Stochastic Processes in Finance

STOCK PRICE MODELING USING MARKOV CHAINS

HARSHAL DUPARE

18MA20015

OBJECTIVE

- To model the stock price using Markov chain and implement them.
- To study the significance of various parameters used in the model and understand how they relate to the various aspects of stock price.
- To study the differences in various models and how they affect the final predictions.

Stochastic processes used to model stock price :

- Discrete time & discrete space Markov chain
    1. Model without memory
    2. Model with memory
- Discrete time & continuous space Markov chain
    1. Model without memory
    2. Model with memory

Specifications of data used in models

- Data is in form of array with arbitrary length, sufficient for the algorithm to work.
- The data points are spaced at uniform intervals of time.

Output format

- Modified array with new predicted values concatenated.
- Plotting the total range.

DATA & MODEL

1. Discrete time & discrete space Markov chain without memory

- Stock price is modeled as discrete values represented as state space of markov chain.
- All the consecutive states are at constant interval of time.
- The number of possible states in MC are dependent on the size of states and range of the stock price.
- Each state has a set of transition probabilities from it to other states, represented by that row in TPM (Transition Probability Matrix).
- We prefer to store TPM by the jump size rather than the next state, this is for the purposes of implementation.
- Each predicted value is a discrete state which may or may not be among the states of Markov chain.

m_s = *state of markov chain for stock price " s "*

s = *value of the stock*

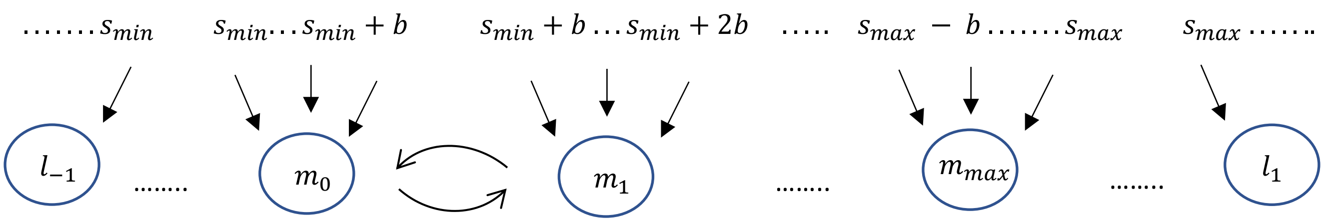
b = *bucket size of the states*

$[\,]$ = *nearest integer function*

l_i = *state not in markov chain*

$$m_s = \left[\frac{s}{b}\right]$$

$$S_p = b \times m_s + \frac{b}{2}$$

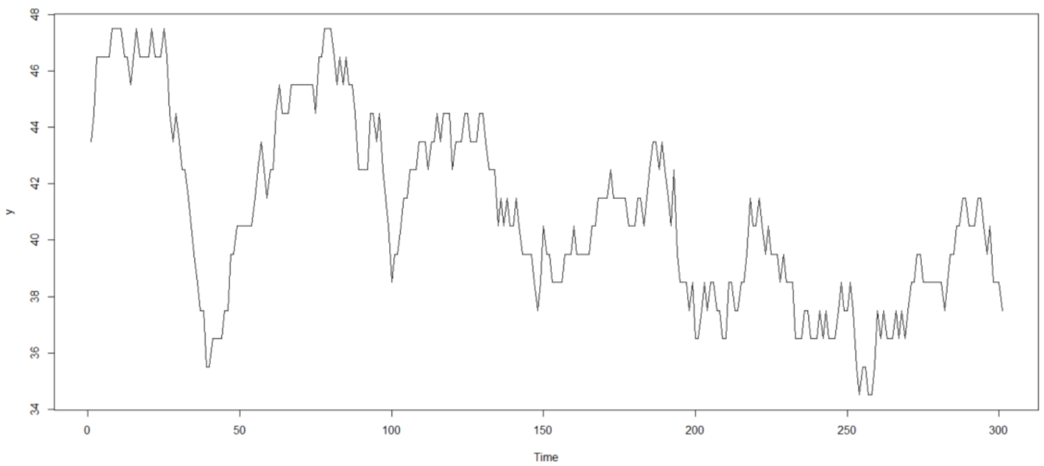


Steps :

1. Data on stock price is translated to new array variable containing its markov chain states, using above formula for " m_s ".
2. TPM is constructed using one pass of for loop over the states array by accounting for the size of transition for consecutive states.
3. Each new predicted state is calculated by a random variable with distribution of probabilities according to that row of the TPM.
4. Predicted states are then translated to stock prices using above formula for " S_p ".
5. For a state which doesn't belong to the State Space of Markov chain, no assumption is made about the transition probabilities, instead a global distribution is obtained which is used for transition probabilities.

Characteristics of model :

1. Each new state derives only from the most recent state, hence predictions are memoryless.
2. Forecast can only take discrete values.
3. Prediction size cannot be big compared to data size or otherwise errors due to lack of data make the predictions diverge and behave randomly.

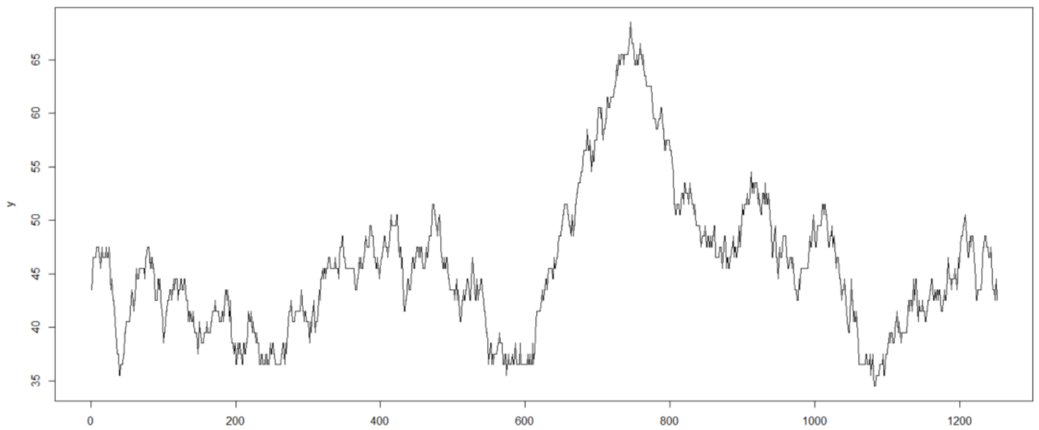


Data size : 251

Forecast size : 50

Negative trend is observed.

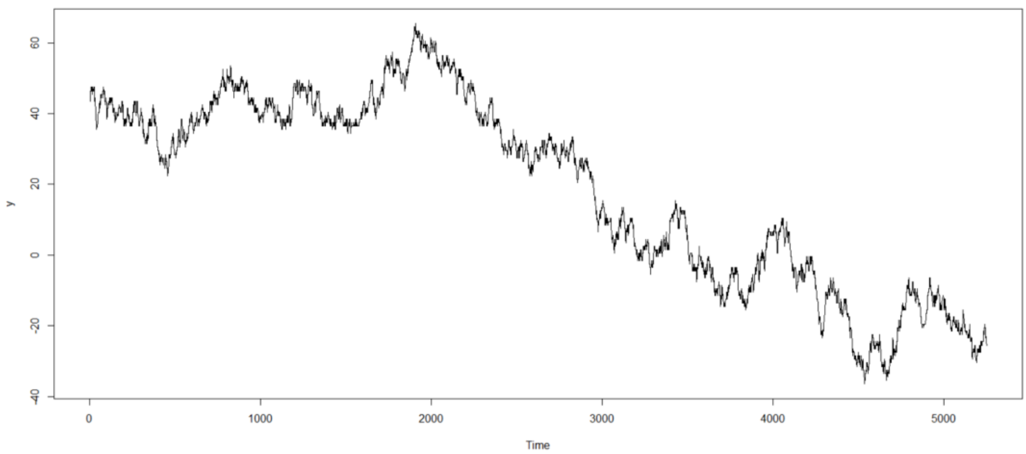
Only discrete values are there.



Data size : 251

Forecast size : 1000

Random behavior due to extreme size.



Data size : 251

Forecast size : 5000

Highly deviating behavior due to extreme size which amplify the small negative trend.

2. Discrete time & Discrete space Markov chain with memory

- Stock price is modeled as discrete values represented as value of random variable, which depends on the state of markov chain.
- The Markov Chain used here has states as a sequence with specific length (the memory) of high and low in stock price.
- For implementation purposes each state of Markov chain is translated to a binary number as " 1 " as high and " 0 " as low. Then that binary number is converted to decimal and that is used as state of Markov chain.
- We note that from each state there are only 2 possible states that can be reached i.e. for example from state with memory " 3 ", " 001 " we can only reach state " 010 " or " 011 " with some probabilities.
- For the states which we have no data, we make no assumptions and assign them global values of high and low probabilities.
- Then for each state we calculate the distribution of jump size in stock value
- For forecast we calculate the state and converting that state to decision if stock value moves high or low is made based on calculated probability.
- Then for each state there is certain size jump in stock value with calculated distribution, using that we calculate the predicted stock value.

$memory = p$

$base = (1, 2, 2^2 \dots 2^{p-1})$

$D_i = stores\ last$ " p " $direction\ the\ stock\ moved\ in$

$e.g.\ D_i = (1,0.\dots.1,0)$

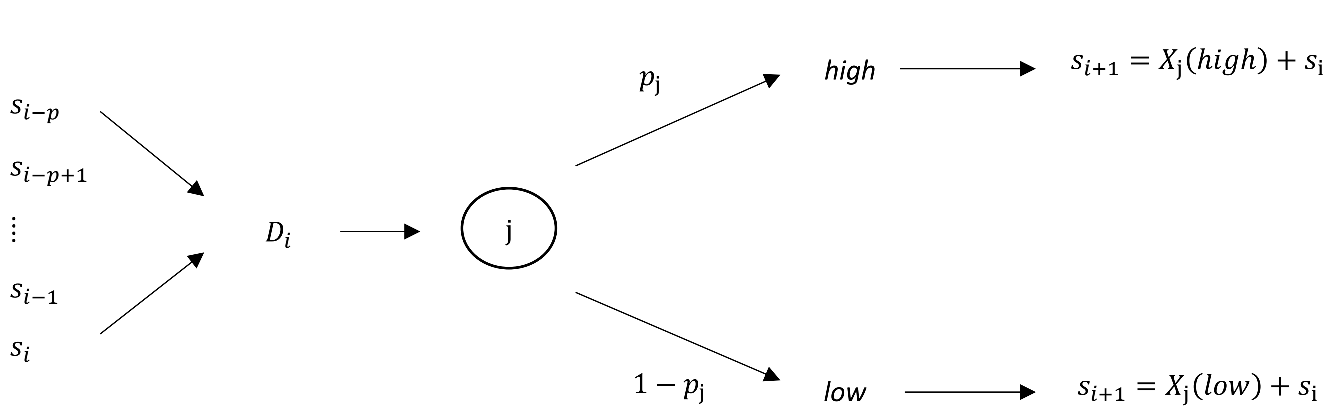
$\mathrm{j} = state\ of\ markov\ chain$

$\mathrm{j} = D_i \cdot \mathrm{base}$

$s = stock\ price$

$X_i(\cdot) = \text{Discrete } random\ variable\ generated\ from\ i'th\ high\ or\ low\ distribution$

$s_{i+1} = X_i(\cdot) + s_i$

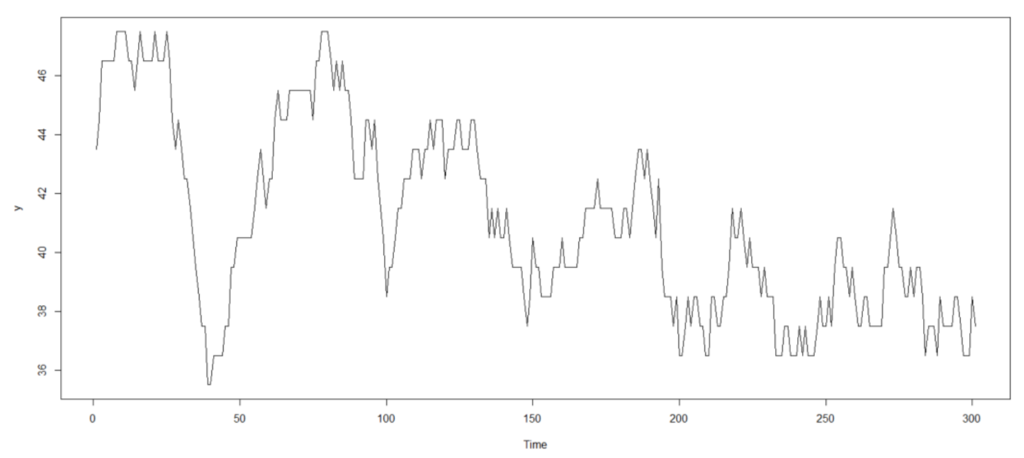


Steps :

1. Data on stock price is translated to new array variable containing its discretized value " s_i ".
2. TPM is constructed using one pass of for loop over the discretized array by accounting for the size of transition for consecutive states.
3. Arrays for high and low probabilities are constructed by another pass of for loop.
4. Global probabilities are also constructed along the high and low probabilities and assigned to the states which we have no information about.
5. For forecast we calculate the state, then based on its high and low probabilities and distribution from TPM we calculate new discretized " s_{i+1} ".

Characteristics of model :

1. Each new value derives only from the last " p + 1 " values , hence prediction is termed as having memory.
2. Forecast can only take discrete values.
3. We observe that there is an optimal value of " p ". Too small choice of " p " leads to lot less consideration of variance, and too high values of " p " leads too much of computation time with no significant improvements or deviation from the trend.
4. Prediction size cannot be big compared to data size as otherwise errors due to lack of data make the predictions diverge and behave randomly.



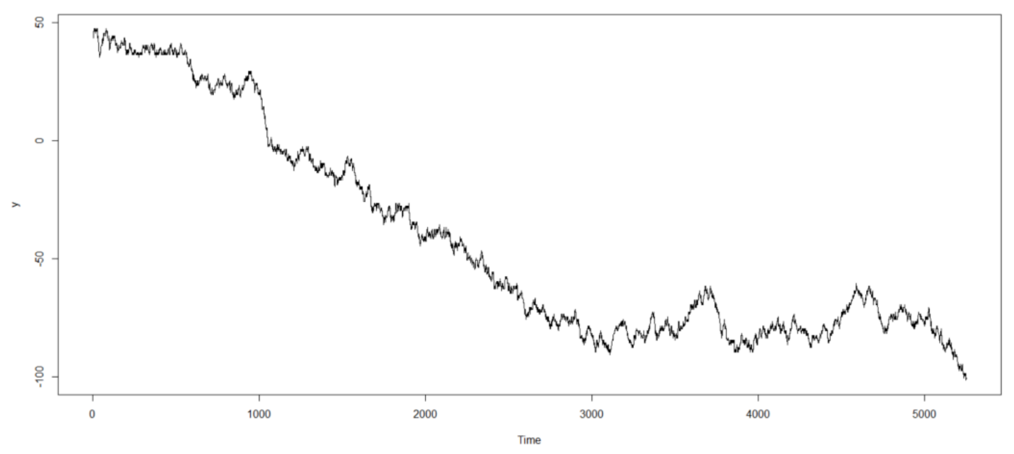
Data size : 251

Forecast size : 50

Memory : 3

Negative trend is observed.

Only discrete values are there.



Data size : 251

Forecast size : 5000

Highly deviating behavior due to extreme size which amplify the small negative trend.

3. Discrete time & continuous space Markov chain without memory

- Stock price is modeled as continuous value represented as value of random variable, which depends on the state of markov chain with discrete state.
- The Markov Chain used here has states as a range of stock price with specific size (the bucket-size).
- For each state there is specific probability of going high (i.e. increase in stock price) or low (i.e. decrease in stock price).
- For each high or low depending on the range in which stock price falls there is a distribution on size of high or low.
- i.e. we assume " $s_{i+1} - s_i \sim \mathrm{X}_{\mathrm{b_i}}$ " where " $\mathrm{b_i} = [\, s_i/\mathrm{B}\,]$ " meaning increment to stock price is decided by the distribution of that bucket.
- We assume that all increments are Normally distributed with some means and standard deviation.
- For the states which we have no data we make no assumptions and assign them global values of mean and standard deviation.
- For forecast we calculate the state and decide if stock value moves high or low based on calculated probabilities.
- Then for each state there is certain size of increment in stock value with calculated distribution, using that we calculate the predicted stock value.

$B = bucket\ size$

$\mathrm{j} = state\ of\ markov\ chain\ at\ i'th\ time$

$\mathrm{s_i} = stock\ price\ at\ i'th\ time$

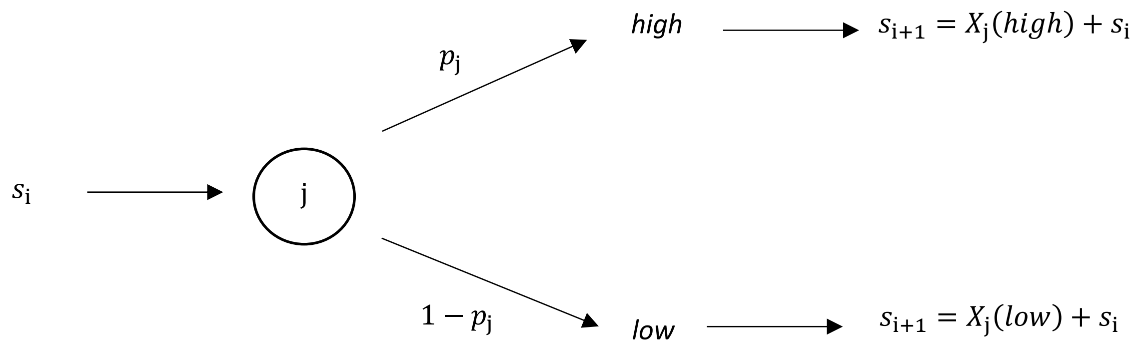
$[\,] = nearest\ integer\ function$

$\mathrm{j} = [\,{}^{s_\mathrm{j}}/_{\mathrm{B}}\,]$

$s = stock\ price$

$X_i(\cdot) = \text{continuous}\ random\ variable\ generated\ from\ i'th\ high\ or\ low\ distribution$

$\mathrm{s_{i+1}} = \mathrm{X}_i(\cdot) + s_\mathrm{i}$

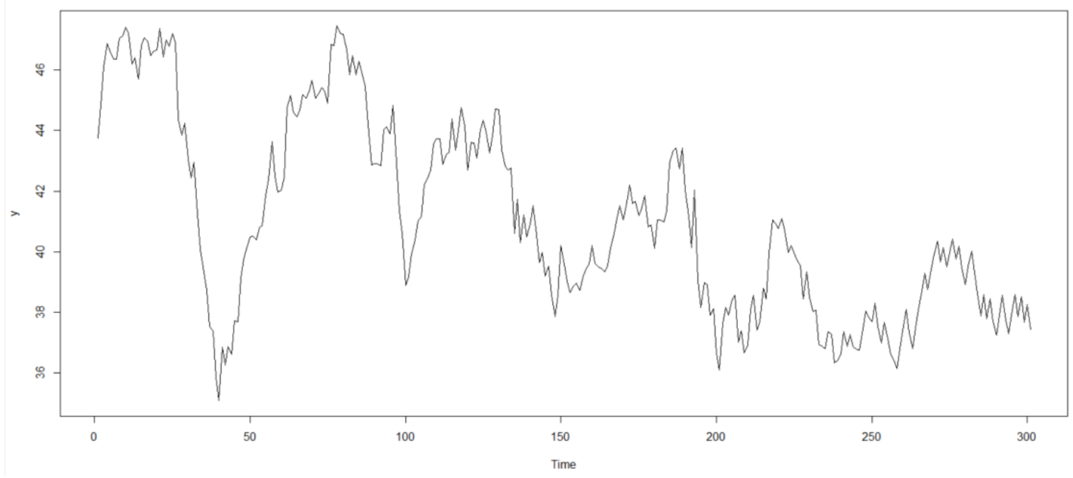


Steps :

1. Arrays for high and low probabilities are constructed, along with the array with information of mean and standard deviation by one pass of for loop.
2. Global probabilities, mean and standard deviation are also calculated along the high and low probabilities and assigned to the states which we have no information about.
3. For forecast we calculate the state, then based on its high and low probabilities and values of mean and standard deviation calculate " S_{i+1} ".

Characteristics of model :

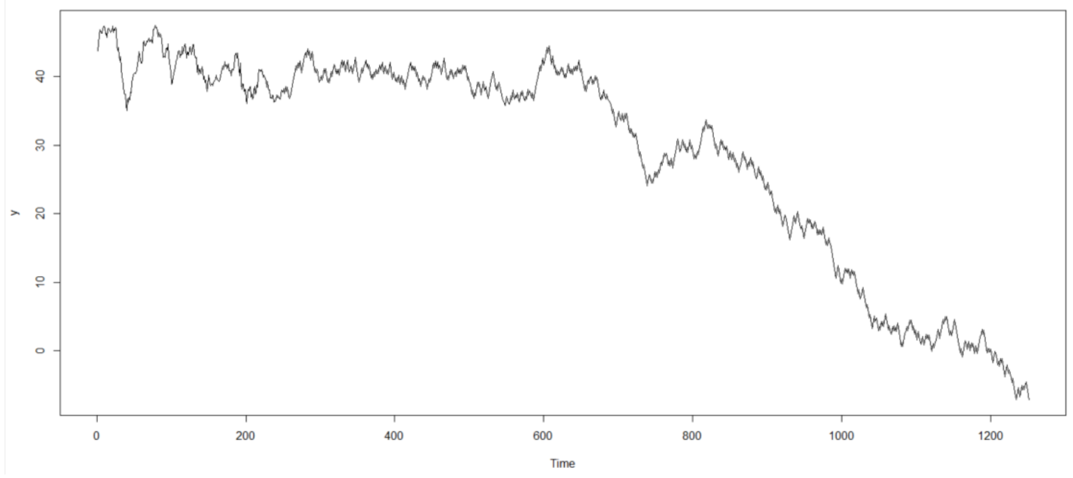
1. Each new state derives only from the most recent state, hence predictions are memoryless.
2. Forecast can take continuous values.
3. Prediction size cannot be big compared to data size as otherwise errors due to lack of data make the predictions diverge and behave randomly.



Data size : 251

Forecast size : 50

Negative trend with continuous values.



Data size : 251

Forecast size : 1000

Highly deviating behavior due to extreme size which amplify the small negative trend.

4. Discrete time & continuous space Markov chain with memory

- Stock price is modeled as continuous value represented as value of random variable, which depends on the state of markov chain with discrete state.
- The Markov Chain used here has states as a sequence with specific length (the memory) of high and low in stock price.
- For implementation purposes each state of Markov chain is translated to a binary number as “ 1 ” as high and “ 0 ” as low. Then that binary number is converted to decimal and that is used as state of Markov chain.
- We note that from each state there are only 2 possible states that can be reached i.e. for example from state with memory “ 3 ”, “ 001 ” we can only reach state “ 010 ” or “ 011 ” with some probabilities.
- We assume " $s_{i+1} - s_i \sim \mathrm{X}_{\mathrm{n_i}}$ " where " $\mathrm{n_i} = \mathrm{D_i} \cdot \mathrm{base}$ " meaning increment to stock price is decided by the distribution of that state.
- We assume that all increments are Normally distributed with some means and standard deviation.
- For the states which we have no data on we make no assumptions and assign them global values of mean and standard deviation.
- For forecast we calculate the state and decide if stock value moves high or low based on calculated probabilities.
- Then for each state there is certain size of increment in stock value with calculated distribution, using that we calculate the predicted stock value.

$memory \;=\; p$

$base = (1, 2, 2^2 \ldots 2^{p-1})$

$D_i = stores\ last\ \text{" p "}\ direction\ the\ stock\ moved\ in$

$e.g.\ D_i \;=\; (1,0.\ldots..1,0)$

$\mathrm{j} \;= state\ of\ markov\ chain$

$\mathrm{j} = D_i \cdot \mathrm{base}$

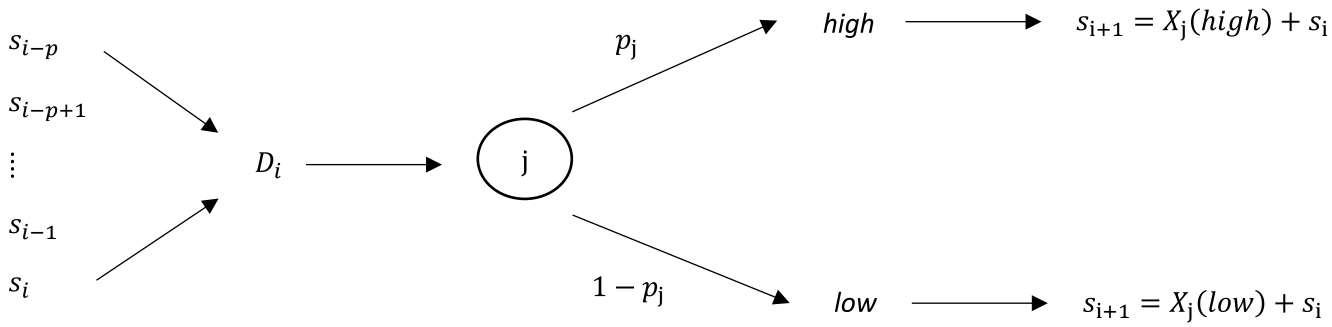
$s \;=\; stock\ price$

$\mathrm{X}_i(\cdot) = \text{continuous } random\ variable$

$generated\ from$

$i'th\ high\ or\ low\ distribution$

$\mathrm{s_{i+1}} = \mathrm{X}_i(\cdot) + s_{\mathrm{i}}$



Steps :

1. Arrays for high and low probabilities are constructed, along with the array with information of mean and standard deviation by one pass of for loop.
2. Global probabilities, mean and standard deviation are also calculated along the high and low probabilities and assigned to the states which we have no information about.
3. For forecast we calculate the state, then based on its high and low probabilities and values of mean and standard deviation calculate " s_{i+1} ".

Characteristics of model :

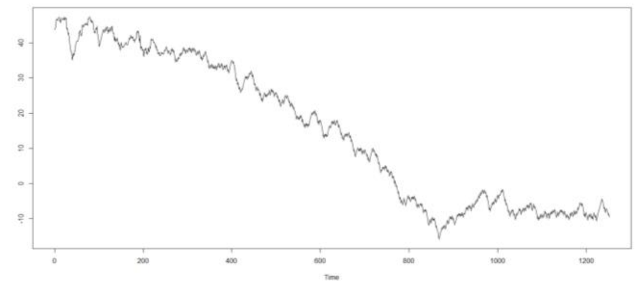
1. Each new value derives only from the last “ p + 1 ” values , hence prediction is termed as having memory.
2. Forecast can take continuous values.
3. We observe that there is an optimal value of “ p ”. Too small choice of “ p ” leads to lot less consideration of variance, and too high values of “ p ” leads too much of computation time with no significant improvements or deviation from the trend.
4. Prediction size cannot be big compared to data size as otherwise errors due to lack of data make the predictions diverge and behave randomly.

Data size : 251

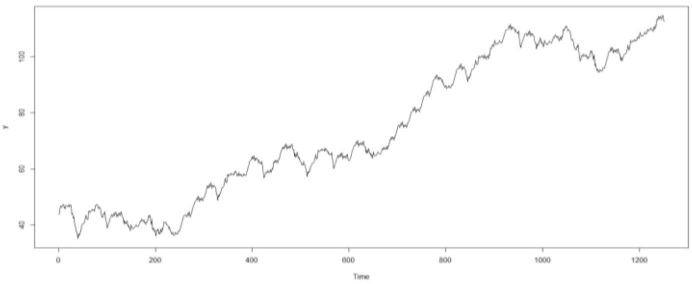
Forecast size : 50

memory : 5

Negative trend with continuous values.



memory = 4



memory = 10

We observe that due to very high values of memory, trend is changed where as it performs best at optimal value of memory which is somewhat at 4-5

SUMMARY

By observing above four models for stock prices forecast we can summarize the following points.

- Markov chain models perform reasonably well for short range of predictions and are able to identify the trends.
- They deviate from the trend as the prediction size increases and behave randomly.
- Between discrete and continuous state model, continuous state model seem to be more volatile and capture more variance.
- Discrete state models seem to recognize the trend in better way as they are not sensitive to small fluctuations.
- When it comes to memory and dependency of stock price on past values, we observe that there is optimal size of memory which gives best results for markov chain model.