INDIAN INSTITUTE OF TECHNOLOGY, KANPUR

M. TECH, INDUSTRIAL & MANAGEMENT ENGINEERING

IME 625A

Introduction to Stochastic Processes and their Applications

Project Report On

"Stock Price Prediction using Hidden Markov Model"

Submitted To:
Dr. Avijit Khanra

Submitted By:
Bhavesh Nehete (21114002)
Jai Wason (21114007)
Shikhar Srivastava (21114021)

INDEX

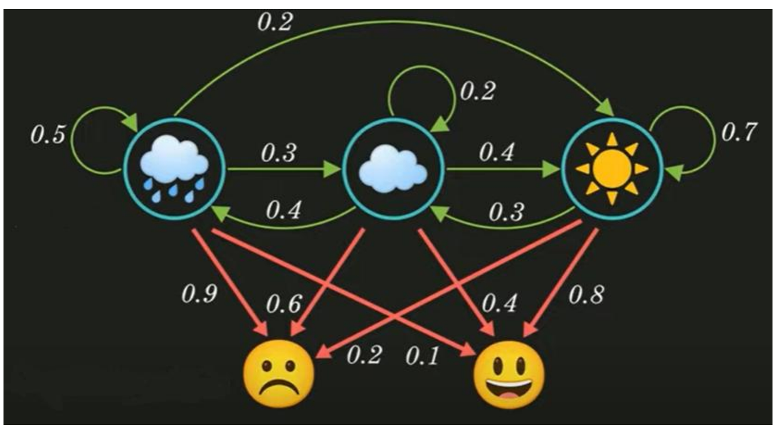
Topic **Page no.**

STOCK PRICE PREDICTION USING HIDDEN MARKOV MODEL

Introduction to Hidden Markov Chain:

Let's get some terms and concepts familiarized for this project, which involves the concept of a hidden Markov chain. Concept of hidden Markov chain was introduced because in many cases it is not possible to observe the transition states but we can observe certain signals which are induced on the basis of current transition states and we want to do some analysis on the transition states.

To understand better consider a scenario where a person(Jai) is living in a remote area and can only be contacted over the phone. There are only three types of weather that can occur in that area on any particular day namely - sunny, cloudy and rainy. Only one weather can occur each day. The weather on each given day is solely determined by the weather the day before. Jai can have one of the two moods namely - sad and happy with some probability based on the weather of each day. His friend (Bhavesh) who is living in another town wants to know the weather in the remote area. But he cannot ask Jai about the weather there. Instead he asked Jai about his mood and then guessed the weather on that particular day. Transition probabilities of weather and probabilities of mood for a given weather is shown in the figure below.



We can observe that the weather follows a Markov chain with a transition probability matrix A on any given day.

$$A = \begin{bmatrix} 0.7 & 0.3 & 0 \\ 0.4 & 0.2 & 0.4 \\ 0.2 & 0.3 & 0.5 \end{bmatrix}_{3 \times 3}$$

and mood follows the following probability matrix(known as *emission matrix*[1]) B:

$$B = \begin{bmatrix} 0.2 & 0.8 \\ 0.6 & 0.4 \\ 0.9 & 0.1 \end{bmatrix}_{3 \times 2}$$

Since Bhavesh cannot ask Jai about the weather there, the weather of that location is hidden and the observable entity is Jai's mood which depends only on the current weather. So this is the hidden Markov chain.

Now let's suppose Bhavesh gets to know that Jai is in a happy mood today. So the question of what is the probability that today's weather is cloudy or which weather is most likely to happen today can be found using the concept of hidden Markov chain.

Let W denote today's weather and M denote today's Jai mood. For sake of simplicity in writing we write sunny, cloudy and rainy by s, c and r respectively. Similarly we write Sd and h for sad and happy respectively. Then to know about which weather is most likely to happen today we have to find out the conditional probability of all the weather and the weather which has the highest probability is most likely to happen. One by one we figure out the probabilities.

$P(W = c|M = h) = \frac{P(M=h|W=c)P(W=c)}{P(M=h)}$ [Bayes theorem]

$\Rightarrow P(W = c|M = h) = \frac{P(M=h|W=c)P(W=c)}{P(M=h|W=s)P(W=s)+P(M=h|W=c)P(W=c)+P(M=h|W=r)P(W=r)}$

Now $P(M|W)$ can be found in matrix B. But individual probability $P(W)$ is not given. It is assumed to take *steady state probabilities*[2] $\pi(s), \pi(c)$ and $\pi(r)$ which comes out to be $\pi(s) = 0.509, \pi(c) = 0.273$ and $\pi(r) = 0.218$.

So,
$$P(M = h) = (0.8)(0.509) + (0.4)(0.273) + (0.1)(0.218) = 0.5382$$
$$\Rightarrow P(W = c|M = h) = \frac{(0.4)(0.273)}{0.5382} = 0.203$$

Similarly,
$$P(W = r|M = h) = \frac{P(M = h|W = r)P(W = r)}{P(M = h)} = \frac{(0.1)(0.218)}{0.5382} = 0.04$$
$$P(W = s|M = h) = \frac{P(M = h|W = s)P(W = s)}{P(M = h)} = \frac{(0.8)(0.509)}{0.5382} = 0.757$$

We get the highest probability of being sunny. Hence it is most likely to be a sunny day.

Now suppose we are given a set of moods of 4 consecutive days and want to find the probability of being rainy on the 4^{th} day. Let M_i and W_i denote the mood and weather on i^{th} day. and set of moods O is given to be: $O = \{M_1, M_2, M_3, M_4\} = \{h, Sd, h, Sd\}$

The probability of being rainy weather on 4^{th} day is
$$P(W_4 = r|\{M_1, M_2, M_3, M_4\} = \{h, Sd, h, Sd\}) = \frac{P(\{M_1, M_2, M_3, M_4\} = \{h, Sd, h, Sd\}, W_4 = r)}{P(\{M_1, M_2, M_3, M_4\} = \{h, Sd, h, Sd\})}$$

We see that it is not easy to calculate numerator as well as denominator in the above expression. So here come some important algorithms related to HMM. Before this we formalize some terms.

Let H denote the set of all possible hidden states (cannot be observed) that a HMM can take and O denote the set of all possible observed states in a HMM. Let $\{X_n: n = 0,1,2, ...\}$ is a stationary Markov chain of hidden states with $X_n \in H$ for $n = 0,1,2, \ldots$ with transition probabilities as $P(X_{n+1} = j | X_n = i) = p_{i,j}$ and initial probabilities as $P(X_0 = i) = p_i \; \forall \; i, j \in H$ and $n = 0,1,2, \ldots$ Let Y_n denote the observed state ($Y_n \in O$) which is induced due to the hidden state X_n, (Y_n depends only on X_n) $\forall \; n = 0,1,2, \ldots$ with emission probabilities as $P(Y_n = k | X_n = i) = e_{i,k} \forall \; i \in H, k \in O$ and $n = 0,1,2, \ldots$ This is called hidden Markov model.

Let P denote the transition probability matrix as $P = [p_{i,j}]_{s \times s}$, E denote the emission probability matrix as $E = [e_{i,k}]_{s \times t}$ and I denote initial probability matrix as $I = [p_i]_{s \times 1}$ where $s = |H|$, cardinality of H and $t = |O|$, cardinality of O.

Let $\boldsymbol{Y}_{n+1}$ denote the sequence of first $n + 1$ random variables $Y_0, Y_1, Y_2, \ldots, Y_n$ of observed states and $\boldsymbol{X}_{n+1}$ denote the sequence of first $n + 1$ random variables $X_0, X_1, X_2, \ldots, X_n$. Let $\boldsymbol{k}_{n+1}$ denote the sequence of fixed variables $k_0, k_1, k_2, \ldots, k_n$ where $k_i \in O \; \forall \; 0 \leq i \leq n$ and $\boldsymbol{j}_{n+1}$ denote the sequence of fixed variables $j_0, j_1, j_2, \ldots, j_n$ where $j_i \in H \; \forall \; 0 \leq i \leq n$.

We'll look at three different forms of HMM queries.

1. What is the probability of observing a given sequence of states in a HMM?

   $P(Y_0 = k_0, Y_1 = k_1, Y_2 = k_2, \ldots, Y_n = k_n)$

   or it can be written as $P(\boldsymbol{Y}_{n+1} = \boldsymbol{k}_{n+1}) \; \forall \; n = 0,1,2, \ldots$

   This is solved using *Forward Backward Algorithm.*

2. If a sequence of observed states is given, find the most likely sequence of hidden states that will induce the given sequence of observed states?

   Finding the sequence of hidden states $\boldsymbol{j}_{n+1} = (j_0, j_1, j_2, \ldots, j_n)$ that maximizes the probability

   $$P(X_0 = j_0, X_1 = j_1, \ldots, X_n = j_n | Y_0 = k_0, Y_1 = k_1, Y_2 = k_2, \ldots, Y_n = k_n)$$

   or it can be written as $P(\boldsymbol{X}_{n+1} = \boldsymbol{j}_{n+1} | \boldsymbol{Y}_{n+1} = \boldsymbol{k}_{n+1}) \; \forall \; n = 0,1,2, \ldots$

   This is solved using *Viterbi Algorithm.*

3. Given the sequence of observed states $\boldsymbol{Y}_{n+1}$, number of hidden states s and number of observed states t, determine the HMM parameters i.e. probability matrices P, E and I.

   This is solved using *Baum-Welch Algorithm.*

[Derivations of first two algorithms are given in Appendix.]

<u>Baum-Welch Algorithm:</u>

The Baum–Welch algorithm is a subset of the Expectation-maximization (EM) technique for locating unknown hidden Markov model parameters. The forward-backward algorithm, which we covered earlier, is used to calculate the parameters for the expectation step.

<u>Working of Baum-Welch algorithm:</u>

1. **<u>Initial Phase:</u>** In this phase, transition probabilities $p_{i,j}$, emission probabilities $e_{j,k}$ and initial probabilities $p_i \; \forall \; i, j \in H, k \in O$ are guessed randomly.
2. **<u>Forward Phase:</u>** In this phase, the recursive function for the *forward algorithm* is calculated which we have already calculated earlier.

$$F_{n+1}(j) = e_{j,k_n} \sum_{i \in H} F_n(i) p_{i,j}$$

3. **<u>Backward Phase:</u>** In this phase, the recursive function for the *backward algorithm* is calculated which we have already calculated earlier.

$$B_m(i) = \sum_{j \in H} p_{i,j} e_{j,k_{m+1}} B_{m+1}(j)$$

The importance of the forward and backward phase formulas can be explained as follows::

- The forward phase formula is really beneficial. When we have one observation point say(o_1), we can plug it into the formula of forward phase to get the probability of each possible hidden state, from which state with highest probability is chosen as our answer. When we enter the next observation point say(o_2)into the formula, it generates a new probability distribution for you to choose from, but this one is based not only on the current observation point o_2 , transition probabilities and emission probabilities but it is also influenced by o_1. This approach is called *filtering*.
- Assume we've already accumulated a significant number of observation points, and we know that the earlier the observation point, the less observable data your answer is based on. As a result, we'd like to improve it by *'injecting'* information from later data into the earlier ones in some way. Here the backward phase formula will be used. This approach is called *smoothing*.

4. **<u>Update Phase:</u>**

   In this phase, we calculate the probabilities for the hidden states using the forward and backward phase formula at any time t, given to us the entire sequence of the observation values.

   $\eta(j) = P(X_m = j | \boldsymbol{Y}_{n+1} = \boldsymbol{k}_{n+1}) = \frac{F_{m+1}(j) B_m(j)}{\Sigma_{j \in H} F_{m+1}(j) B_m(j)}$

   $\xi(j,k) = P(X_m = j, X_{m+1} = k | \boldsymbol{Y}_{n+1} = \boldsymbol{k}_{n+1}) = \frac{F_{m+1}(j)\, B_m(k)\, p_{j,k}\, e_{j,s_2}}{\Sigma_{j \in H} F_{m+1}(j)\, B_m(j)\, p_{j,k}\, e_{j,s_2}}$ where s_2 is observed state at step $m+1$.

   The probability values of the HMM can now be updated as follows:

   - $p_i^* = \eta(j)$
   - $p_{i,j}^* = P(X_m = j | X_{m-1} = i) = \frac{\sum_m \xi(i,j)}{\sum_m \eta(i)}$
   - $e_{j,s_1}^* = P(Y_m = s_1 | X_m = j) = \frac{\sum_m \eta(j)\, \mathbb{1}(Y_m = s_1)}{\sum_m \eta(j)}$

We previously discussed how the Baum-Welch method is a subset of the EM (Expectation Maximization) technique. The E-step is formed by the forward and the backward phase formulas as they predict the expected hidden states given the observation values, transition probabilities, emission probabilities and initial probabilities. The update phase creates the M-step by calculating expressions so that the L.H.S. parameters best fit the expected hidden states given the observation data.

Now, let us try to understand the essence of the Baum-Welch algorithm through a simple example. Let us assume we have a chicken from which we collect eggs everyday. A variety of unknown and hidden circumstances influence whether or not the chicken has laid eggs for collection. We can, however, assume (for the sake of simplicity) that the chicken is always in one of two states (H_1, H_2) that influences whether or not it lays eggs, which are our observed states $(O_{1,}O_2)$, and that this state is exclusively determined by the state the previous day. Now, we don't know the initial state (H_0), the transition probability matrix for the states (H_1, H_2) and also the conditional probability that a chicken lays eggs given a particular state. We start our analysis by making an initial guess for transition probabilities $p_{i,j}$, emission probabilities $e_{j,k}$ and the initial states probabilities $p_i \; \forall \; i,j \in \{H_1, H_2\} = H, k \in \{O_1, O_2\} = O$.

H_1, H_2 denote the unknown hidden states.

O_1 denotes the observed state of getting no eggs.

O_2 denotes the observed state of getting eggs.

Transition probabilities

p_{ij}	H_1	H_2
H_1	0.5	0.5
H_2	0.3	0.7

Emission probabilities

$e_{j,k}$	O_1	O_2
H_1	0.3	0.7
H_2	0.8	0.2

Initial state probabilities

	p_i
H_1	0.2
H_2	0.8

Suppose we observe the following set of observations: $\{O_1, O_1, O_1, O_1, O_1, O_2, O_2, O_1, O_1, O_1\}$. This leads to following set of transitions between days:

$$\{O_1 \to O_1, O_1 \to O_1, O_1 \to O_1, O_1 \to O_1, O_1 \to O_2, O_2 \to O_2, O_2 \to O_1, O_1 \to O_1, O_1 \to O_1\}$$

We will now try to obtain the updated transition probabilities. The probability of observing the sequence $O_1 \to O_1$ and the corresponding states being H_1 and then H_2 is given by: $p_{H_1} e_{H_1,O_1} p_{H_1,H_2} e_{H_2,O_1}$.

Observed sequence ($\boldsymbol{O_i \to O_j}$)	**Probability of observing the sequence if state is $\boldsymbol{H_1}$then $\boldsymbol{H_2}$** $\{p_{H_i} e_{H_i O_i} p_{H_i H_j} e_{H_j O_j}\}$	**Highest probability of observing that sequence**
$O_1 \to O_1$	$0.2 * 0.3 * 0.5 * 0.8 = 0.024$	$0.3584\ (H_2 \to H_2)$
$O_1 \to O_1$	$0.2 * 0.3 * 0.5 * 0.8 = 0.024$	$0.3584\ (H_2 \to H_2)$

$O_1 \rightarrow O_1$	$0.2 * 0.3 * 0.5 * 0.8 = 0.024$	$0.3584\ (H_2 \rightarrow H_2)$
$O_1 \rightarrow O_1$	$0.2 * 0.3 * 0.5 * 0.8 = 0.024$	$0.3584\ (H_2 \rightarrow H_2)$
$O_1 \rightarrow O_2$	$0.2 * 0.3 * 0.5 * 0.2 = 0.006$	$0.1344\ (H_2 \rightarrow H_1)$
$O_2 \rightarrow O_2$	$0.2 * 0.7 * 0.5 * 0.2 = 0.014$	$0.0490\ (H_1 \rightarrow H_1)$
$O_2 \rightarrow O_1$	$0.2 * 0.7 * 0.5 * 0.8 = 0.024$	$0.0896\ (H_2 \rightarrow H_2)$
$O_1 \rightarrow O_1$	$0.2 * 0.3 * 0.5 * 0.8 = 0.024$	$0.3584\ (H_2 \rightarrow H_2)$
$O_1 \rightarrow O_1$	$0.2 * 0.3 * 0.5 * 0.8 = 0.024$	$0.3584\ (H_2 \rightarrow H_2)$
Total	$\mathbf{0.22}$	$\mathbf{2.4234}$

Now, the new estimation of the probability of transition from $H_1 \rightarrow H_2$ is given by: $\frac{0.22}{2.4234} = 0.091$. This new probability value is referred to as *Pseudo probability*. Similarly we will do the same step for the transition from $H_2 \rightarrow H_1$,$X_1 \rightarrow H_1$ and finally $H_2 \rightarrow H_2$. We will then normalize these pseudo probability values so they add up to 1.

The updated transition probability matrix after all these iterations will look like:

Old transition probabilities

p_{ij}	H_1	H_2
H_1	0.5	0.5
H_2	0.3	0.7

New transition probabilities (Pseudo probabilities)

p_{ij}	H_1	H_2
H_1	0.0598	0.0908
H_2	0.2179	0.9705

New transition probabilities (normalized)

p_{ij}	H_1	H_2
H_1	0.3973	0.6027
H_2	0.1833	0.8167

Now, we will do a similar thing for obtaining the updated emission probabilities. At first we will assume that observation O_2 comes from H_1.

Observed sequence $(\boldsymbol{O_i} \rightarrow \boldsymbol{O_j})$	Probability of observing the sequence if $\boldsymbol{O_2}$ is assumed to come from $\boldsymbol{H_1}$	Highest probability of observing that sequence
$\boldsymbol{O_1} \rightarrow \boldsymbol{O_2}$	$0.1344\ (\boldsymbol{H_2} \rightarrow \boldsymbol{H_1})$	$0.1344\ (\boldsymbol{H_2} \rightarrow \boldsymbol{H_1})$
$\boldsymbol{O_2} \rightarrow \boldsymbol{O_2}$	$0.0490\ (\boldsymbol{H_1} \rightarrow \boldsymbol{H_1})$	$0.0490\ (\boldsymbol{H_1} \rightarrow \boldsymbol{H_1})$

$\boldsymbol{O_2} \rightarrow \boldsymbol{O_1}$	0.0560 ($\boldsymbol{H_1} \rightarrow \boldsymbol{H_2}$)	0.0896 ($\boldsymbol{H_2} \rightarrow \boldsymbol{H_2}$)
Total	0.2394	0.2730

The updated conditional probability value of observing $\boldsymbol{O_2}$ given $\boldsymbol{H_1}$ becomes $\frac{0.2394}{0.2730} = 0.8769$.

We will repeat the same procedure for observing $\boldsymbol{O_1}$ from $\boldsymbol{H_1}$, $\boldsymbol{O_2}$ from $\boldsymbol{H_2}$ and finally $\boldsymbol{O_1}$ from $\boldsymbol{H_2}$. Then we will add up the probability values for the respective observations and then normalize so they add up to 1.

Old emission probabilities

$e_{j,k}$	S_1	S_2
X_1	0.3	0.7
X_2	0.8	0.2

New emission probabilities (Estimated)

$e_{j,k}$	S_1	S_2
X_1	0.0404	0.8769
X_2	1.0000	0.7385

New emission probabilities (Normalized)

$e_{j,k}$	S_1	S_2
X_1	0.0441	0.9559
X_2	0.5752	0.4248

To determine the initial probabilities p_i, we assume that observed sequences start from state S_1 and calculate the highest probability, then repeat for S_2. We then normalize the updated probability values to get an updated initial vector. When the probability starts converging to some satisfactory level, we will stop.

<u>Stocks Prices as Hidden Markov Model</u>

There are many underlying hidden states that derive the stock prices such as demand for the stock, any mergers and acquisitions etc. which generally are hidden to the investors and the investors can only observe the daily opening and closing prices, high and low prices for any given stock.

We don't know the number of hidden states as well as the transition probabilities, emission probabilities, or initial probabilities and we want to find the optimal model parameter for the set of observations we have for our stocks.

Thus, the Baum-welch algorithm (as stated above) can be used to calculated the parameters of the model.

In the above example the emission of the states was discrete but that's not the case for the stock prices parameters that we are observing. Therefore, we have modelled the observations at the given state as Gaussian emission.

There are to find . The optimal number of states for the model in HMM can be calculated using a statistical techniques BIC (Bayesian Information Criteria)(*see reference 2)* but we are assuming 3 hidden states in the model namely Bearish, stagnant and Bullish for simplicity.

So, we have used the Axis bank stock data containing the daily opening, closing, high and the low prices of past 110 working days and we are using these data to form an observation vector at time t as,

$$\widehat{O}_t = \left(\frac{Close - open}{open}, \frac{high - open}{open}, \frac{open - low}{open} \right)$$
$$\widehat{O}_t = \left(fracChange, fracHigh, fracLow \right)$$

So, the parameters of the model are,

$\lambda = (A, \mu^k, \Sigma^k, \pi)$

$A =$ Transition Probability Matrix.

$\mu^k =$ Mean Vector for emission being in state k.

$\Sigma^k =$ Covariance Matrix for emission being in state k.

$\pi =$ The Initial Probability Matrix.

We have used the open-source Python library hmmlearn to implement our model and calculate the parameters. We will split our data in training and testing data and the former will be used to build the model while the later will be used to plot the graph between the predicted and the actual stock price of the data.

First we will initialize the model parameters randomly and then we will update it by the method discussed in the above example for 10 iterations on the observed data.

We used the model to predict the stock closing price in the test dataset using a latency of observed vectors of 20 days $(d)$to predict the fraction closing price of the $\widehat{O}_{d+1}$ given the opening price of the $(d + 1)$.

To compute the fractional change $\widehat{O}_{d+1}$ we used the MAP (Maximum A Posteriori)(*see reference 2*) estimate of the observation on the $d + 1$ day.

$$\widehat{O}_{d+1} = \arg\max_{\widehat{O}_{d+1}} P\left(O_{d+1} \middle| O_1, O_2, O_3 .....O_d, \lambda\right)$$
$$\widehat{O}_{d+1} = \frac{P\left(O_1, O_2, O_3 .....O_d, O_{d+1} \middle| \lambda\right)}{P\left(O_1, O_2, O_3 .....O_d, \lambda\right)}$$

So, we have taken 50 observations of fracChange, 10 observations of fracHigh and 10 observations of fracLow in their range evenly distributed and thus merged all 50*10*10 = 5000 discrete observations that are possible for $\widehat{O}_{d+1}$. As, the denominator is not dependent on $\widehat{O}_{d+1}$ the MAP estimates now become, $\widehat{O}_{d+1} = \arg\max_{\widehat{O}_{d+1}} P\left(O_1, O_2, O_3 .....O_d, O_{d+1} \middle| \lambda\right)$

We vary the value of $\widehat{O}_{d+1}$ over all the discrete observation and thus calculate the maximum MAP.

We are maximizing the likelihood of the sequence of observations to predict the closing price of the stock. The closing price can be calculated by the given opening price for that day and the predicted $\widehat{O}_{d+1}$ .We have plotted the predicted and the actual prices for 15 days.

The first predicted day is the 89 day which is the first day of the test data and the graph for axis bank stock is as follows.

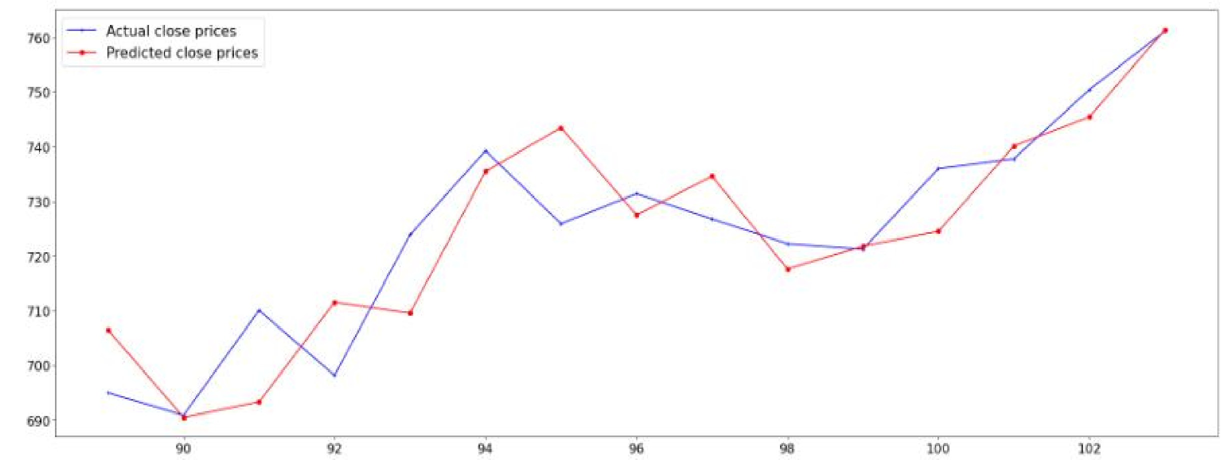


Figure 1: Actual vs Predicted stock price for 15 days graph for the stock prices of Axis Bank.

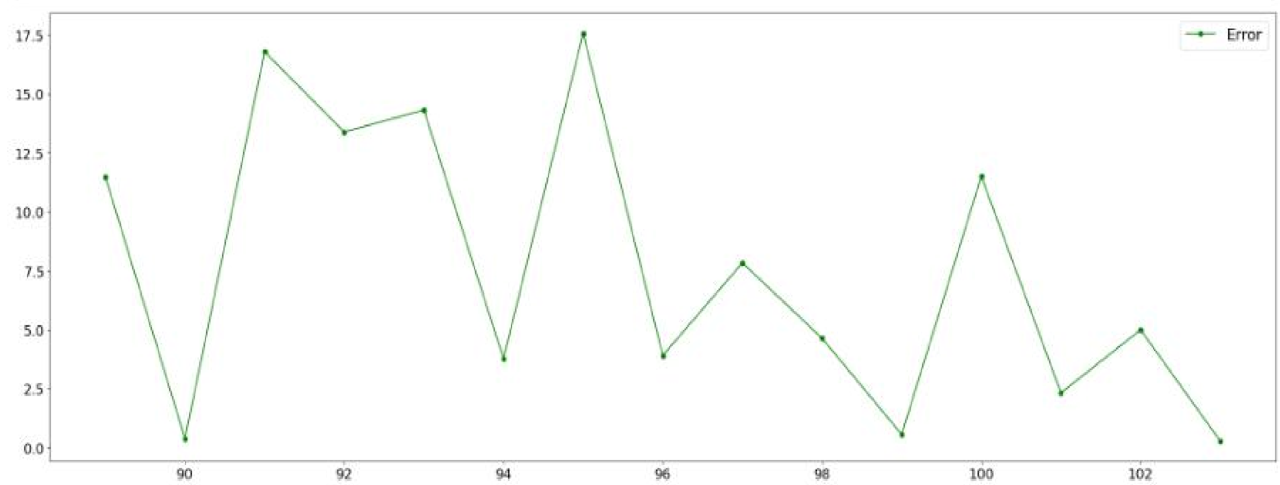


Figure 2: Prediction error for stock price for 15 days graph for the stock prices of Axis Bank.

The maximum error, minimum error and mean error in our prediction model is 17.5, 0.27 and 7.579 respectively.

The above graphs shows that our model is a good model for predicting the stock closing prices as seen by the closeness of the predicted and actual stock prices graph.

The parameters of our model $\lambda = (A, \mu^k, \Sigma^k, \pi)$ are as follows,

$$A = \begin{pmatrix} 0.989 & 0.04 & 0 \\ 0.71 & 0.289 & 0 \\ 0.177 & 0.822 & 0 \end{pmatrix}$$

$$\mu^k = \begin{pmatrix} -0.002 & 0.011 & 0.014 \\ 0.046 & 0.051 & 0.008 \\ 0.049 & 0.054 & 0.009 \end{pmatrix}$$

$$\Sigma^k = \begin{pmatrix} \begin{pmatrix} 0.0003 & 0 & 0 \\ 0 & 0.0002 & 0 \\ 0 & 0 & 0.0002 \end{pmatrix} \\ \begin{pmatrix} 0.0081 & 0 & 0 \\ 0 & 0.0081 & 0 \\ 0 & 0 & 0.0079 \end{pmatrix} \\ \begin{pmatrix} 1000 & 0 & 0 \\ 0 & 1000 & 0 \\ 0 & 0 & 1000 \end{pmatrix} \end{pmatrix}$$

$$\pi = \begin{bmatrix} 0.89 & 0.11 & 0 \end{bmatrix}$$

Appendix

1. Emission matrix is the matrix of probabilities of observing a state given that some HMM has reached some hidden state. Let $o_1, o_2, \dots, o_n$ be the observed states and $h_1, h_2, \dots, h_m$ be the hidden states. Let the probability of observing the state o_i given that the HMM is in hidden state h_j is given by $e_{i,j}$. Then emission probability matrix is given by $E = [e_{i,j}]_{m\times n}$.
2. Steady state probability is defined as probability of a Markov chain being in a state when there is no change in probability of being in that state thereafter. We say chain has reached stationarity. Let π_j denote the steady state probability of state j then it is calculated by solving the following set of equations: $\pi_j = \sum_{i\in\Omega} \pi_j p_{i,j} \;\forall\, j \in \Omega$ and $\sum_{j\in\Omega} \pi_j = 1$ where Ω is the set of states and $p_{i,j}$ is the transition probability.
3. **Derivations:**

   (a) **Forward Algorithm:**

Probability of sequence of observed states in a HMM determined as follows:

For any $n \geq 0$, $P(Y_0 = k_0, Y_1 = k_1, Y_2 = k_2, \dots, Y_n = k_n) = P(\boldsymbol{Y}_{n+1} = \boldsymbol{k}_{n+1})$

$$\Rightarrow P(\boldsymbol{Y}_{n+1} = \boldsymbol{k}_{n+1}) = \sum_{j\in H} P(\boldsymbol{Y}_{n+1} = \boldsymbol{k}_{n+1}, X_n = j)$$

Let us define $F_{n+1}(j) := P(\boldsymbol{Y}_{n+1} = \boldsymbol{k}_{n+1}, X_n = j)$ $\quad \dots$ **Eq. (1)**

Then $P(\boldsymbol{Y}_{n+1} = \boldsymbol{k}_{n+1}) = \sum_{j\in H} F_{n+1}(j)$ $\quad\ldots$ **Eq. (2)**

The required probability can be found out using Eq. (2) if we know $F_{n+1}(j)\ \forall\, j \in H$.

Now,

$F_{n+1}(j) = P(\boldsymbol{Y}_{n+1} = \boldsymbol{k}_{n+1}, X_n = j)$ $\quad$ [from Eq. (1)]

$\Rightarrow F_{n+1}(j) = P(\boldsymbol{Y}_n = \boldsymbol{k}_n, Y_n = k_n, X_n = j) = \sum_{i\in H} P(\boldsymbol{Y}_n = \boldsymbol{k}_n, X_{n-1} = i, Y_n = k_n, X_n = j)$

$$= \sum_{i\in H} P(\boldsymbol{Y}_n = \boldsymbol{k}_n, X_n = j | \boldsymbol{Y}_n = \boldsymbol{k}_n, X_{n-1} = i) P(\boldsymbol{Y}_n = \boldsymbol{k}_n, X_{n-1} = i) \quad [\because P(A) = P(A|B)P(B)]$$

$$= \sum_{i\in H} F_n(i) P(\boldsymbol{Y}_n = \boldsymbol{k}_n, X_n = j | \boldsymbol{Y}_n = \boldsymbol{k}_n, X_{n-1} = i) \qquad [\because F_n(i) = P(\boldsymbol{Y}_n = \boldsymbol{k}_n, X_{n-1} = i)]$$

$= \sum_{i\in H} F_n(i) P(\boldsymbol{Y}_n = \boldsymbol{k}_n | X_n = j) P(X_n = j | X_{n-1} = i)$ [$\because$ Y_n depends on X_n, X_n depends on X_{n-1}]

$$= \sum_{i\in H} F_n(i) e_{j,k_n} p_{i,j}$$

$\Rightarrow F_{n+1}(j) = e_{j,k_n} \sum_{i\in H} F_n(i) p_{i,j}$ $\quad\ldots$ **Eq. (3)**

Thus we see that expression of $F_{n+1}(j)$ is recursive in nature. The base case $F_1(j)$ can be calculated by Eq. (1)

$F_1(j) = P(\boldsymbol{Y}_1 = \boldsymbol{k}_1, X_0 = j) = P(Y_0 = k_0, X_0 = j) = P(Y_0 = k_0 | X_0 = j) P(X_0 = j) = p_j e_{j,k_0}$

Using Eq. (1) we can recursively find $F_2(j), F_3(j), \ldots,$ upto $F_{n+1}(j)\ \forall\, j \in H$. Then using Eq. (2) we can calculate $P(\boldsymbol{Y}_{n+1} = \boldsymbol{k}_{n+1})$.

Since $F_{n+1}(\cdot)$ is calculated using $F_n(\cdot)$, this is known as the *Forward Algorithm*.

(b) **Backward Algorithm:**

There is another approach to find $P(\boldsymbol{Y}_{n+1} = \boldsymbol{k}_{n+1})$; $n \geq 0$

$P(\boldsymbol{Y}_{n+1} = \boldsymbol{k}_{n+1}) = P(Y_0 = k_0, Y_1 = k_1, Y_2 = k_2, \ldots, Y_n = k_n)$

$$= \sum_{i\in H} P(Y_0 = k_0, Y_1 = k_1, Y_2 = k_2, \ldots, Y_n = k_n | X_0 = i) P(X_0 = i)$$

$$= \sum_{i\in H} P(Y_0 = k_0 | X_0 = i) P(Y_1 = k_1, Y_2 = k_2, \ldots, Y_n = k_n | Y_0 = k_0, X_0 = i) p_i$$

Since Y_i depends only on X_i, and X_i is influenced only by latest known state, here X_0, we have

$$P(\boldsymbol{Y}_{n+1} = \boldsymbol{k}_{n+1}) = \sum_{i\in H} p_i e_{ik_0} P(Y_1 = k_1, Y_2 = k_2, \ldots, Y_n = k_n | X_0 = i)$$

$\Rightarrow P(\boldsymbol{Y}_{n+1} = \boldsymbol{k}_{n+1}) = \sum_{i\in H} p_i e_{ik_0} B_0(i)$ $\quad\ldots$ **Eq. (4)**

where $B_0(i) = P(Y_1 = k_1, Y_2 = k_2, \ldots, Y_n = k_n | X_0 = i)$

In general, we define for $0 \leq m < n$

$B_m(i) := P(Y_{m+1} = k_{m+1}, Y_{m+2} = k_{m+2}, \ldots, Y_n = k_n | X_m = i)$ $\quad\ldots$ **Eq. (5)**

$$= \sum_{j\in H} P(Y_{m+1} = k_{m+1}, Y_{m+2} = k_{m+2}, \ldots, Y_n = k_n | X_{m+1} = j, X_m = i) P(X_{m+1} = j, X_m = i)$$

$= \sum_{j\in H} P(Y_{m+1} = k_{m+1}, Y_{m+2} = k_{m+2}, \dots, Y_n = k_n | X_{m+1} = j) p_{i,j}$ [$\because$ latest state X_{m+1} is known]

$= \sum_{j\in H} p_{i,j} P(Y_{m+1} = k_{m+1} | X_{m+1} = j) P(Y_{m+2} = k_{m+2}, \dots, Y_n = k_n | Y_{m+1} = k_{m+1}, X_{m+1} = j)$

Since X_{m+1} is known, knowing Y_{m+1} is redundant in probability of further observed states, so,

$$B_m(i) = \sum_{j\in H} p_{i,j} e_{j,k_{m+1}} P(Y_{m+2} = k_{m+2}, \dots, Y_n = k_n | X_{m+1} = j)$$

$$\Rightarrow B_m(i) = \sum_{j\in H} p_{i,j} e_{j,k_{m+1}} B_{m+1}(j) \qquad \dots \textbf{Eq.}\,(\mathbf{6})$$

Thus we see that expression of $B_m(i)$ is recursive in nature. The base case here is $B_{n-1}(i)$ and can be calculated by Eq. (5)

$$B_{n-1}(i) = P(Y_n = k_n | X_{n-1} = i) = \sum_{j\in H} P(Y_n = k_n | X_n = j, X_{n-1} = i) P(X_n = j | X_{n-1} = i)$$

$\Rightarrow B_{n-1}(i) = \sum_{j\in H} P(Y_n = k_n | X_n = j) p_{i,j}$ [$\because X_n$ is known so X_{n-1} is redundant]

$$\Rightarrow B_{n-1}(i) = \sum_{j\in H} p_{i,j} e_{j,k_n}$$

Using Eq. (6) we can recursively find $B_{n-2}(i), B_{n-3}(i), \dots,$ upto $B_0(i)\ \forall\, i \in H$. Then using Eq. (4) we can calculate $P(\boldsymbol{Y}_{n+1} = \boldsymbol{k}_{n+1})$.

Since $B_m(\cdot)$ is calculated using $B_{m+1}(\cdot)$, this is known as the *Backward Algorithm*.

(c) **Forward-Backward Algorithm:**

There is yet another approach to find $P(\boldsymbol{Y}_{n+1} = \boldsymbol{k}_{n+1})$ which involves both Forward and Backward algorithms.

Forward approach: $P(\boldsymbol{Y}_{n+1} = \boldsymbol{k}_{n+1}) = \sum_{j\in H} P(\boldsymbol{Y}_{n+1} = \boldsymbol{k}_{n+1}, X_n = j) = \sum_{j\in H} F_{n+1}(j)$

Backward approach: $P(\boldsymbol{Y}_{n+1} = \boldsymbol{k}_{n+1}) = \sum_{j\in H} P(\boldsymbol{Y}_{n+1} = \boldsymbol{k}_{n+1}, X_0 = j) = \sum_{j\in H} p_j e_{jk_0} B_0(j)$

Forward-Backward approach: $P(\boldsymbol{Y}_{n+1} = \boldsymbol{k}_{n+1}) = \sum_{j\in H} P(\boldsymbol{Y}_{n+1} = \boldsymbol{k}_{n+1}, X_m = j)\,; 1 \le m < n$

$$\Rightarrow P(\boldsymbol{Y}_{n+1} = \boldsymbol{k}_{n+1}) = \sum_{j\in H} P(\boldsymbol{Y}_{m+1} = \boldsymbol{k}_{m+1}, Y_{m+1} = k_{m+1}, \dots, Y_n = k_n, X_m = j)$$

$$\Rightarrow P(\boldsymbol{Y}_{n+1} = \boldsymbol{k}_{n+1}) = \sum_{j\in H} P(\boldsymbol{Y}_{m+1} = \boldsymbol{k}_{m+1}, X_m = j) P(Y_{m+1} = k_{m+1}, \dots, Y_n = k_n | \boldsymbol{Y}_{m+1} = \boldsymbol{k}_{m+1}, X_m = j)$$

Since latest hidden state X_m is known, knowing before observed states is irrelevant in probability of further observed states so we have

$$\Rightarrow P(\boldsymbol{Y}_{n+1} = \boldsymbol{k}_{n+1}) = \sum_{j\in H} F_{m+1}(j) P(Y_{m+1} = k_{m+1}, \dots, Y_n = k_n | X_m = j)$$

$$\Rightarrow P(\boldsymbol{Y}_{n+1} = \boldsymbol{k}_{n+1}) = \sum_{j\in H} F_{m+1}(j) B_m(j) \text{ for any } m, 1 \le m < n \qquad \dots \textbf{Eq.}\,(\mathbf{7})$$

We see that $P(\boldsymbol{Y}_{n+1} = \boldsymbol{k}_{n+1})$ is a sum of product of $F_{m+1}(j)$ and $B_m(j)$ over $j \in H$.

One can start finding forward sequence $F_1(j), F_2(j), \ldots$ by Eq. (3) and backward sequence $B_{n-1}(j), B_{n-2}(j), \ldots$ by Eq. (6) simultaneously and stop when $F_{m+1}(j)$ and $B_m(j)$ is known for some m, $1 \leq m < n$ and for all $j \in H$.

In fact, we can find one m, at 1^{st} step, we calculated $F_1(j)$ and $B_{n-1}(j)$, at 2^{nd} step we calculated $F_2(j)$ and $B_{n-2}(j)$ and so on at k^{th} step we calculated $F_k(j)$ and $B_{n-k}(j)$, calculation goes until $k = m+1, n-k = m \Rightarrow k-(n-k) = m+1-m \Rightarrow 2k = n+1 \Rightarrow k = \lceil \frac{n+1}{2} \rceil$ since k is a step. So $m = k-1 = \lceil \frac{n+1}{2} \rceil - 1 = \lceil \frac{n-1}{2} \rceil$.

$\Rightarrow P(\boldsymbol{Y}_{n+1} = \boldsymbol{k}_{n+1}) = \sum_{j \in H} F_k(j) B_{k-1}(j)$ where $k = \lceil \frac{n+1}{2} \rceil, n = 0,1,2, \ldots$ Note that $\lceil \cdot \rceil$ is a ceiling function operator.

(d) **Viterbi Algorithm:**

Most likely sequence of hidden states which induces the given sequence of observed states is determined as follows:

For any $n \geq 0$, required sequence of hidden states $\boldsymbol{j}_{n+1} = (j_0, j_1, j_2, \ldots, j_n)$ is the sequence that maximizes $P(X_0 = j_0, X_1 = j_1, \ldots, X_n = j_n | Y_0 = k_0, Y_1 = k_1, \ldots, Y_n = k_n)$

or $P(\boldsymbol{X}_{n+1} = \boldsymbol{j}_{n+1} | \boldsymbol{Y}_{n+1} = \boldsymbol{k}_{n+1})$

Now,

$$P(\boldsymbol{X}_{n+1} = \boldsymbol{j}_{n+1} | \boldsymbol{Y}_{n+1} = \boldsymbol{k}_{n+1}) = \frac{P(\boldsymbol{X}_{n+1} = \boldsymbol{j}_{n+1}, \boldsymbol{Y}_{n+1} = \boldsymbol{k}_{n+1})}{P(\boldsymbol{Y}_{n+1} = \boldsymbol{k}_{n+1})} \qquad \ldots \textbf{Eq. (8)}$$

Since denominator is independent of $\boldsymbol{j}_{n+1}$, our problem becomes to maximize the numerator over $\boldsymbol{j}_{n+1}$, i.e.

$$\max_{j_0, j_1, \cdots, j_n} P\left(X_0 = j_0, X_1 = j_1, \ldots, X_n = j_n, Y_0 = k_0, Y_1 = k_1, \ldots, Y_n = k_n\right) \qquad \ldots \textbf{Eq.(9)}$$

To solve this problem, for $0 \leq m \leq n$, let

$$V_m(j) = \max_{j_0, j_1, \ldots, j_{m-1}} P\left(X_0 = j_0, X_1 = j_1, \ldots, X_{m-1} = j_{m-1}, X_m = j, \boldsymbol{Y}_{m+1} = \boldsymbol{k}_{m+1}\right) \qquad \ldots \textbf{Eq.(10)}$$

Then our problem becomes $\max_j V_n(j)$

Let us solve recursively for $V_m(j)$, $0 \leq m \leq n$,

$$V_m(j) = \max_i \max_{\mathbf{j}_{m-1}} P\left(\boldsymbol{X}_{m-1} = \mathbf{j}_{m-1}, X_{m-1} = i, X_m = j, \boldsymbol{Y}_{m+1} = \boldsymbol{k}_{m+1}\right)$$

$$= \max_i \max_{\mathbf{j}_{m-1}} P\left(\boldsymbol{X}_{m-1} = \mathbf{j}_{m-1}, X_{m-1} = i, \boldsymbol{Y}_m = \boldsymbol{k}_m, X_m = j, Y_m = k_m\right)$$

$$= \max_i \max_{\mathbf{j}_{m-1}} \left\{ \begin{array}{l} P\left(\boldsymbol{X}_{m-1} = \mathbf{j}_{m-1}, X_{m-1} = i, \boldsymbol{Y}_m = \boldsymbol{k}_m\right) \\ \times P\left(X_m = j, Y_m = k_m | \boldsymbol{X}_{m-1} = \mathbf{j}_{m-1}, X_{m-1} = i, \boldsymbol{Y}_m = \boldsymbol{k}_m\right) \end{array} \right\}$$

$$= \max_i \max_{\mathbf{j}_{m-1}} \left\{ P\left(\boldsymbol{X}_{m-1} = \mathbf{j}_{m-1}, X_{m-1} = i, \boldsymbol{Y}_m = \boldsymbol{k}_m\right) \times P\left(X_m = j, Y_m = k_m | X_{m-1} = i\right) \right\}$$

$$=\max_{i} P\left(X_m=j, Y_m=k_m \middle| X_{m-1}=i\right) \times \max_{\mathbf{j}_{m-1}} P\left(\boldsymbol{X}_{m-1}=\mathbf{j}_{m-1}, X_{m-1}=i, \boldsymbol{Y}_m=\boldsymbol{k}_m\right)$$

$$=\max_{i} p_{i,j}\, e_{j,k_m} V_{m-1}(i)$$

$$\Rightarrow V_m(j) = e_{j,k_m} \max_{i} p_{i,j}\, V_{m-1}(i) \qquad \ldots\textbf{Eq.(11)}$$

The base case of this recursive sequence is $V_0(j)$ and can be calculated using Eq. (10),

$$V_0(j) = P\left(X_0=j, Y_1=k_1\right)$$

$$=P\left(X_0=j, Y_0=k_0\right) = P\left(Y_0=k_0 \middle| X_0=j\right) P\left(X_0=j\right)$$

$$= e_{j,k_0} p_j$$

Using Eq. (11) we can recursively find $V_1(j), V_2(j),\ldots,$ upto $V_n(j)\ \forall\, j \in H$.

To get the sequence of hidden states that maximizes the probability of observing $\boldsymbol{k}_{n+1}$ as observed states, we work in the reverse direction to find optimal solution to maximize $V_n(j)$. Let j_n^* be the value of j that maximizes $V_n(j)$. Then j_n^* is the last state of sequence $\boldsymbol{j}_{n+1}$. Next we find the next to last known state by the following algorithm. Let for $m < n$, $i_m^*(j)$ be the value of i that maximizes $p_{i,j}V_m(i)$ then

$$\max_{i_1, i_2, \cdots, i_n} P\left(\boldsymbol{X}_{n+1}=\mathbf{i}_{n+1} \middle| \boldsymbol{Y}_{n+1}=\boldsymbol{k}_{n+1}\right) = \max_{j} V_n(j)$$

$$=V_n\left(j_n^*\right)$$

$$= \max_{i_1, \cdots, i_{n-1}} P\left(\boldsymbol{X}_n=\mathbf{i}_n, X_n=j_n^* \middle| \boldsymbol{Y}_{n+1}=\boldsymbol{k}_{n+1}\right)$$

$$=e_{j_n^*, k_n} \max_{i} p_{i, j_n^*} V_{n-1}(i)$$

$$=e_{j_n^*, k_n} p_{i_{n-1}^*\left(j_n^*\right), j_n^*} V_{n-1}\left(i_{n-1}^*\left(j_n^*\right)\right)$$

Thus $i_{n-1}^*(j_n^*)$ is next to last state of sequence $\boldsymbol{j}_{n+1}$. Continuing in the same manner we can find all the states of the sequence $\boldsymbol{j}_{n+1}$ and thus we found the most likely sequence of hidden states that induces the given sequence of observed states. This algorithm is known as the *Viterbi Algorithm.*

4. To evaluate our model we have used the Mean Absolute Percentage Error (*MAPE*) as a metric(*see reference 2*).

$$MAPE = \left(\frac{1}{n}\sum_{i=1}^{n}\frac{\left|p_i - a_i\right|}{a_i}\right)$$

p_i = predicted stock price for the ith day

a_i = actual stock price for the ith day

References

1. Introduction to Probability Models by Sheldon M. Ross.
2. Stock Market Prediction Using Hidden Markov Models by Aditya Gupta and Bhuwan Dingra, Non-Students member, IEEE.
3. Internet webpage - medium.com