*Journal of Computer Science and Cybernetics, V.28, N.3 (2012), 206–216*

# A NORMAL - HIDDEN MARKOV MODEL IN FORECASTING STOCK INDEX

LUC TRI TUYEN[1], NGUYEN VAN HUNG[1],
THACH THI NINH[1], PHAM QUOC VUONG[1], NGUYEN MINH DUC[1], DAO XUAN KY[2]

[1]*Institute of Information Technology, Vietnam Academy of Science and Technology*
[2]*Department of Information and Communications, Ninh Thuan*

**Tóm tắt.** Phân tích và dự đoán thị trường cổ phiếu là một trong những lĩnh vực thú vị mà trong đó dữ liệu lịch sử có thể được sử dụng để ước tính và dự đoán dữ liệu và thông tin của tương lai. Về mặt kỹ thuật mà nói, lĩnh vực này có tầm quan trọng cho các chuyên gia trong tài chính và thị trường chứng khoán như việc họ có thể nắm bắt và điều chỉnh xu hướng tương lai hoặc quản lý khủng hoảng theo thời gian. Bài báo sử dụng tiện ích của mô hình Markov ẩn để phân tích, mô hình và dự báo dữ liệu mong muốn khi có dữ liệu quá khứ.

**Abstract.** Stock market analysis and prediction are one of the interesting areas in which past data could be used to anticipate and predict data and information about the future. Technically speaking, this area is of high importance for professionals in the industry of finance and stock exchange as they can lead and direct future trends or manage crises over time. In this paper, we try to take advantage of Hidden Markov Models to analyze, model and predict the required data having the past data.

## 1. INTRODUCTION

Stock market analysis and prediction have a great significance for many professionals in the fields of finance and stock exchange. There are a lot of models fitting the data of many stock prices, hence they estimate the trends, the option prices and give some predictions. Hidden Markov Models is a good candidate to do this.

Hidden Markov Model (HMM) is a widely tool to analyze and predict time series phenomena. HMM has been used successfully to analyze various types of time series including DNA sequence analysis (Cheung, 2004)[2], Speech Signal recognition (Xie, Andreae, Zhang, & Warren, 2004)[12], ECG analysis (Coast, Stern, Cano, & Briller, 1990)[3] etc. In finance, an earlier study (Hassan & Nath, 2005)[8] HMM has been used to generate one-day forecasts of stock prices in a novel way. Other study of Hassan (M. Rafiul Hassan, et al, 2007)[5] combined the HMM used in Hassan and Nath (2005) with an Artificial Neural Network (ANN) and a Genetic Algorithm (GA) to achieve better forecasts. We can refer a more recent study of Rafiul Hassan (Hassan, Elsevier Ltd, 2009)[6] as the other combination of HMM and fuzzy model for stock market forecasting.

In Vietnam, study of financial models is a new issue. There are some studies which used to predict stock price mainly are not based on HMM. A study of Bui Cong Cuong, Pham Van Chien ([1]) based on adaptive Neuro-Fuzzy inference system to predict stock price. In the same way, the authors of [11] have presented an application of a computational intelligence technique - a fuzzy inference system, namely Standard Additive Model (SAM), for predicting stock price time series data.

In this paper, we use HMM with Normal distribution to find out the optimal number of states fitted VN-Index (Vietnam) data from 4/11/2009 to 13/5/2011. After training the model, we propose some predictions and compare with the real data in some days later.

Section 2 gives an over view of HMM and definitions, problems, EM algorithm. Section 3 trains the model with EM algorithm to VN-Index data. Section 4 introduces some predictions about states and close prices in the short future. Throughout the experiment, we used **R** software for calculations and data source in `http://www.cophieu68.com` website.

## 2. HIDDEN MARKOV MODELS

The HMM can be described by different ways of notation, but it has only single interpretation. The EM algorithm is an popular algorithm which used to estimate parameters that maximizing the likelihood function. The Viterbi algorithm finds the best states sequence with respect to the sequence of observations which maximize the likelihood function. These can be found in [13, 10] for more detail of solving HMM problems. In this regard, we use notations and algorithms similar to [13]. The algorithms are built up for HMMs with any probability distributions. In the case of Normal distribution, we replace the algorithm's parameters by Normal distribution's parameters. We then write **R** codes for the algorithms with this replacement, using VN-Index data.

### 2.1. Definitions

A hidden Markov model (HMM) is a statistical Markov model in which the system being modeled is assumed to be a Markov process with unobserved (hidden) states. An HMM can be considered as the simplest dynamic Bayesian network. In a regular Markov model, the state is directly visible to the observer, and therefore the state transition probabilities are the only parameters. In a hidden Markov model, the state is not directly visible, but output, dependent on the state, is visible. Each state has a probability distribution over the possible output tokens. Therefore the sequence of tokens generated by an HMM gives some information about the sequence of states.

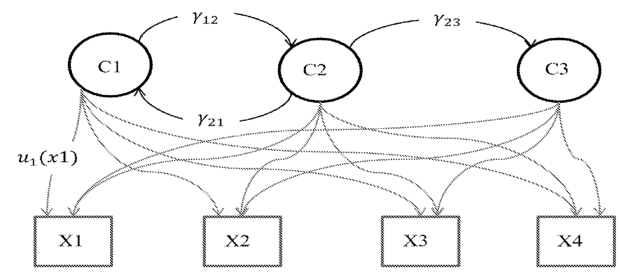


*Figure 2.1.* Probabilistic parameters of a hidden Markov model (example)

$c$ - state;

$x$ - possible observation;
$\Gamma$ - state transition probability; $u$ - output probability.

### 2.2. Elements of Hidden Markov Models

Briefly to delve into the concepts of HMM proposed in [13, 7, 9, 4, 14] , an HMM is a state machine for a system adherent to a Markov process with unobserved states. Specifically, regarding the time series analysis applications, if we denote the hidden state at time $t$ as $C_t$ and the observation at the same time as $X_t$ then the following facts are always true in the HMM:

1. $C_t$ is dependent only on $C_{t-1}$.
2. $X_t$ is dependent only on $C_t$.

To define an HMM, we need

**States Q**: An HMM has a number of states $m$ and it is usually desired to denote the state that the model goes through in time as $\{q_1, q_2, ...q_n\}$ with $n$ as the time length of the observation.
**Observations O**: in any HMM, during time till $n$, there is a sequence of observations as $\{x_1, x_2, ..., x_n\}$.
**Transition Matrix** $\Gamma_{m\times m}$: Each element $\gamma_{ij}$ denotes the probability of transition from state $i$ to state $j$.
**Observation Emission Matrix B**: in which $u_j(x_t)$ denotes the probability of observing $x_t$ in state $j$.
**Prior Probability** $\delta_{m\times 1}$: in which $\delta_i$ denotes the probability of being in state $i$ when at time $t = 1$.

We denote, for $i = 1, 2, ..., m$,

$$p_i(x) = Pr(X_t = x|C_t = i).$$

That is, $p_i$ is the probability mass function of $X_t$ if the Markov chain is in state $i$ at time $t$. The continuous case is treated similarly with $p_i$ to be the probability density function of $X_t$. For discrete-valued observation $X_t$, defining $u_i(t) = Pr(C_t = i)$ for $t = 1, ..., n$, we have

$$Pr(X_t = x) = \sum_{i=1}^{m} Pr(C_t = i)Pr(X_t = x|C_t = i) = \sum_{i=1}^{m} u_i(t)p_i(x).$$

Note that $\mathbf{u(t)} = \mathbf{u(1)\Gamma^{(t-1)}}$, hence

$$Pr(X_t = x) = \mathbf{u(1)\Gamma^{(t-1)}P(x)1'}.$$

where, $\mathbf{u(t)}$ is the row vector and **1'** is the column vector elements 1. $\mathbf{P(x)}$ is the diagonal matrix with $i$ th diagonal element $p_i(t)$.

Thus, we have the likelihood function as

$$L_n = \delta\mathbf{P(x_1)\Gamma P(x_2)\Gamma P(x_3)...\Gamma P(x_n)1'}.$$

If $\delta$, the distribution of $C_1$, is the stationary distribution of the Markov chain, then

$$L_n = \delta\mathbf{\Gamma P(x_1)\Gamma P(x_2)\Gamma P(x_3)...\Gamma P(x_n)1'}.$$

### 2.3. Three basic problems

1. Given observations $\{x_1, x_2, ..., x_n\}$ and model $\lambda = (\Gamma, B, \delta)$, effeciently compute $P(x_1, x_2, ..., x_n|\lambda)$
   - Hidden states complicate the evaluation.
   - Given two models $\lambda_1$ and $\lambda_2$, this can be used to choose the better one.

2. Given observations $\{x_1, x_2, ..., x_n\}$ and model $\lambda = (\Gamma, B, \delta)$, find the optimal state sequence $\{q_1, q_2, ..., q_n\}$:
   - Optimality criterion has to be decided (e.g. maximum likelihood).
   - "Explanation" for the data.

3. Given observations $\{x_1, x_2, ..., x_n\}$, estimate model parameters $\lambda = (\Gamma, B, \delta)$ that maximize the probability $P(x_1, x_2, ..., x_n|\lambda)$.

### 2.4. The EM algorithm (Solution to problem 3)

Since the sequence of states occupied by the Markov chain component of an HMM is not observed, a very natural approach to parameter estimation in HMMs is to treat those states as missing data and to employ EM algorithm [13] in order to find maximum likelihood estimates of the parameters.

In the case of an HMM it is here convenient to represent the sequence of states $c_1, c_2, ..., c_n$ followed by the Markov chain by the zero-one random variables defined as:

$$u_j(t) = 1 \text{ if only if } c_t = j, (t = 1, 2, ..., n)$$

and $\nu_{jk} = 1$ if and only if $c_{t-1} = j$ and $c_t = k$ $(t = 2, 3, ..., n)$.

By this notation, the complete-data log-likelihood (CDLL) of an HMM plus the missing data $c_1, c_2, ..., c_n$ is given by

$$\begin{aligned}\log\left(Pr(x^{(n)}, c^{(n)})\right) &= \log\left(\delta_{c_1}\prod_{t=2}^{n}\gamma_{c_{t-1},c_t}\prod_{t=1}^{n}p_{c_t}(x_t)\right)\\ &= \log\delta_{c_1} + \sum_{t=2}^{n}\log\gamma_{c_{t-1},c_t} + \sum_{t=1}^{n}\log p_{c_t}(x_t).\end{aligned}$$

Hence, the CDLL is

$$\begin{aligned}\log\left(Pr(x^{(n)}, c^{(n)})\right) &= \sum_{j=1}^{m}u_j(1)\log\delta_j + \sum_{j=1}^{m}\sum_{k=1}^{m}\left(\sum_{t=2}^{n}\nu_{jk}(t)\right)\log\gamma_{jk}\\ &\quad + \sum_{j=1}^{m}\sum_{t=1}^{n}u_j(t)\log p_j(x_t) \qquad (2.1)\\ &= \text{term1 } + \text{term2 } + \text{term3.}\end{aligned}$$

The EM algorithm for HMMs proceeds as follows.

- **E step** Replace all the quantities $\nu_{jk}$ and $u_j(t)$ by

$$\widehat{u}_j(t) = Pr(C_t = j|x^{(n)}) = \alpha_t(j)\beta_t(j)/L_n;$$

and

$$\widehat{\nu}_{jk}(t) = Pr(C_{t-1} = j, C_t = k|x^{(n)}) = \alpha_{t-1}(j)\gamma_{jk}p_k(x_t)\beta_t(k)/L_n,$$

where $\alpha_t$ and $\beta_t$ are coresponding to *forward and backward probabilities* (see [13]).

- **M step** Having replaced $\nu_{jk}(t)$ and $u_j(t)$ by $\widehat{u}_j(t)$ and $\widehat{\nu}_{jk}(t)$, maximize the CDLL, expression (2.1), with respect to the three sets of parameters: The initial distribution $\delta$, the transition probability matrix $\mathbf{\Gamma}$ and the parameters of the state-dependent distribution.

The solution is given as follows.

1. For term1: Set $\delta_j = \widehat{u}_j(1)/\sum_{j=1}^m \widehat{u}_j(1) = \widehat{u}_j(1)$.
2. For term2: Set $\gamma_{jk} = f_{jk}/\sum_{k=1}^m f_{jk}$, where $f_{jk} = \sum_{t=2}^n \widehat{\nu}_{jk}(t)$.
3. For term3: This may be easy or difficult, depending on the nature of the assumed state-dependent distributions. For a normal - HMM the state-dependent density is of the form $p_j(x) = (2\pi\sigma_j^2)^{-1/2}\exp\left(-\frac{1}{2\sigma_j^2}(x-\mu_j)^2\right)$, and the maximizing values of the state-dependent parameters $\mu_j$ and $\sigma_j^2$ are

$$\widehat{\mu}_j = \sum_{t=1}^n \widehat{u}_j(t)x_t/\sum_{t=1}^n \widehat{u}_j(t),$$

and

$$\widehat{\sigma}_j^2 = \sum_{t=1}^n \widehat{u}_j(t)(x_t-\widehat{\mu}_j)^2/\sum_{t=1}^n \widehat{u}_j(t).$$

### 2.5. Forecast distributions

For discrete-valued observations the forecast distribution $Pr(X_{n+h} = x|X^{(n)} = x^{(n)})$ can be computed in essentially the way as a ratio of likelihoods:

$$\begin{aligned} Pr(X_{n+h} = x|X^{(n)} = x^{(n)}) &= \frac{Pr(X^{(n)} = x^{(n)}, X_{n+h} = x)}{Pr(X^{(n)} = x^{(n)})} \\ &= \frac{\delta\mathbf{P}(\mathbf{x_1})\mathbf{B_2}\mathbf{B_3}...\mathbf{B_n}\mathbf{\Gamma^h}\mathbf{P}(\mathbf{x})\mathbf{1}'}{\delta\mathbf{P}(\mathbf{x_1})\mathbf{B_2}\mathbf{B_3}...\mathbf{B_n}\mathbf{1}'} \\ &= \frac{\alpha_{\mathbf{n}}\mathbf{\Gamma^h}\mathbf{P}(\mathbf{x})\mathbf{1}'}{\alpha_{\mathbf{n}}\mathbf{1}'}. \end{aligned}$$

Writing $\phi_{\mathbf{n}} = \alpha_{\mathbf{n}}/\alpha_{\mathbf{n}}\mathbf{1}'$, we have

$$Pr(X_{n+h} = x|X^{(n)} = x^{(n)}) = \phi_{\mathbf{n}}\mathbf{\Gamma^h}\mathbf{P}(\mathbf{x})\mathbf{1}'.$$

The forecast distribution can therefore be written as a mixture of the state-dependent probability distribution:

$$Pr(X_{n+h} = x|X^{(n)} = x^{(n)}) = \sum_{i=1}^m \xi_i(h)p_i(x),$$

where the weight $\xi_i(h)$ is the $i$ th entry of the vector $\phi_{\mathbf{n}}\mathbf{\Gamma^h}$.

### 2.6. Viterbi algorithm (Solution to problem 2)

The aim of the algorithm is to find the best states sequence $i_1, i_2, ..., i_n$ with respect to the sequence of observations $x_1, x_2, ..., x_n$ which maximize the likelihood function.

We begin by defining

$$\xi_{1i} = Pr(C_1 = i, X_1 = x_1) = \delta_i p_i(x_1),$$

and for $t = 2, 3, ..., n$,

$$\xi_{ti} = \max_{c_1, c_2, ..., c_{t-1}} Pr(C^{(t-1)} = c^{(t-1)}, C_t = i, X^{(n)} = x^{(n)}).$$

It can then be shown that the probabilities $\xi_{tj}$ satisfy the following recursion, for $t = 2, 3, ..., n$ and $i = 1, 2, ..., m$:

$$\xi_{tj} = \left( \max_i (\xi_{t-1,i}\gamma_{ij}) \right) p_j(x_t).$$

The required maximizing sequence of states $i_1, i_2, ..., i_n$ can be detemined recursively from

$$i_n = \underset{i=1,..,m}{\text{argmax}} \xi_{ni}$$

and, for $t = n-1, n-2, ..., 1$, from

$$i_t = \underset{i=1,...,m}{\text{argmax}} (\xi_{ti}\gamma_{i,i_{t+1}}).$$

### 2.7. State prediction

For $i = 1, 2, ..., m$,

$$Pr(C_{n+h} = i | X^{(n)} = x^{(n)}) = \alpha_{\mathbf{n}} \mathbf{\Gamma^h}(, \mathbf{i})/L_n = \phi_{\mathbf{n}} \mathbf{\Gamma^h}(, \mathbf{i})$$

Note that, as $h \to \infty$, $\phi_{\mathbf{n}} \mathbf{\Gamma^h}$ approaches the stationary distribution of the Markov chain.

## 3. NORMAL - HMM FOR VN-INDEX

Now we use Normal distribution to HMM with different states to fit VN-Index data with 376 close prices from 11/4/2009 to 13/5/2011. The plots of VN-Index data were presented in Figure 3.2. It means the $i$ th state, the probability density $p_i(x) \sim Normal(x, \mu_i, \sigma_i)$. The first thing to do is to train the model to find number of states best and estimate the parameters in the model. The EM algorithm would help us in this.

### 3.1. Finding the best model

A problem which aries naturally when one uses HMM is that of selecting an appropriate model, e.g. of choosing the appropriate number of states $m$, or of choosing between the competing state-dependent distributions such as Poisson and binomial or Normal. There are two criteria called Akaike information criterion (AIC) and Bayesian information criterion (BIC) to do that (see [13]).


*Figure 3.2.* Plots of VN-Index with 376 close prices from 11/4/2009 to 13/5/2011

*Table 1.* VN-Index data: comparison of (stationary) hidden Markov

| Model | -log $L$ | AIC | BIC |
|---|---|---|---|
| 2-state HM | 1597.832 | 3205.664 | 3225.312 |
| 3-state HM | 1510.989 | 3043.978 | 3087.204 |
| 4-state HM | 1439.179 | **2916.358** | **2991.02** |
| 5-state HM | not converge | | |

For the VN-index series, AIC and BIC both select four states. The values of the two criteria are given in Table 1. According to both AIC and BIC, the model with four states is the most appropriate.

Four-state model with initial distribution $(1/4, 1/4, 1/4, 1/4)$, fitted by EM:

$$\mathbf{\Gamma} = \begin{pmatrix} 0.9717 & 0.0283 & 0.0000 & 0.0000 \\ 0.0927 & 0.8106 & 0.0804 & 0.0163 \\ 0.0000 & 0.0748 & 0.8624 & 0.0628 \\ 0.0000 & 0.0000 & 0.0818 & 0.9182 \end{pmatrix}$$

$$\mu = (453.9839, 484.6801, 505.9007, 530.8300)$$

$$\sigma = (10.6857, 7.1523, 6.4218, 13.0746)$$

Figure 3.3 shows us comparison between VN-Index data and a sample data generated by our 4-state HMM. It shows that the model fits the data well.

Figure 3.4 displays, for the fitted four-state model for the VN-Index data, the paths obtained by the Viterbi algorithm. The Viterbi algorithm gives a sequence of states for sequence of data. This figure illustrates the sequence of states which starts at state 4 and ends of state 2. Each dotted line in figure 3.4 presents a state of the fitted HMM.

## 4. PREDICTIONS

### 4.1. State prediction

Table 2 gives, for range of the days, the state predictions based on the four-state model for the VN-Index data. The last day of the data is 13/05/2011, we predict for 30 days later

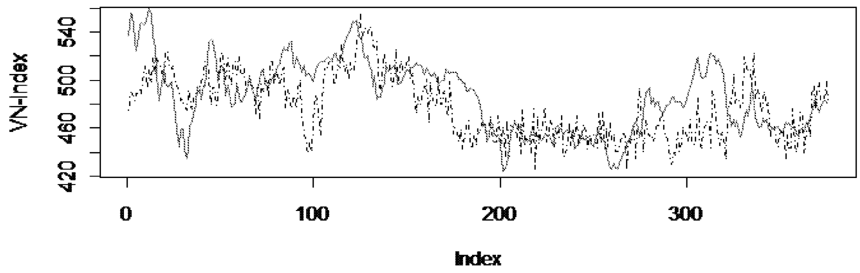


*Figure 3.3.* A generated sample in dotted line with VN-Index data in continuous line

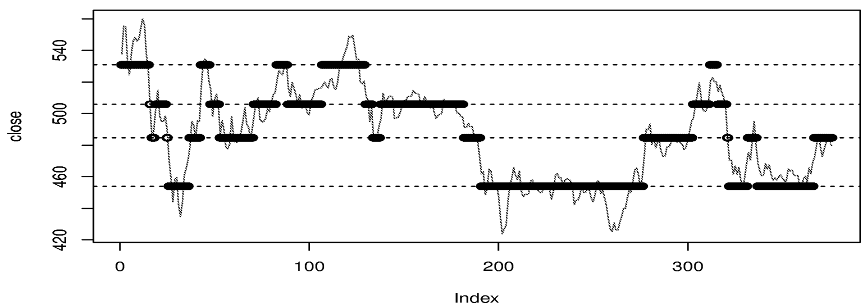


Figure 3.4. VN-Index data: global decoding according to four-state HMMs. The horizontal lines indicate the state-dependent means

and see the state which is the most probability for each day.

We see that the most probability for the first 7 days is the state 2, and the left days is the state 1 and the predicted state sequence converges to state 1. Therefore, the model is not good to predict for long time but it is significant for short time. However, we can update new data to have the new predictions.

Now we update the VN-Index data from 14/5/2011 to 23/6/2011 with 30 close prices in oder to compare the real states with the predicted states.

Figure 4.5 shows that the close prices for this 30 days are almost in state 1. It is clear that our predictions are correct.

## 4.2. Forecast distributions

As we mentioned in Section 2.2, the figure 4.6 which displays ten of the forecast distributions for the VN-Index data finds out that the forecast distribution approaches its limiting distribution quite fast.

Since the entire probability distribution of the forecast is known, it is possible to make interval forecast. This is illustrated in Table 3, which lists statistics of some forecast distributions for the VN-Index data fitted with four-state Normal HMM.

Table 2. VN-Index data. State prediction using a four-state Normal HMM: the probability that the Markov chain will be in a given state in the specified day.

| Days | [,1] | [,2] | [,3] | [,4] | [,5] | [,6] | |
|---|---|---|---|---|---|---|---|
| State=[1,] | 0.0975 | 0.1695 | 0.2261 | 0.2709 | 0.3065 | 0.3350 | |
| [2,] | 0.8062 | 0.6622 | 0.5517 | 0.4665 | 0.4005 | 0.3492 | |
| [3,] | 0.0799 | 0.1351 | 0.1724 | 0.1971 | 0.2128 | 0.2223 | |
| [4,] | 0.0162 | 0.0330 | 0.0496 | 0.0653 | 0.0800 | 0.0933 | |
| | [,7] | [,8] | [,9] | [,10] | [,11] | [,12] | [,13] |
| [1,] | 0.3579 | 0.3764 | 0.3915 | 0.4039 | 0.4141 | 0.4225 | 0.4296 |
| [2,] | 0.3092 | 0.2778 | 0.2530 | 0.2334 | 0.2177 | 0.2052 | 0.1951 |
| [3,] | 0.2274 | 0.2296 | 0.2298 | 0.2288 | 0.2270 | 0.2248 | 0.2224 |
| [4,] | 0.1053 | 0.1160 | 0.1255 | 0.1338 | 0.1410 | 0.1473 | 0.1527 |
| | [,14] | [,15] | [,16] | [,17] | [,18] | [,19] | [,20] |
| [1,] | 0.4355 | 0.4405 | 0.4448 | 0.4484 | 0.4515 | 0.4542 | 0.4565 |
| [2,] | 0.1870 | 0.1803 | 0.1749 | 0.1705 | 0.1669 | 0.1639 | 0.1614 |
| [3,] | 0.2200 | 0.2176 | 0.2154 | 0.2133 | 0.2113 | 0.2096 | 0.2080 |
| [4,] | 0.1573 | 0.1613 | 0.1647 | 0.1676 | 0.1701 | 0.1722 | 0.1739 |
| | [,21] | [,22] | [,23] | [,24] | [,25] | [,26] | [,27] |
| [1,] | 0.4586 | 0.4604 | 0.4619 | 0.4633 | 0.4646 | 0.4657 | 0.4667 |
| [2,] | 0.1593 | 0.1576 | 0.1561 | 0.1549 | 0.1539 | 0.1530 | 0.1523 |
| [3,] | 0.2066 | 0.2053 | 0.2041 | 0.2031 | 0.2022 | 0.2014 | 0.2007 |
| [4,] | 0.1754 | 0.1766 | 0.1776 | 0.1784 | 0.1791 | 0.1797 | 0.1801 |
| | [,28] | [,29] | [,30] | | | | |
| [1,] | 0.4676 | 0.4684 | 0.4692 | | | | |
| [2,] | 0.1517 | 0.1512 | 0.1507 | | | | |
| [3,] | 0.2000 | 0.1995 | 0.1990 | | | | |
| [4,] | 0.1805 | 0.1807 | 0.1809 | | | | |

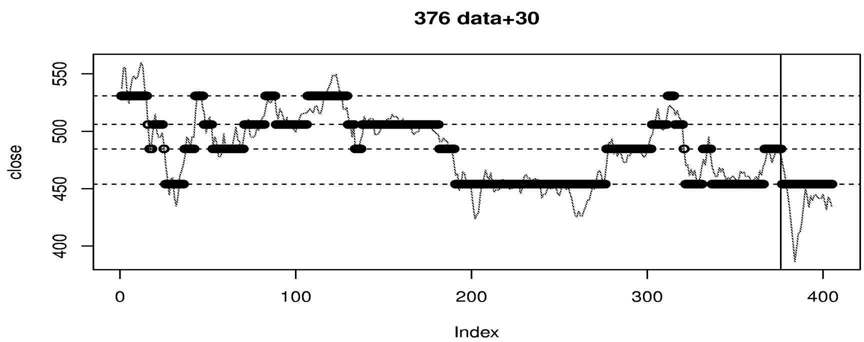


*Figure 4.5.* The comparison between state-predictions and real data-state for 30 days.

## 5. CONCLUSION AND FUTURE WORK

In this paper, we have modeled the stock return as a mixture of Gaussian and discrete Markov Chain in order to improve the predictability of the stock model. We simply see that

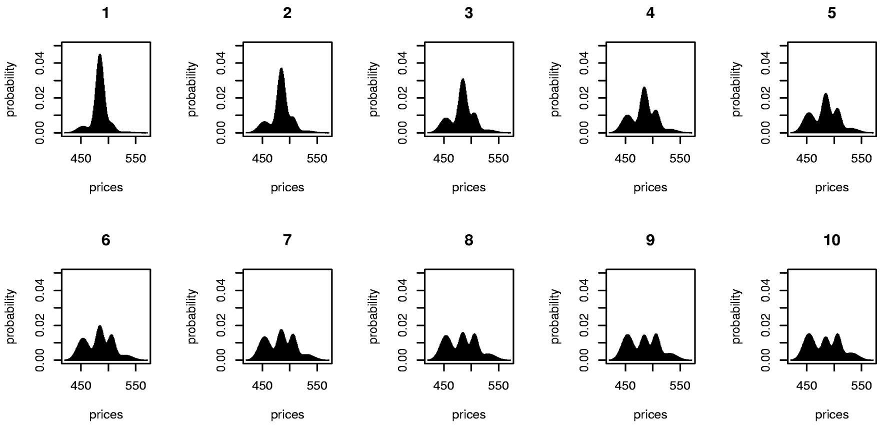


Figure 4.6. VN-Index data, four-state Normal HMM: forecast distributions for 1 to 10 days ahead.

*Table 3*. VN-Index data, four-state Normal HMM: forecasts

| horizone | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| forecast mode | 484.9625 | 484.9625 | 484.9625 | 484.9625 | 484.9625 | 484.9625 |
| forecast mean | 484.0872 | 483.7894 | 483.5800 | 483.4329 | 483.3288 | 483.2544 |
| forecast interval | [469; 499] | [454; 504] | [449; 507] | [450; 500] | [434; 494] | [420; 504] |
| probability | 0.805 | 0.807 | 0.806 | 0.680 | 0.682 | 0.791 |
| observations | 479.7 | 471.5 | 464.4 | 454.9 | 444.9 | 432.9 |

the model gives the predictions correctly.

As a future work, we can test the effectiveness of other economic data. We used default data for the economic states. However, the volatility index data can be a good candidate to extract the economic situation because it can give us direct estimation of variance. As we mentioned in the introduction, the macroeconomic data is also directly connected to the economic situation. By testing the effectiveness of the candidates, we can improve the predictability. Moreover, if we can construct a choosing algorithm that chooses the most effective candidate by learning, then we might build efficient trading machine.

## REFERENCES


[1] Bui Cong Cuong, Pham Van Chien, An experiment result based on adaptive neuro-fuzzy inference system for stock price, *Journal of Computer Science and Cybernetics* **27** (2011).

[2] L. W. K. Cheung, Use of runs statistics for pattern recognition in genomic DNA sequences, *Journal of Computational Biology* **11** (1) (2004) 107—124.

[3] D. A. Coast, R. M. Stern, G. G. Cano, S. A. Briller, An approach to cardiac arrhythmia analysis using hidden Markov models, *IEEE Transactions on Biomedical Engineering* **37** (9) (1990) 826—836.

[4] Phil Blunsom, *Hidden Markov Models*, 2004.

[5] Rafiul Hassan, Baikunth Nath, Michael Kirley, A fusion model of HMM, ANN and GA for stock market forecasting, *Expert Systems with Applications* **33** (2007) 171-–180.

[6] Rafiul Hassan, *A combination of hidden markov model and fuzzy model for stock market forecasting*, Elsevier, 2009.

[7] Rafiul Hassan, "Hybrid hmm and soft computing modeling with applications to time series analysis" Phd thesis, Computer Science and Software Engineering Department, University of Melbourne, Australia, 2007.

[8] M. R. Rafiul Hassan, B. Nath, Stock market forecasting using hidden markov model: a new approach, *Proceedings of 5th international conference on intelligent system design and application*, Poland, 2005 (192-–196).

[9] Ramaprasad Bhar, Shigeyuki Hamori, *Hidden Markov Models Applications to Financial Economics*, Springer Science + Business Media, Inc., 2004.

[10] Ramaprasad Bhar, Shigeyuki Hamori, *Hidden Markov Models: Applications to Financial Economics*, Kluwer Academic Publishers, 2004.

[11] Sang Thanh Do, Thi Thanh Nguyen, Dong-Min Woo and Dong-Chul Park, Standard Additive Fuzzy System for Stock Price Forecasting, *Lecture Notes in Computer Science* **5991** (2010) 279–288.

[12] H. Xie, P. Andreae, M. Zhang, & Warren, P., Learning models for english speech recognition, *Proceedings of the 27th Conference on Australasian Computer Science*, 2004 (323—329).

[13] Walter Zucchini, Iain MacDonald, *Hidden Markov Models for Time series: An introduction Using* ***R***, Chapman and Hall, New York, 2009.

[14] Wikipedia. Hidden markov model, 2009.
`http://en.wikipedia.org/wiki/Hidden_Markov_model.`

*Received on May 06, 2012*
*Revised on December 19, 2012*