. In fact, Markov property plays an important role in financial time series prediction due to the short-term and long-term correlations found in empirical time series. A large amount of research of using HMM to predict financial markets have been done in recent years. Most of them consider first-order HMM based on the assumption that short-term memories exist in financial temporal dynamics. Hassan and Nash [13] made use of first-order HMM to find some day in the past which is the most similar with the current day in order to predict next day's stock price. Gupta and Dhingra [14] forecasted the next day by making a maximum a posteriori decision over all the possible stock values. Park and Lee [15] used continuous first-order HMM to forecast change direction of next day's closing price. Seethalakshmi and Krishnakumari [16] took advantage of first-order HMM to classify data in crisis and steady periods. Rebagliati, Sara and Sasso, Emanuela [17] used the HMMs to establish a set of methods to recognize the M trading patterns in finance.

The stochastic and nonstationary characteristics of financial time series make it challenging for forecasting trend in an uniform manner. Particularly, for the current stock markets, first-order HMM is strictly limited to cases where the observation at each time step is conditionally independent of the observation history and state history, given the current state [18]. However, in the field of finance, financial time series are observed to have time memories of various scales. If one only use first-order HMM, which means it only postulate first-order temporal dynamics while ignores the possibility of longer temporal dynamics in the finance time series. In this sense, one should consider longer range memories while selecting proper HMM forms. In fact, high-order hidden Markov model could provide a possible way to incorporate long memory in the dynamic of states. Different from the first-order model, high-order HMM consider the next state in the Markov chain depends on several prior states, instead of considering only one previous state.

## 2. First-order continuous Hidden Markov Model for prediction

Generally we use a continuous Hidden Markov Model to model the stock index data as a time series. An HMM is a stochastic process connecting a Markov chain which has a finite number of states with a set of random functions (observations) associated with each hidden state [15]. It can be denoted by a compact notation $\lambda = (A, B, \pi)$, where $A$ is the transition matrix, whose elements $a_{ij} = P(i_{t+1} = j | i_t = i)$ representing the probability of a transition from one state $i$ to another $j$. $B$ is the emission matrix giving the observation symbol probability $b_i(o_t)$, which is the probability of observing $o_t$ when in state $i$. That is, $b_i(o_t) = P(o_t | i_t = i)$. $\pi$ is the initial state distribution, $\pi_i = P(i_1 = i)$. In Fig. 1, we demonstrate the relationship between the hidden states and the observations.

Generally the hidden states have no practical meanings. However, in real applications, there is often some physical significance corresponding to the hidden states [19]. In practice, the corresponding hidden state sequence to an observation sequence $O = (o_1, o_2, \ldots, o_T)$ is denoted as $I = (i_1, i_2, \ldots, i_T)$, where $o_t = (o_t^1, o_t^2, \ldots, o_t^d)$, $d$ is the dimension of observation value. For a continuous HMM, the emission probability is generally modeled as Gaussian mixture distributions

$$b_i(o_t) = \sum_{k=1}^{K} c_{ik} g(o_t, \mu_{ik}, \Sigma_{ik}). \tag{1}$$

Here $K$ is the number of Gaussian mixture components, $c_{ik}$ is the mixture coefficient for the $k$th mixture in state $i$, $g(o_t, \mu_{ik}, \Sigma_{ik})$ is the multivariate Gaussian probability density function:

$$g(o_t, \mu_{ik}, \Sigma_{ik}) = \frac{1}{(\sqrt{2\pi})^d \sqrt{det(\Sigma_{ik})}} exp[-\frac{1}{2}(o_t - \mu_{ik})\Sigma_{ik}^{-1}(o_t - \mu_{ik})^T]. \tag{2}$$

**Table 1**
CSI 300 Data Format.

| Date | Open | Close | High | Low | Volume |
|---|---|---|---|---|---|
| 2005-4-08 | 984.66 | 1003.45 | 1003.7 | 979.53 | 14762500 |
| 2005-4-11 | 1003.88 | 995.42 | 1008.73 | 992.77 | 15936100 |
| ... | ... | ... | ... | ... | ... |
| 2017-7-07 | 3647.64 | 3655.93 | 3657.11 | 3631.87 | 103735497 |
| 2017-7-10 | 3647.94 | 3653.69 | 3667.85 | 3641.53 | 120591910 |

**Table 2**
Descriptive statistics of $g_t$.

| Statistics | Values |
|---|---|
| Size | 2978 |
| Min value | −5.358 |
| Max value | 4.889 |
| Mean value | 0.000 |
| Standard deviation | 1.000 |
| Skewness | −0.537 |
| Kurtosis | 3.541 |
| Skewtest $p$-value | 0.000 |
| Kurtotest $p$-value | 0.000 |
| Kolmogorov–Smirnov test $p$-value | 0.000 |

Thus, all the parameters of first-order HMM could be denoted as

$$\lambda = \{\pi, A, c_{ik}, \mu_{ik}, \Sigma_{ik}, i \in S\},$$

where $S = \{0, \ldots, N-1\}$ and $N$ is the number of hidden states. Training algorithms are used to determine the parameters$\{\pi, A, c_{ik}, \mu_{ik}, \Sigma_{ik}, i \in S\}$ by maximizing the probability of the observation sequence. Generally one may maximize the posterior likelihood function

$$P(O|\lambda) = \sum_{I} \pi_{i_1} b_{i_1}(o_1) \prod_{t=1}^{T-1} a_{i_t i_{t+1}} b_{i_{t+1}}(o_{t+1})$$

by maximizing Baum's auxiliary function [20] with Expectation–Maximization (EM) algorithm of statistics, which is known as the Baum–Welch algorithm [21–23].

In our discussion, we firstly apply the simple first-order HMM to the CSI 300 Index data, which is a capitalization-weighted stock market index designed to replicate the performance of 300 stocks traded in the Shanghai and Shenzhen stock exchanges (hereafter CSI 300). The data set is obtained from the Wind database.[1] The sample period is from April 8th 2005 to July 1st 2017. Each data point contains the daily close, open, high, low price and trading volume. The daily data format is given in Table 1.

Here the observation sequence $o_t$ is set to be the normalized daily logarithmic return series $g_t$, which is defined as

$$g_t = \frac{c_t - E(c_t)}{std(c_t)}. \tag{3}$$

Here $c_t = \log p_t - \log p_{t-1}$ where $p_t$ is the closing price of day $t$. To describe the emission probability, we first check whether $g_t$ follow the Gaussian distribution. In Table 2, we show the statistical characteristics of $g_t$ of the CSI 300. Since $g_t$ is the data after normalized, the mean value is 0 and the standard deviation is 1. Skewness is a measure of symmetry, which indicates the skewness for a normal distribution is zero. In our data set, the negative value −0.537 for the skewness indicates the distribution of data is skewed left. Kurtosis is a measure of estimating whether the distribution of data is fat-tailed compared to a normal distribution. The standard normal distribution has a kurtosis of zero, here the positive value 3.541 implies that the daily returns of the CSI 300 have leptokurtosis and fat tails. The $p$-value of kurtosis and skewness test is zero, which means the hypothesis that the kurtosis and skewness of the population is the same as that of a corresponding normal distribution is rejected. The non-Gaussianity can be confirmed by the $p$-value of approximately zeros of *Kolmogorov–Smirnov* (KS) test.

In order to inspect non-Gaussianity, we further fit a normal distribution to the empirical data and compare it to the empirical kernel density in Fig. 2 (left). It is shown the empirical kernel density has the leptokurtosis in the middle and the fat tails at both sides. Fig. 2 (right) shows a Quantile–Quantile(QQ) plot of the empirical distribution to a theoretical normal distribution. It is found that the empirical quantiles fit the normal quantiles well in the middle part, while diverge at tails, which confirms the heavy tail of the daily returns . These two figures indicates that the distribution of $g_t$ deviate significantly from the normal distribution.

1 www.wind.com.cn/en/edb.html.

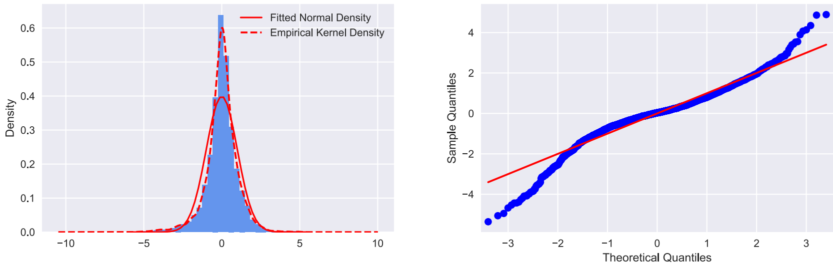



**Fig. 2.** Non-Gaussianity of $g_t$ distribution. Left panel:Empirical Kernel Density vs Fitted Normal Density. Right Panel:Quantile–Quantile Plot.

**Table 3**
Various parametric distribution fittings.

| Distribution | Log likelihood | AIC | BIC | KS test |
|---|---|---|---|---|
| Normal | −4225.598 | 8455.198 | 8467.169 | 0.00% |
| Gaussian mixture(2) | −3960.356 | 7931.618 | 7979.611 | 86.20% |
| Gaussian mixture(3) | −3957.809 | 7930.712 | 7960.707 | 12.20% |

In order to evaluate which parametric distributions is suitable for the daily returns of the CSI300. We use four evaluation indicators, log likelihood, Akaike information criterion (AIC), Bayesian information criterion (BIC), and Kolmogorov–Smirnov test $P$-value, to measure the quality of the fitting. AIC penalizes the number of parameters and the BIC considers both the number of parameters as well as the sample size. A better model has a smaller AIC and BIC.

As can be seen in Table 3, the normal distribution has the lowest log likelihood, and the highest AIC and BIC, which confirmed the aforementioned explanation of non-Gaussianity. The Gaussian mixture distribution with three components produces the highest log likelihood, lowest AIC and BIC, with a Kolmogorov–Smirnov test $P$-value of 12.20 (a *Kolmogorov–Smirnov* test cannot reject Gaussian mixture(3) at the 5% level). The study of the fitting of various parametric distributions suggests that Gaussian mixture (3) is a good candidate to capture the distributional properties of Chinese stock index returns. In this way, we model the observation sequence $g_t$ as Gauss Mixture Distribution as mentioned in Eq. (1) and Eq. (2) with $K = 3$ components. The fitting parameters are list as follows, for component1, mean is −0.022, std is 0.254 with weight 0.644, for component2, mean is −1.327, std is 1.606, with weight 0.137, for component3, mean is 0.896, std is 0.898 with weight 0.219.

Once the HMM model is trained, Viterbi algorithm [24,25] is used to determine a hidden state sequence $\{i_t\}$ which can best explains the observations. Here we propose a classification strategy to explain the corresponding market meaning of the assumed hidden states. Suppose the current day is $t$, the corresponding hidden state of the current day is $i_t$, we make a statistic of $g_{t+1}$, which is the next day's log return. By checking $g_{t+1}$ over all the hidden states $i_t$, the number of days when $g_{t+1} > 0$ and $g_{t+1} < 0$ can be achieved for each hidden states $i$. As shown in Fig. 3, we show intuitive features of the three hidden states. For hidden state 0, the number of $g_t < 0$ dominates, while for hidden state 2, the number of $g_t > 0$ dominates. For hidden state 1, the number of $g_t > 0$ and $g_t < 0$ is well-marched. Therefore, we argue that the hidden state 2 corresponds to the increasing trend of price, while hidden state 0 corresponds to the decreasing trend and hidden state 1 may indicate a fluctuation trend. For each hidden state, we can also illustrate the overall increasing or decreasing trend by calculating the cumulative logarithmic return, which is shown in Fig. 4. In this way, we show that there is a relationship between the market index prices performance and the hidden states, which can be used to interpret the possible state of market and predict price trend.

## 3. High-order Hidden Markov Model for prediction

In this section, we propose a high-order HHM based strategy for stock index time series forecasting. The suggested high-order HMM is first tested on CSI 300 Index data. The sample period is from April 8, 2005 (the launch date of the CSI300) to July 1, 2017. The normalized log return $\{g_t\}$ is chosen as the observation sequence for high-order HMM. Concretely, the hidden state transition probability contains the condition of previous $n$ states, that is

$$P(i_t|\{i_l\}_{l<t}) = P(i_t|\{i_l\}_{l=t-n}^{t-1}),\ i_t \in S. \tag{4}$$

Different from the first-order HMM, the observation not only depends on the current state but also depends on previous $m-1$ hidden states, that is

$$P(o_t|\{o_l\}_{l<t}, \{i_l\}_{l\le t}) = P(o_t|\{i_l\}_{l=t-(m-1)}^{t}). \tag{5}$$

In this way, we construct a high-order HMM of order $(n, m)$. Particularly, the hidden state sequence $\{i_t\}_{t=1}^{T}$ is an homogeneous Markov process of order $n$ over a finite state set $S$. To train the above high-order HMM from given sequence of observation, the following parameters should be optimized,

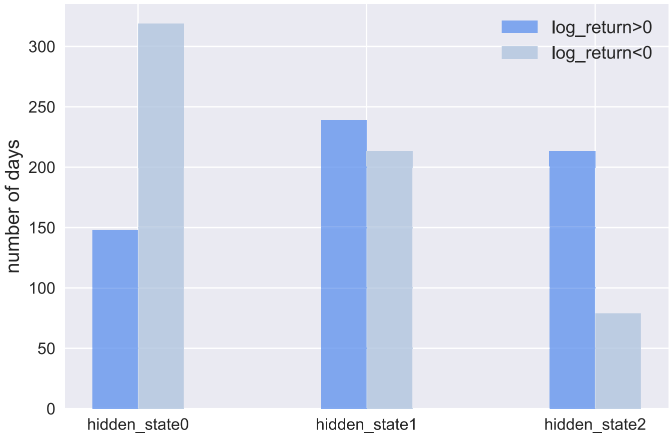



**Fig. 3.** Distribution of the number of days with logarithmic returns $g_t$ greater than 0 and less than 0 corresponding to different hidden states.

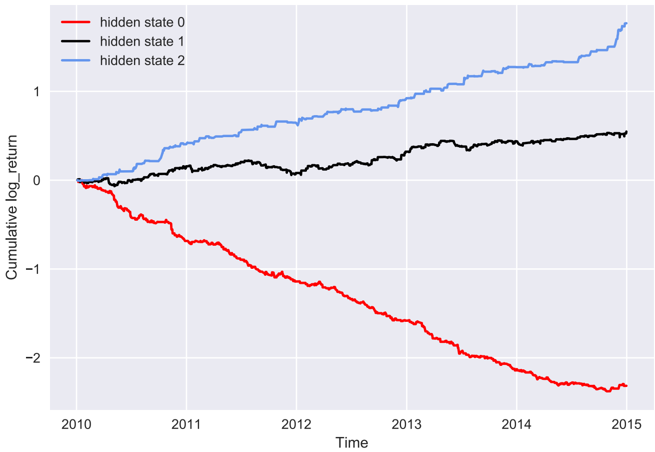



**Fig. 4.** The cumulative logarithmic return of the classified observation sequences $g_t$ corresponding to different hidden states. The red line, black line and blue line corresponds to hidden state 0, 1, 2 respectively.

- State Transition Probability,

$$a_{i_{t-n}\cdots i_t} = P(i_t|i_{t-1}, \dots, i_{t-n}).$$

- Observation Probability,

$$b_{i_{t-m+1}\cdots i_t}(o_t) = P(o_t|i_t, i_{t-1}, \dots, i_{t-m+1}).$$

- Initial state probabilities

$$\pi_{i_1\cdots i_r} = P(i_1, i_2, \dots, i_r),$$

  where $r = \max\{n, m\}$.

Thus, the parameters of high-order HMMs $\{o_t, i_t\}_{t=1}^T$ can be written as

$$\lambda = \{\{\pi_{i_1\cdots i_r}\}, \{b_{i_{t-m+1}\cdots i_t}(o_t)\}, \{a_{i_{t-n}\cdots i_t}\}\}.$$

The special case of $n = m = 1$ is degenerated to the first-order HMM. In Fig. 5, we illustrate the structure of a 2-order HMM with $n = 2$ and $m = 1$.

In order to facilitate the training process and improve the calculation efficiency of parameter estimation in high-order HMM, we introduce the state-transformation approach, which is proposed by Hadar and Messer [18], to solve the high-order HMM.

Let $r = \max\{n, m\}$, denote

$$q_t = (i_t, i_{t-1}, \dots, i_{t-r+1}),$$

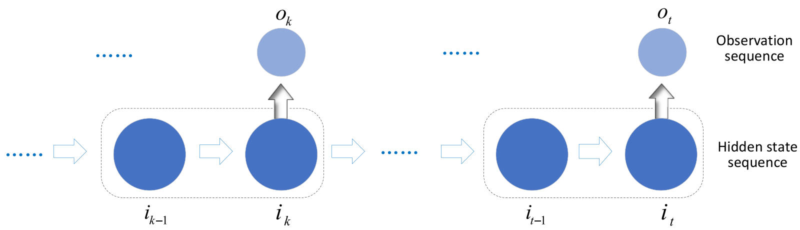



**Fig. 5.** The graph structure of the 2-order HMM.

then $\{q_t\}_{t=1}^T$ forms a new first order homogeneous Markov process. In this way, the original transition probability $a_{i_{t-n}\cdots i_t} = P(i_t|i_{t-1},\ldots,i_{t-n})$ in the high order HMM could be rewritten as

$$\tilde{a}_{q_{t-1}q_t} = P(i_t|q_{t-1}).$$

However, the original state variable $i_t$ is still involved in the form of transition probability. To reconstruct the model into a self-consistent form, we introduce a new hidden state variable $\hat{q}_t$ which is defined as:

$$\begin{aligned}\hat{q}_t &= f(q_t)\\ &= (N^{r-1},\ldots,N,1)\cdot(i_t,i_{t-1},\ldots,i_{t-r+1})^T\\ &= \sum_{l=0}^{r-1} i_{t-l}N^{r-1-l},\end{aligned} \tag{6}$$

where $f$ is a mapping of any base $N$ number to its decimal value proposed in Hadar and Messer's algorithm [18]. After some simple algebra, the new state transition probability

$$\hat{a}_{ij} = P(\hat{q}_t = j|\hat{q}_{t-1} = i).$$

Consequently, the process $\hat{q}_t$ becomes a first order homogeneous Markov process. Sequence $\{\hat{q}_t\}$ and $\{o_t\}$ constitute a first-order HMM $\{o_t,\hat{q}_t\}$ which is equivalent to the high-order HMM $\{o_t,i_t\}$. In this way, we could solve the problems of high-order HMM by applying the well known first-order HMM formulation [19].

## 4. Trading strategy based on high-order HMM

In this section, we use high-order HMM to predict CSI300 index change trend and present a trading strategy according to the predicted results. The main idea is using HMM to obtain the well-fitted hidden state $i_t$ of the current day $t$. Once the current hidden state $i_t$ is obtained, the next step is to find the days $t_s$ in the past which have the same hidden state as day $t$. For each $t_s$, we collect the price returns on the day after $t_s$, i.e., $g_{s+1}$ in the history to be the predicted trend for tomorrow index price of $t$.

Suppose that one try to predict tomorrow's index price trend, the prediction process can be explained as follows. At first, one choose a sequence $\{g_t\}_{t=1}^T$ of the index price as the input observation sequence $\{o_t\}_{t=1}^T$. $T$ is today and $o_T$ represents today's observation. $\{o_t\}_{t=1}^T$ is used to estimate the high-order HMM's parameters $\hat{\lambda}$. By using Viterbi algorithm, we determine a hidden state sequence $\{\hat{q}_t\}_{t=1}^T$ that best explains the observations $\{o_t\}_{t=1}^T$. According to Eq. (6), the accurate hidden state $i_t$ can be represented by the transformed hidden state $\hat{q}_t$ as

$$i_t = \lfloor \frac{\hat{q}_t}{N^{r-1}} \rfloor.$$

In order to use $i_T$ to predict the next day's trend, we summarize $\{i_t\}_{t=1}^{T-1}$ to find all the days $s_j$ of which the corresponding hidden state $i_{s_j} = i_T$. Next, we estimate the total return $R_{i_T}$ as the sum of all the next days' log return of day $s_j$, that is

$$R_{i_T} = \sum_j g_{s_j+1}. \tag{7}$$

Usually we consider the price trend has three different states: rise, constant or drop. Generally, the three states related to $R_{i_T}$ can be described in Table 4.

Next, an dynamic training algorithm is developed instead of the previous static training algorithm. An observation sequence $\{o_k\}_{t-W+1}^t$ is set to be the input of the high-order HMM. $\{o_k\}_{t-W+1}^t$ moves along with time $t$ such that many corresponding models $\hat{\lambda}$ can be obtained. In this sense, the length of each observation sequence $W$ is called the time window size. For each observation sequence $\{o_k\}_{t-W+1}^t$, after training and decoding, one can obtain the corresponding hidden state sequence $\{i_k\}_{t-W+1}^t$ and hidden states posterior probability $P(i_t|\{o_k\}_{k-W+1}^t)$.

**Table 4**
Trend states prediction by $R_{i_T}$.

| State | Meaning | Definition |
|---|---|---|
| 1 | Rise | $R_{i_T} > \Delta$ |
| 0 | Constant | $-\Delta \le R_{i_T} \le \Delta$ |
| −1 | Drop | $R_{i_T} < -\Delta$ |

For each time window, we use the current hidden state $i_t$ to generate a trading signal $y_{t+1}$ for the next trading day $t+1$. $y_{t+1}$ can take values of 1, −1, 0. If $y_{t+1}$ is 1, the index price trend is predicted to rise, i.e. the closing price is higher than the opening price on $t+1$ day. while $y_{t+1} = -1$, the index price trend is predicted to drop, i.e. the closing price is lower than the opening price on $t+1$ day. If $y_{t+1} = 0$, the index price is predicted to be constant.

Then, we explain how to generate trading signal according to the sequence $\{o_k\}_{t-W+1}^{t}$ in each sliding window. After training the HMM model through the observation sequence $\{o_k\}_{t-W+1}^{t}$ to obtain the model parameter $\lambda$, one can estimate the probability of the hidden state $i_t$ according to

$$i_t = argmax_i P(i_t = i|\{o_k\}_{k-W+1}^{t}, \lambda), i = 0, \ldots, N-1. \tag{8}$$

Usually one might encounter two problems, one is that the difference in posterior probability of different hidden states is relatively too small, which makes it difficult to determine the hidden state, the other is that the hidden state one obtained appears rarely in the history. In order to reduce the risk of trading, we generate signal $y_{t+1} = 0$ for these situations. We also define the cumulative return and the win rate of the hidden states in the window respectively to determine the trading signals. The cumulative return in the sliding window is

$$\sum_{k=t-W+1}^{t-1} g_{k+1} I\{i_k = i_t\}.$$

In this way, long position win rate and short selling win rate in the window are

$$\frac{\sum_{k=t-W+1}^{t-1} I\{g_k > 0\} \cdot I\{i_k = i_t\}}{\sum_{k=t-W+1}^{t-1} I\{i_k = i_t\}}$$

and

$$\frac{\sum_{k=t-W+1}^{t-1} I\{g_k < 0\} \cdot I\{i_k = i_t\}}{\sum_{k=t-W+1}^{t-1} I\{i_k = i_t\}}$$

respectively. We determine the generation of trading signals by judging the values of the above different indicators. The detailed process of the above trading signal generation algorithm is as follows:

**Step**1: Input:$\{o_k\}_{t-W+1}^{t}$, if

$$P(i_t|\{o_k\}_{t-W+1}^{t}, \lambda) > \frac{1}{N}$$

and

$$\sum_{k=t-W+1}^{t-1} I\{i_k = i_t\} > \frac{W}{3N}$$

go to **Step2**, where

$$I\{A\} = \begin{cases} 1, \text{ if } A \textit{ is true} \\ 0, \text{ if } A \textit{ is false}. \end{cases}$$

$1/N$ and $W/3N$ are predefined thresholds to ensure the probability $P(i_t|\{o_k\}_{t-W+1}^{t}, \lambda)$ of hidden state and the number of occurrences of the predicted hidden state $\sum_{k=t-W+1}^{t-1} I\{i_k = i_t\}$ in the historical data to be not too small.
Else, go to **Step3**.

**Step**2: If

$$\sum_{k=t-W+1}^{t-1} g_{k+1} I\{i_k = i_t\} > 0$$

and

$$\frac{\sum_{k=t-W+1}^{t-1} I\{g_k > 0\} \cdot I\{i_k = i_t\}}{\sum_{k=t-W+1}^{t-1} I\{i_k = i_t\}} > \omega$$

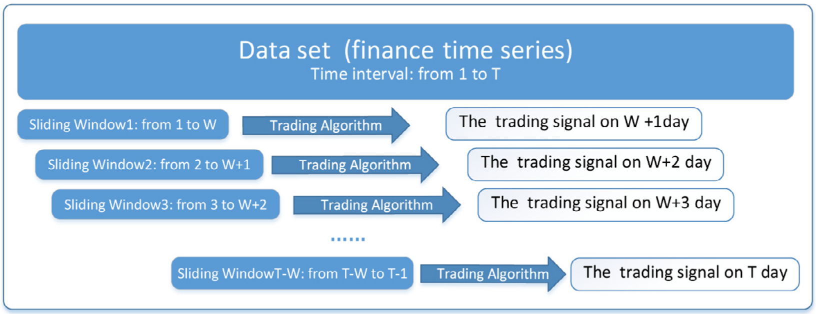



**Fig. 6.** Forecasting using dynamic trading algorithm.

then, $y_{t+1} = 1$.
Else if

$$\sum_{k=t-W+1}^{t-1} g_{k+1} I\{i_k = i_t\} < 0$$

and

$$\frac{\sum_{k=t-W+1}^{t-1} I\{g_k < 0\} \cdot I\{i_k = i_t\}}{\sum_{k=t-W+1}^{t-1} I\{i_k = i_t\}} > \mu$$

then, $y_{t+1} = -1$.
Else, go to **Step3**.

**Step**3: $y_{t+1} = 0$.

The entire dynamic training and trading process is illustrated by Fig. 6.

In this sense the trading can be executed as follows: if the trading signal $y_{t+1} = 1$, buy the stock index at the opening price of the next day, if the trading signal $y_{t+1} = -1$, sell the stock index future at the closing price of the day, if the trading signal $y_{t+1} = 0$, stay still. The following hyper-parameters is needed to be determined before proceeding the strategy:

- $W$: the size of sliding window
- $N$: the number of underlying hidden state.
- $d$: the dimension of observation state, namely, the number of time series features.
- $\omega$: the threshold that long position win rate in trading algorithm.
- $\mu$: the threshold that take short wining rate in trading algorithm.
- $n$: the order of Markov chain.

## 5. Experiment

In this section, we first introduce some indicators to evaluate the quality of the trading strategy. Then we show the results of using high-order HMM to predict and trade on CSI 300 Index and S&P 500 Index. For both data sets, the performances of the first-order HMM and the high-order HMM are compared.

### 5.1. Evaluation indicators

In order to evaluate the performance of the trading strategies, in addition to the traditional indicators such as recall and precision, we also introduce the following indicators.

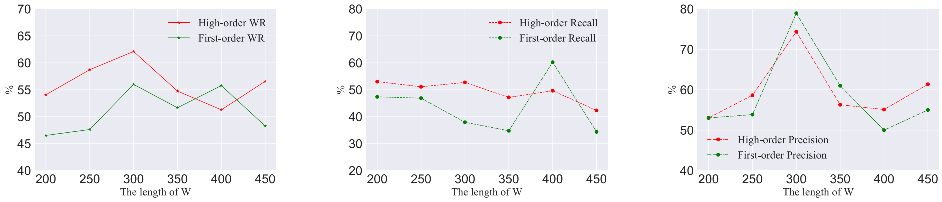
**Winning rate**: The winning rate(WR) is the ratio of the total number of trade profits to the total number of trade during the trading periods of the trading strategy. WR is defined as:

$$WR = \frac{\sum_{\{y_t|g_t<0\}} I\{y_t = -1\} + \sum_{\{y_t|g_t>0\}} I\{y_t = 1\}}{\sum_D I\{y_t \neq 0\}} \times 100\%, \tag{9}$$

where $D$ is all trading signal during the trading period.

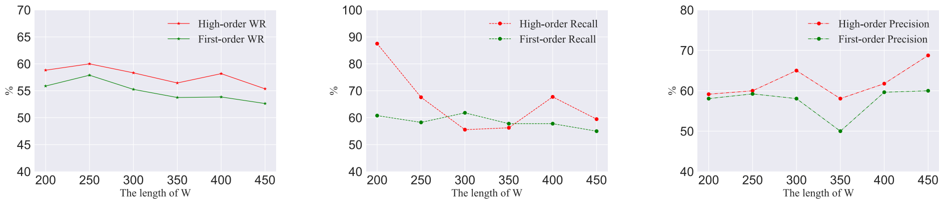
**Maximum drawdown**: The maximum drawdown (MDD) [26] is the maximum loss from a peak to a trough of a portfolio, before a new peak is attained. MDD is an indicator of downside risk over a specified time period. MDD is expressed in percentage terms and computed as:

$$MDD = (TroughValue - PeakValue)/PeakValue \times 100\% \tag{10}$$




(a) Comparison of WR (b) Comparison of Recall (c) Comparison of Precision

**Fig. 7.** Comparison of indicators for different W values of Result 1.




(a) Comparison of WR (b) Comparison of Recall (c) Comparison of Precision

**Fig. 8.** Comparison of indicators for different W values of Result 2.

Intuitively, the MDD describes the worst case scenario possible for a trading strategy. In practice, we prefer to choose a trading strategy with a small MDD.

**Annual return**: Annual return is the return an investment provides over a period of time, expressed as a time-weighted annual percentage:

$$Annual\ Return = ((1+P)^{252/n} - 1) \times 100\%, \tag{11}$$

where P is total return of trading strategy, n is the number of days of trading strategy execution.

**Sharpe ratio**: The Sharpe ratio [27] is a way to examine the performance of an investment by adjusting for its risk in finance. The Sharpe ratio is defined as:

$$Sharpe\ Ratio = \frac{R_p - R_f}{\sigma_p} \times 100\%, \tag{12}$$

where $R_p$, $R_f$ and $\sigma_p$ are the annual return, risk free rate and portfolio standard deviation of trading strategy, respectively. The Sharpe ratio characterizes how well the return of an asset compensates the investor for the risk taken. When evaluating two trading strategies, the one with a higher Sharpe ratio provides better return for the same risk.

The hyper-parameters are optimized via extensive grid search on the validation set, and the best hyper-parameter are selected by the trading strategy optimal mean of WR, recall, precision on the data set. The Hyper-parameter searching interval is as follows: W: {200, 250, 300, 350, 400, 450}, $n$: {2,3}, $N$: {3, 4}, $w$ and $u$: {0.6, 0.62.0.64, 0.66, 0.68}. In order to facilitate calculation and evaluation, axes, trading spreads, trading commissions, and fees were not included in the back test calculations.

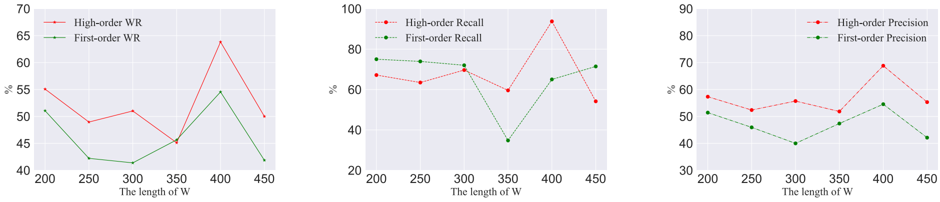
### *5.2. CSI 300 data*

In order to ensure that the model operates in a relatively stable market, we divide the CSI300 into three parts. Each part of the data set corresponds to a relatively stable stock market period with stable policies and strong market. We select different hyper-parameters under the same trading framework for different data sets. This is also consistent with the actual quantitative trading operation, adjusting the model according to market changes. The first half of the testing data is used as a validation set to adjust the hyper-parameters.

**Result1:** Data from January 1st 2013 to June 1st 2013 is used as validation set for adjusting hyper-parameters, we set $N = 4$, $W = 300$, $w = 0.62$, $u = 0.62$, $n = 2$ according to the performance of the trading strategy on the validation set. Data from June 1st 2013 to June 1st 2014 is used for testing, the results are shown in Table 5. We also compare WR, Recall and Precision as W takes 200, 250, 300, 350, 400 and 450 respectively, see Fig. 7.

**Result2:** Data from January 1st 2014 to June 1st 2014 is used as validation set for adjusting hyper-parameters, we set $N = 4$, $W = 200$, $w = 0.6$, $u = 0.68$, $n = 2$. Data from June 1st 2014 to June 1st 2015 for testing, the results are shown in Table 6. The comparison of WR, Recall, Precision under the different $W$ can be found in Fig. 8.

**Result3:** Data from June 1st 2015 to December 31st 2015 is used as validation set, $N = 4$, $W = 400$, $w = 0.6$, $u = 0.6$, $n = 3$, data from January 1st 2016 to January 1st 2017 is used for testing, the results are shown in Table 7. The comparison of WR, Recall, Precision for different $W$ can be found in Fig. 9.




(a) Comparison of WR (b) Comparison of Recall (c) Comparison of Precision

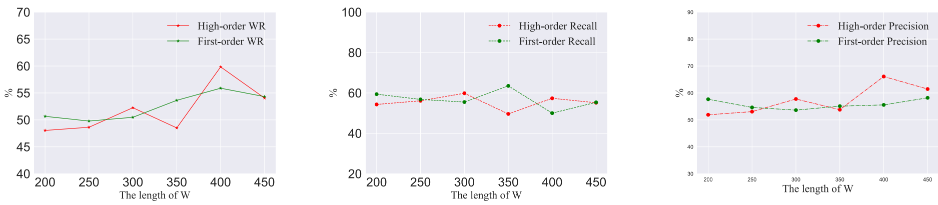
**Fig. 9.** Comparison of indicators for different W values of Result 3.

**Table 5**
Results for 2013-06-01 to 2014-06-01.

| Evaluation indicators | 1-order HMM | 2-order HMM |
|---|---|---|
| WR | 56.00% | 62.11% |
| Long times | 19 | 39 |
| Short times | 31 | 56 |
| Annual return | 10.31% | 37.51% |
| Sharpe ratio | 1.04 | 3.63 |
| MDD | 6.15% | 4.60% |
| Recall | 46.88% | 52.73% |
| Precision | 78.95% | 74.36% |

**Table 6**
Results for 2014-06-01 to 2015-01-01.

| Evaluation indicators | 1-order HMM | 2-order HMM |
|---|---|---|
| WR | 55.88% | 58.82% |
| Long times | 62 | 71 |
| Short times | 6 | 14 |
| Annual return | 19.77% | 26.58% |
| Sharpe ratio | 1.30 | 2.44 |
| MDD | 8.85% | 5.65% |
| Recall | 90% | 87.5% |
| Precision | 58.06% | 59.15% |




(a) Comparison of WR (b) Comparison of Recall (c) Comparison of Precision

**Fig. 10.** Comparison of indicators for different W values of S&P 500.

### 5.3. S&P 500 data

In order to validate the validity and generality of our trading strategy, we also tested our trading strategy on the S&P500 Index data set. The S&P500 is an American stock market index based on the market capitalization of 500 large companies having common stock listed on the NYSE or NADAQ. The data set is obtained from YAHOO Finance.[2] We selected a relatively fixed model hyper-parameters. The observation distribution is set to be Gaussian distribution. We use data from January 1st 2015 to December 31st 2015 for turning hyper-parameters, use data from June 1st 2016 to June 1st 2018 for testing. According to the validation set, we set the strategy parameters: $N = 4$, $W = 400$, $w = 0.6$, $u = 0.6$, $n = 2$, the results are shown in Table 8. The comparison of WR, Recall, Precision for different $W$ can be found in Fig. 10.

2 https://finance.yahoo.com/.

**Table 7**
Results for 2016-01-01 to 2017-01-01.

| Evaluation indicators | 1-order HMM | 3-order HMM |
|---|---|---|
| WR | 54.55% | 63.83% |
| Long times | 27 | 47 |
| Short times | 0 | 5 |
| Annual return | 1.17% | 16% |
| Sharpe ratio | 0.22 | 1.63 |
| MDD | 7.85% | 6.86% |
| Recall | 1 | 93.75% |
| Precision | 54.55% | 63.83% |

**Table 8**
Results for 2016-01-01 to 2018-01-01.

| Evaluation indicators | 1-order HMM | 2-order HMM |
|---|---|---|
| WR | 55.88% | 59.84% |
| Long times | 45 | 59 |
| Short times | 57 | 63 |
| Annual return | 16% | 23% |
| Sharpe ratio | 0.89 | 1.54 |
| MDD | 8.85% | 5.29% |
| Recall | 50% | 57.35% |
| Precision | 55.56% | 66.11% |

### *5.4. Analysis*

For the CSI300 data set, by using the same hyper-parameters, we can see trading strategy based on high-order model take short and long position more frequently than the strategy based on the first-order model. However, the Sharpe ratio of high order model is significantly higher than the first-order one, which indicates that the trading strategy based on high-order model have higher risk resistance and can better identify risks and avoid trading risks. The MDD of high-order model is also much smaller than the first-order model, indicating that the stability of trading strategy based high-order is stronger than strategy based on first-order one. Moreover, the annual return of the trading strategy is at a high level, which shows that this trading strategy is effective in Chinese stock market. For the S&P 500 data, our method also achieves better results. These results indicate that the trading framework based on high-order model is effective and performs much better than the traditional first-order one. We argue the high-order HMM could capture the trend of the stock index and outperforms in various time intervals.

## 6. Conclusion

In this paper, we present a stock market price trend forecasting method by using the high-order hidden Markov model. A state dimension reduction method is used to solve the problem of parameters estimation and decoding of high-order HMM. By making statistical analysis of the daily return of the CSI 300 index, we demonstrate the relationship between the hidden states and the market index price change trend. Based on a dynamic training strategy, we propose an efficient predicting and trading algorithm which requires only a limited amount of historical training data. Experimental results show that our approach performs well in CSI 300 and S&P 500 index trend prediction. Compared to the commonly used first-order HMM, this high-order HMM has higher prediction accuracy and trading frequency. We argue that the high-order HMM might be powerful in modeling long-range time dependence financial phenomena. In the future research, we will study the scale effects of financial time series by using turning parameter techniques in high-order HMMs.

## Acknowledgment

This work is supported by National Key Research and Development Program of China under Grants No. 2017YFB0701702. and No. 2018YFB11074.

## References

[1] M. Ghanavati, R.K. Wong, F. Chen, J. Lee, J. Lee, A hierarchical beta process approach for financial time series trend prediction, in: Pacific-Asia Conference on Knowledge Discovery and Data Mining, 2016, pp. 227–237.
[2] J.R. Koza, S. Bookstore, Trend prediction in financial time series, J. Perinat. Med. 30 (4) (2010) 333–335.
[3] N.F. Johnson, D. Lamper, P. Jefferies, M.L. Hart, S. Howison, Application of multi-agent games to the prediction of financial time series, Physica A 299 (1) (2001) 222–227.
[4] M.A. Kaboudan, Genetic programming prediction of stock prices, Comput. Econ. 16 (3) (2000) 207–236.
[5] M.T. Leung, H. Daouk, A.S. Chen, Forecasting stock indices: a comparison of classification and level estimation models, Int. J. Forecast. 16 (2) (2000) 173–190.

[6] H.S. Zhang, X.Y. Shen, J.P. Huang, Pattern of trends in stock markets as revealed by the renormalization method, Physica A 456 (2016) 340–346.
[7] Y. Yang, G. Liu, Z. Zhang, Stock market trend prediction based on neural networks, multiresolution analysis and dynamical reconstruction, Syst. Eng.-Theory Prac. 391 (12) (2001) 155–156.
[8] J. Yang, Y. Zhang, Application Research of Support Vector Machines in Condition Trend Prediction of Mechanical Equipment, Springer Berlin Heidelberg, 2005, pp. 857–864.
[9] B.B. Nair, V.P. Mohandas, N.R. Sakthivel, A decision tree- rough set hybrid system for stock market trend prediction, Int. J. Comput. Appl. 6 (9) (2010) 1–6.
[10] D.M.Q. Nelson, A.C.M. Pereira, R.A.D. Oliveira, Stock market's price movement prediction with LSTM neural networks, in: International Joint Conference on Neural Networks, 2017, pp. 1419–1426.
[11] W. Huang, Y. Nakamori, S.Y. Wang, Forecasting stock market movement direction with support vector machine, Comput. Oper. Res. 32 (10) (2005) 2513–2522.
[12] L. Shi, X. Ma, L. Xi, X. Hu, Financial data mining based on support vector machines and ensemble learning, in: International Conference on Intelligent Computation Technology and Automation, 2010, pp. 313–314.
[13] M.R. Hassan, B. Nath, StockMarket forecasting using hidden Markov model: A new approach, in: International Conference on Intelligent Systems Design and Applications, 2005. Isda '05. Proceedings, 2006 pp. 192–196.
[14] A. Gupta, B. Dhingra, Stock market prediction using Hidden Markov Models, in: Engineering and Systems, 2012, pp. 1–4.
[15] S.H. Park, J.H. Lee, J.W. Song, T.S. Park, Forecasting Change Directions for Financial Time Series Using Hidden Markov Model, Springer Berlin Heidelberg, 2009, pp. 184–191.
[16] R. Seethalakshmi, B. Krishnakumari, V. Saavithri, Gaussian kernel based HMM for time series data analysis, in: Management Issues in Emerging Economies, 2012, pp. 105–109.
[17] S. Rebagliati, E. Sasso, Pattern recognition using hidden markov models in financial time series, Acta Comment. Univ. Tartu. Math. 21 (1) (2017) 25.
[18] U. Hadar, H. Messer, High-order hidden Markov models - estimation and implementation, in: Statistical Signal Processing, 2009. SSP '09. IEEE/SP 15th Workshop on, 2009, pp. 249–252.
[19] L.R. Rabiner, A tutorial on hidden Markov models and selected applications in speech recognition, Read. Speech Recognit. 77 (2) (1990) 267–296.
[20] L.E. Baum, G.R. Sell, Gro wth transformations for functions on manifolds, Pacific J. Math. 27 (2) (1968) 211–227.
[21] L. Liporace, Maximum likelihood estimation for multivariate observations of Markov sources, IEEE Trans. Inform. Theory 28 (5) (1982) 729–734.
[22] B.H. Juang, Maximum-likelihood estimation for mixture multivariate stochastic observations of markov chains, Bell Labs Tech. J. 64 (6) (1985) 1235–1249.
[23] B.H. Juang, S.E. Levinson, M.M. Sondhi, Maximum likelihood estimation for multivariate mixture observations of markov chains, IEEE Trans. Inform. Theory (1986) 307–309.
[24] A. Viterbi, Error bounds for convolutional codes and an asymptotically optimum decoding algorithm, IEEE Trans. Inform. Theory 13 (2) (1967) 260–269.
[25] G. Pulford, The viterbi algorithm, in: IEE Seminar on Target Tracking: Algorithms and Applications, 2006, pp. 53–65.
[26] A. Chekhlov, S. Uryasev, M. Zabarankin, Drawdown measure in portfolio optimization, Int. J. Theor. Appl. Finance 8 (01) (2005) 13–58.
[27] W.F. Sharpe, The sharpe ratio, J. Portf. Manag. 21 (1) (1994) 49–58.