STOCHASTIC MODELS ASSIGNMENT

MCMC SIMULATION: NIFTY50 INDEX PRICE PROJECTION

Submitted to: Dr Rupel Nargunam

Assistant Professor,

Madras School of Economics

By:

Athul Joby, AE/2023-25/007, Documentation

Harish Chander, EE/2023-25/011, Documentation

Ritwiz Sarma, GE/2023-25/025, Coding

Samarth Bhatnagar, FE/2023-25/023, Coding

Introduction

The Nifty 50 is the National Stock Exchange of India's main broad-based stock market index for the Indian equity market. It is the weighted average of 50 major Indian companies listed on the NSE, spanning across different sectors of the economy. The Nifty 50 serves as a common gauge for both the Indian stock market and broader economic well-being.

Using Markov chain simulation is very beneficial for studying and forecasting the Nifty 50 index's behavior due to various factors.

The Nifty 50's fluctuations can be classified into distinct states according to the relatively close price value, which correspond effectively with Markov chain's state-oriented method.

Analysis of Nifty 50's past data shows a time series pattern, with Markov chains being a useful tool for modeling sequential data. Markov chain models can aid in evaluating the risk and volatility of the index by simulating various possible paths and help making strategic decisions. These simulations have the potential to uncover concealed patterns in the index's behavior that may not be easily noticeable through basic observation.

Literature review

Monte Carlo simulation is a prominent method used for predicting stock market behavior, including indices like the Nifty 50. This literature review examines similar research studies that have applied Monte Carlo simulations to stock indices and discusses their methodologies and findings.

Singh (2021) applied Monte Carlo simulations to predict short-term price movements of the NIFTY 50 index from 2011 to 2021, focusing on data preprocessing and implementation techniques. Divyank (n.d.) combined ARIMA and Monte Carlo methods to forecast NIFTY 50 prices for a 10-year period, using monthly data from 2003 to 2021. Bolia & Juneja (2005) provided a comprehensive review of Monte Carlo methods for financial option pricing, offering theoretical foundations applicable to index prediction.

A study in the Global Journal of Business and Integral Security examined Monte Carlo simulation for forecasting stock prices of NSE-listed companies, with potential relevance to NIFTY 50 prediction.

These studies collectively demonstrate the application of Monte Carlo techniques in Indian stock market prediction, covering both theoretical aspects and practical implementations for the NIFTY 50 index and related securities. Research on the Nifty 50 index has utilized Monte Carlo simulations to predict future price movements. For instance, a study focused on short-term price movement prediction for the Nifty 50 index used Monte Carlo simulations to estimate the deviation and close price ranges, demonstrating the method's effectiveness in capturing the index's volatility. Another study combined Monte Carlo simulations with ARIMA models to forecast the Nifty 50 index's future prices, showcasing the integration of different predictive techniques to enhance accuracy.

Research has also focused on improving the accuracy of Monte Carlo simulations by adjusting key parameters. A study conducted at the KTH Royal Institute of Technology explored how using weighted means and standard deviations could enhance the accuracy of Monte Carlo simulations for stock prices. This study found that by refining these parameters, the accuracy of predictions improved significantly, from 30% to 50% at the lowest accuracy level and from 70% to 90% at the highest.

Monte Carlo simulations have also been applied in the context of portfolio optimization and risk management. Studies have demonstrated the utility of these simulations in creating risk-adjusted portfolios by modeling the probability of different financial outcomes. This approach allows investors to optimize their portfolios by considering a range of potential market scenarios, thereby minimizing risk and maximizing returns.

Methodology

The data used for analysis was sourced from the package available in Python, *yfinance,* which had details of the Opening, Closing, Lowest, Volume of prices, dividends and stock splits spanning from October 2007 to September 2024, with a daily frequency of the NIFTY price index in which our target is to simulate on the Closing prices of the Nifty price index. The methodology adopted for the MCMC simulation by finding the **Monthly averages** and **Monthly standard deviations** of each observation.

Identifying State Space:

We classified the closing price into six states by modelling into the following methodology:

- **Beyond 2**: The Closing price being two standard deviations above the moving average.
- **Between 1 and 2**: The Closing price being Between one and two standard deviations above the moving average.
- **Between 0 and 1**: The Closing price being under one standard deviation above the moving average.
- **Between 0 and -1**: The Closing price being under one standard deviation below the moving average.
- **Between -1 and -2**: The Closing price being between one and two standard deviations below the moving average.
- **Beyond 2**: The Closing price being over two standard deviations above the moving average.

3.1 Prior Distribution:

The Bayesian statistics states that a prior distribution reflects your initial understanding or assumptions about a parameter or state before any data is observed. By assigning a probability of 1 to the initial state (signifying complete certainty) and 0 to all other states (indicating a lack of information), you clearly define your prior beliefs. This forms the foundation for Bayesian inference, where the incorporation of observed data helps update and refine these beliefs, leading to more informed decisions and conclusions.

For our Simulation, the graph below explains that the Prior Distribution is set at the state of **Beyond 2** with the full certainty of that event and while the other states are assigned with the probability of 0.

This prior distribution forms the foundational belief system for our Bayesian analysis, particularly when simulating future states of the normalized closing price of the Nifty price index based on current observations, where the current state is established through empirical data.

3.2 Transition Probabilities-:

Transition probabilities are a crucial element in probability theory and Markov chain modeling, being essential in this study. The observational data, denoted as $x_1, x_2, ..., x$-------$_N$, is key to our analysis. n_{ij} is defined as the number of transitions seen from state i to state j, where it records when $x_t = i$ and $x_{t+1} = j$ for every t ($1 \leq t \leq N - 1$). Moreover, the quantity of changes from state i, labelled as n_{ij}, indicates the occurrences of $x_t = i$ for every ($1 \leq t \leq N - 1$). The calculation of transition probabilities, p_{ij}, is determined using the following equation:

$$\hat{p}_{ij} = \frac{number\ of\ transitions\ from\ state\ i\ to\ j}{number\ of\ transitions\ from\ state\ i} = n_{ij} / n_i$$

3.3 Transition Probabilities- Matrix:

The transition matrix is a key tool in modelling state transitions over time, capturing the probabilities of moving between states in a dynamic system. Typically structured as a square matrix, it condenses the complex process of state changes into a manageable and analytically useful format. Each entry in the matrix represents the probability of transitioning from one state to another. This matrix is vital in Markov chain analysis, enabling the prediction of future states and providing insight into the system's long-term behaviour.

In our simulation, the transition matrix plays a central role, detailing the probabilities of state changes within the dynamic system. Each element specifies the likelihood of moving from a current state to a future state. By clearly summarizing these probabilities, the matrix provides a thorough understanding of the system's evolution over time, facilitating accurate predictions and in-depth analysis of complex processes.

The transition Matrix evaluated is:

$$
\begin{bmatrix}
0.08108108 & 0.16216216 & 0.48648649 & 0.08108108 & 0.05405405 & 0.13513514 \\
0.03703704 & 0.33925926 & 0.5437037 & 0.05185185 & 0.00592593 & 0.02222222 \\
0.00210526 & 0.58035088 & 0.17824561 & 0.2 & 0.00280702 & 0.03649123 \\
0.00232198 & 0.24535604 & 0.02167183 & 0.54334365 & 0.01470588 & 0.17260062 \\
0.00445765 & 0.0653789 & 0.01188707 & 0.38484398 & 0.04160475 & 0.49182764 \\
0. & 0.03278689 & 0.01639344 & 0.13114754 & 0.06557377 & 0.75409836
\end{bmatrix}
$$

3.4 Simulation

- The sequence of states provides a distribution of how often each interval is visited. This can help identify which intervals are more likely to be occupied over time using monthly weighted mean and standard deviation on the Nifty 50 index.
- By examining the transitions between intervals, you can understand how the chain moves through the state space. For example, frequent transitions between adjacent intervals may indicate a smooth transition process.
- Over many steps, the chain should converge to a distribution that reflects the stationary distribution of the defined transition matrix, assuming the matrix is Aperiodicity, positive recurrence and irreducible.

This will help to analyze the behavior of the Nifty 50 index being modeled, such as understanding the likelihood of the index being in a particular state or interval over time.

3.5 Metropolis-Hastings Algorithm

The Metropolis-Hastings algorithm is a key Markov Chain Monte Carlo (MCMC) method widely used in statistics, machine learning, and scientific research. It enables sampling from complex probability distributions through a systematic process of proposing and accepting or rejecting new states based on specific criteria. This algorithm is particularly valuable for Bayesian inference, parameter estimation, and model fitting, serving as a powerful tool for tackling intricate probabilistic problems in various fields.In this study, a small positive value (epsilon, e-6) is added to the diagonal elements of the transition matrix to prevent division by zero when calculating acceptance ratios in the Metropolis-Hastings algorithm. This addition acts as a regularization term, ensuring numerical stability throughout the simulation.

Methodology

The research methodology is structured into distinct phases over 10,000 iterations:

1. Initialization: The states associated with the Nifty 50 index are initialized, setting the foundation for subsequent transitions.
2. Current State: The initial state is set to state 3 (between 0 and 1), reflecting the index's prevailing characteristic.
3. Proposed Next State: In each iteration, a potential next state is randomly sampled based on the transition probabilities in the transition matrix.
4. Transition Probabilities: Two key probabilities are calculated:

- Transition probability proposed: The likelihood of moving from the current state to the proposed state.
- Transition probability current: The chance of remaining in the current state.

5. Acceptance Ratio: The Metropolis-Hastings acceptance ratio is computed as:

$$Acceptance\ ratio = min\ \{1,\ transition\_prob\_proposed\ /\ ,max(transition\_prob\_current, \epsilon)\ \}$$

This ratio compares the probability of transitioning to the proposed state against the probability of staying in the current state. The proposed state is accepted with a probability equal to this ratio.

6. State Update: If the proposed state is accepted, the current state index is updated to reflect the system's new state.

This iterative process forms the core of the MCMC simulation, enabling the exploration and approximation of complex probability distributions governing the Nifty 50 index.

3.6 Posterior distribution

Bayesian statistics canters on the posterior distribution, which updates beliefs about a state or parameter by incorporating observed data. This distribution is derived using prior knowledge, a transition probability matrix, and Markov Chain Monte Carlo simulation, particularly the Metropolis-Hastings algorithm. It quantifies the likelihood of different states or parameters based on both prior beliefs and new evidence. Bayesian inference iteratively combines these elements, enabling refined understanding and informed decision-making within a probabilistic framework. This approach allows for continuous updating of knowledge as new data becomes available, providing a powerful tool for analysing complex systems and phenomena under uncertainty.

4 Model Fitting

4.1 Assessment of Convergence

The observed decline in autocorrelation values as the lag increases in the autocorrelation plot provides further evidence of the NSE index simulation converging towards its equilibrium state. This pattern in the autocorrelation function indicates that the simulation is nearing its stationary distribution. It can be observed from the ACF plot above.

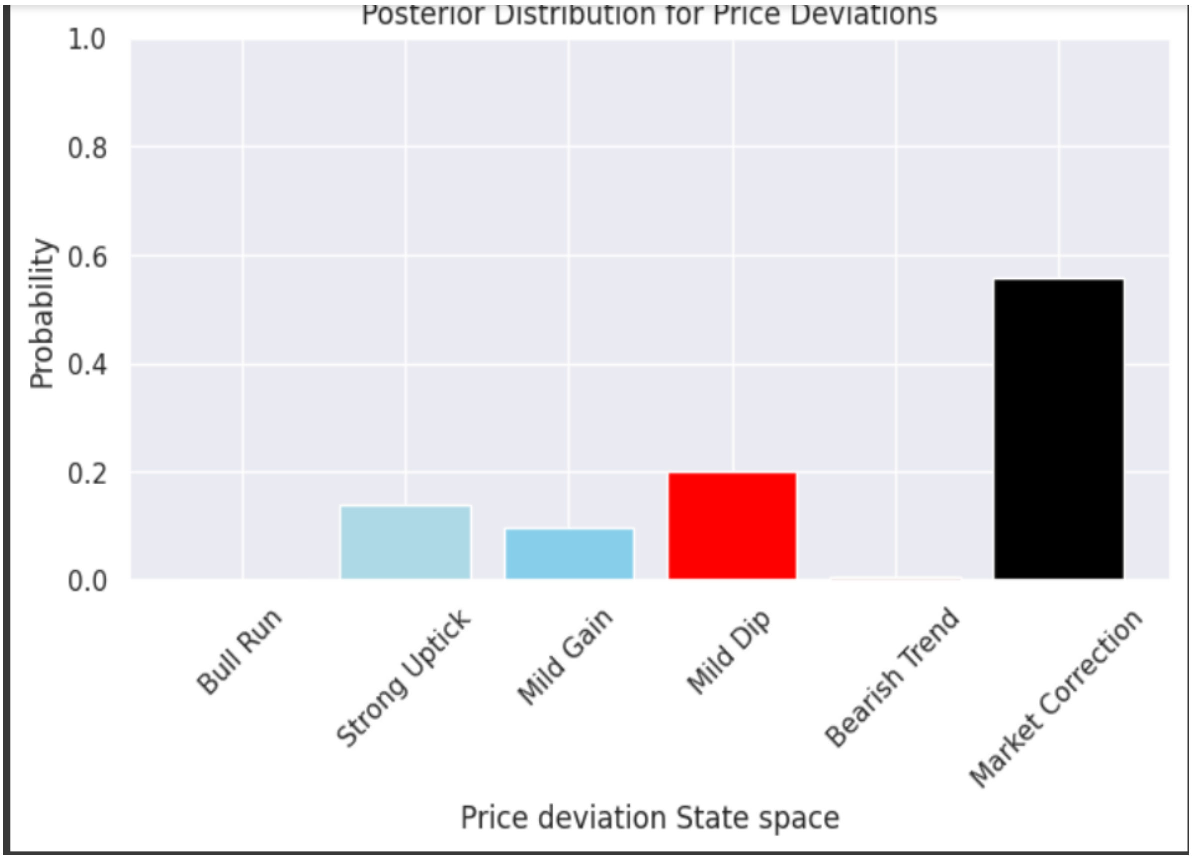
4.2 Assessment of Stationarity

In a time-series that is not moving, the autocorrelations should decrease rapidly to almost zero as the lag increases. This suggests that previous values do not show any relationship with future values, a main feature of stationarity.

If the autocorrelation function displays brief dependence (e.g., strong correlations for short lags followed by a quick decrease), it indicates that the time series is probably stationary as observed in ACF plots pasted above.

5 Empirical results

5.1Posterior Distribution



State: Bull Run, Probability: 0.0008
State: Strong Uptick, Probability: 0.1374
State: Mild Gain, Probability: 0.0970
State: Mild Dip, Probability: 0.2020
State: Bearish Trend, Probability: 0.0042
State: Market Correction, Probability: 0.5586

- The distribution represents the estimated probabilities of the Nifty 50 index being in a particular state after running the simulation using monthly weighted mean and standard deviation. This distribution is an empirical approximation of the true stationary distribution of the Markov chain.
- The probabilities indicate how often each state is visited relative to others. Higher probabilities suggest that certain states are more stable or attractive in the context of the simulated system.
- Including states with zero probability ensures that the analysis considers the entire state space, which is important for understanding the full scope of potential outcomes.
- The visualization shows that below 2 standard deviations has the highest probability of occurrence. This may be because the prior distribution of Nifty 50 index taken for the analysis has shown a great degree of downward fluctuation in a short time span. But over the long term the index has shown steady upward movement.

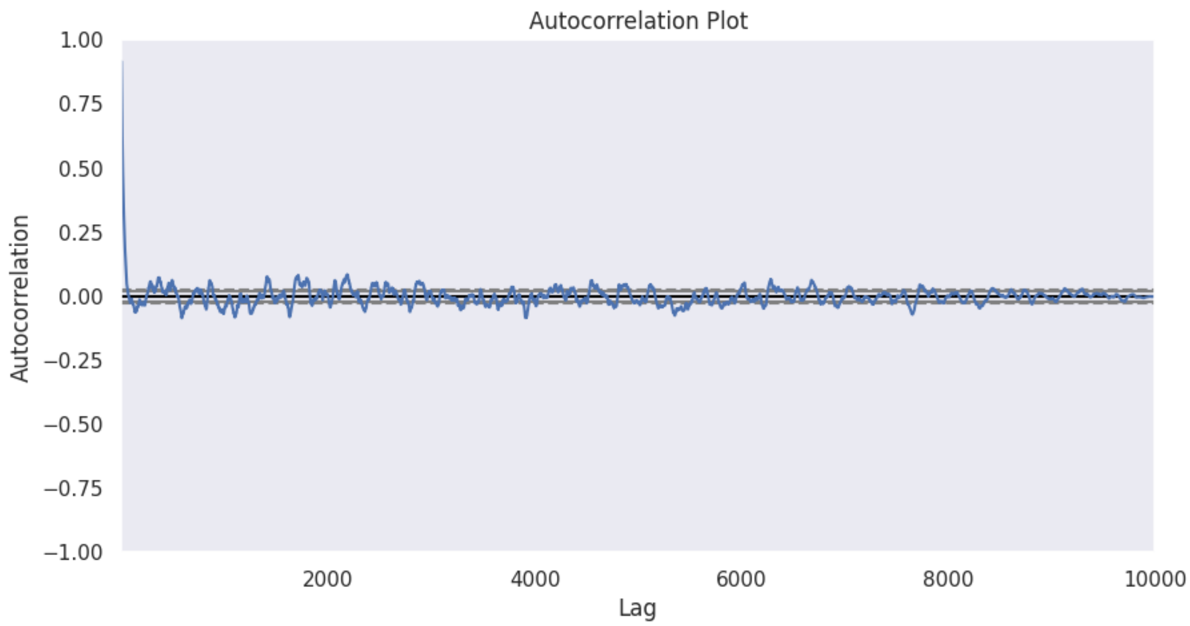
5.2 Stationary Distribution

- The plot starts with the third state ie. Between 0 and 1, having a probability of 1, as all probability is initially concentrated there.
- Over time, the probabilities of all states change as the chain evolves. The plot shows how these probabilities stabilize, indicating convergence towards a stationary distribution.
- The plot reveals that state 2 i.e. between 1 and 2 become most probable over time, providing insights into the system's long-term behavior.
- However, state 2 is closely followed by state 6 showing that wild fluctuation greater than 2 standard deviation is possible in the Nifty 50 index.

  Stationary probability of Bull Run = 0.012304
  Stationary probability of Strong uptick = 0.277517
  Stationary probability of Mild gain = 0.20253
  Stationary probability of Mild dip = 0.218715
  Stationary probability of Bearish trend= 0.024453
  Stationary probability of Market correction = 0.26448

Overall, the plot effectively demonstrates the dynamic behavior of the Markov chain and its convergence properties, offering a clear visual representation of how state probabilities evolve over time.

5.3 Autocorrelation plot



- Lag: The x-axis represents the lag, or the number of time steps between the compared values.
- The y-axis shows the autocorrelation values, indicating the degree of similarity between the state sequence and itself at different lags.
  - The plot shows high autocorrelation at low lags, it suggests that the states are highly dependent on their recent history.
  - A gradual decay in autocorrelation as lag increases indicates that the influence of past states diminishes over time.
  - Since the autocorrelation stabilizes around zero at higher lags, it suggests that the Markov chain has reached a stationary distribution.

This plot helps assess the temporal dependencies in the simulated state sequence and provides insights into the dynamics of the Markov chain model.

5.4 Trace plot

- The plot shows the sequence of states visited over time. Each point represents a state at a particular time step, connected by lines to indicate transitions.
- If the plot shows that the state indices stabilize over time, it suggests that the Markov chain is converging to a stationary distribution.
- Frequent horizontal lines indicate that the chain remains in the same state for multiple steps, while vertical jumps indicate transitions between states.
- The plot can reveal patterns or cycles in the state transitions, providing insights into the dynamics of the system being modeled.

Overall, the trace plot is a useful tool for visually assessing the behavior and convergence of the Markov chain over time.

6. Conclusion

The long-term outlook for the Nifty 50 index shows a balanced market with a slight bullish tendency. Strong Upticks (27.75%) are most probable, followed closely by Market Corrections (26.45%). Mild fluctuations, both positive and negative, are common (Mild Gain: 20.25%, Mild Dip: 21.87%). Extreme states (Bull Run: 1.23%, Bearish Trend: 2.45%) are rare, indicating overall market stability. The combined probability of positive states (Strong Uptick + Mild Gain + Bull Run = 49.24%) slightly outweighs negative states (Market Correction + Mild Dip + Bearish Trend = 50.76%).

This distribution suggests a stable, cyclical market with a propensity for growth. Investors can anticipate frequent minor fluctuations, occasional strong upward movements, and periodic corrections. The data points to a resilient index with a slight long-term positive bias, offering opportunities for strategic investment across various market conditions.