Predicting Stock Prices using Hidden Markov Models: A Study of Markov Models and Hidden Markov Models

A Thesis

Submitted to the Indian Institute of Technology Hyderabad

in partial fulfillment of requirements for the award of degree

Master of Science

in

Mathematics

by

Nikhil Kumar Patel

MA21MSCST11010

DEPARTMENT OF Mathematics

Indian Institute Of Technology

Hyderabad

June 2023

DECLARATION

I Nikhil Kumar Patel hereby declare that the thesis entitled **Predicting Stock Prices using Hidden Markov Models: A Study of Markov Models and Hidden Markov Models**, submitted for partial fulfillment of the requirements for the award of degree of Master of Science of the Indian Institute of Technology, Hyderabad is a bonafide work done by me under supervision of Amit Tripathi

This submission represents my ideas in my own words and where ideas or words of others have been included, I have adequately and accurately cited and referenced the original sources.

I also declare that I have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained.

Nikhil Kumar Patel

MA21MSCST11010

Acknowledgement

I would like to place on record my sincere gratitude to my project guide Dr. **Amit Tripathi**, Assistant Professor, Department of Mathematics , Indian Institute Of Technology Hyderabad for his support, co-operation, guidance and for providing me with all the necessary facilities.

I take this opportunity to express my most profound sense of gratitude and sincere thanks to everyone who helped me to complete this work successfully.

Finally, I thank my family and friends who contributed to the successful fulfillment of this project work.

Nikhil Kumar Patel

Approval sheet

This thesis entitled **"Predicting Stock Prices using Hidden Markov Models: A Study of Markov Models and Hidden Markov Models"** by Nikhil Kumar Patel is approved for the degree of partial fulfillment of Master of Science in Mathematics from Indian Institute Of Technology, Hyderabad.

Signature

Dr. **Amit Tripathi** (Advisor)

Department of Mathematics

IIT Hyderabad

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Financial markets have always fascinated traders, investors, analysts, and researchers because of their inherent unpredictability and the potential for significant gains and losses. Among the many aspects of financial analysis, accurately predicting stock prices remains an interesting and challenging task. Many investors and traders attempt to uncover patterns, trends, and indicators to make informed investment decisions and maximize profits.

There are a number of methods for forecasting stock prices. Some of them, such as regression analysis, time series models, and artificial neural networks, have proven their effectiveness in specific market conditions. However, these methods often fail to capture the sudden shifts in market sentiment and also struggle to deal with the complex interactions between several market factors. As a result, researchers and market participants are hunting for more powerful and accurate forecasting models.

In recent years, advanced probabilistic models have caught the attention of many researchers for forecasting financial entities. One such model that has proven to capture uncertain market conditions is the Hidden Markov Model (HMM). HMMs are statistical models that incorporate hidden states, which represent unobservable elements that drive observed data sequences. They are based on Markov process concepts. The inherent flexibility of HMMs allows researchers to adapt to different market regimes, capture the transitions between different states, and adapt to changing

market conditions.

1.2 Objective of the Thesis

In this thesis, we attempt to validate the potential of the HMM in stock price forecasting and provide an extensive analysis of the effectiveness of the HMM compared to the traditional historical average return (HAR) approach. We also aim to explore the theoretical foundations of Markov models, with a specific focus on hidden Markov models, by delving into the underlying principles and mathematical frameworks of HMMs.

1.3 Thesis Organization

Chapter 2 dives into the fundamentals of Markov models. Chapter 3 explores the foundation and fundamentals of hidden Markov models. Chapter 4 describes the methodology and implementation of the HMMs in predicting stock prices. Chapter 5 investigates and describes the implementation's outcomes. Finally, Chapter 6 provides the conclusion of the project.

Chapter 2

Markov Models

2.1 Introduction

Sequential data sets arise in various fields ranging from finance, speech recognition, DNA sequence, and many more, which often play a crucial role in allowing us to gain insights, make predictions, and understand underlying patterns in time series and other forms of ordered data. Often times we assume that the data points are independent and identically distributed (i.i.d.). While the i.i.d. assumption is useful in many situations, it fails to capture the interdependencies and temporal dynamics seen in sequential data. For example, in time series analysis, we might seek to forecast future values using past or historical data. However, as the number of observations rises, it becomes impractical to assume that all prior observations equally influence future predictions. Additionally, the i.i.d. assumption does not account for the correlations and dependencies present in the sequential data.

To take advantage of these patterns and to capture the dependencies between observations that are close in sequence, we need to loosen up the i.i.d. assumption and consider models that can account for the sequential nature of data. The concept of memory and the assumptions of conditional independence make Markov models promising for modeling sequential data. Assuming that future predictions will depend on the most recent observations, Markov models capture the interdependencies while managing the computational complexity associated with considering all past observations. This assumption aligns with our intuition that recent observations are

more informative in predicting future values.

A canonical reference for the material in this chapter is ”Pattern Recognition and Machine Learning” by Christopher M. Bishop, specifically [3, Chapter 13], but it has been expanded with additional explanations and examples to make it easier to understand.

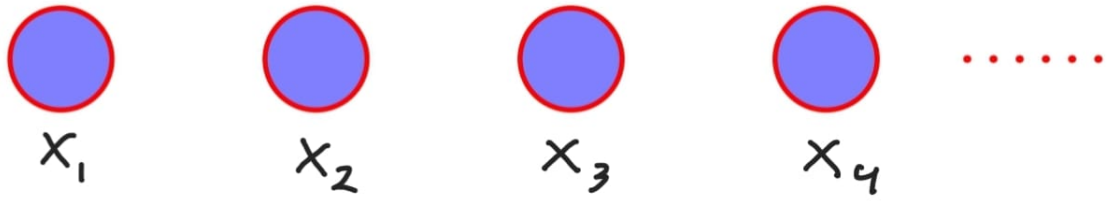


Figure 2.1: Graphical representation of i.i.d. sequence of observation

2.2 First order Markov Chain

The product rule, which is used to represent the joint distribution of a series of data, allows us to take into account the relationships between numerous observations. We introduce the idea of a Markov model by assuming that each conditional distribution depends simply on the most recent observation.

Let $\{X_1, X_2, \ldots, X_N\}$ be a sequence of random variables, then the joint distribution for this sequence of random variables can be represented as:

$$P(X_1, \ldots, X_N) = \prod_{n=1}^{N} P(X_n | X_1, \ldots, X_{n-1}) \tag{2.1}$$

One of the simplest types of Markov models is the first-order Markov chain. It assumes that the conditional distribution of each observation depends only on its immediate predecessor. Figure (2.2), taken from Bishop [3], represents this concept using a graphical model, where each observation acts as a node in a graph and the relationships between observations are represented by the edges of the graph.

Under a first-order Markov chain, the joint distribution for a sequence of observa-

tions $\{X_1, X_2, \ldots, X_N\}$ is given by:

$$P(X_1, \ldots, X_N) = P(X_1) \prod_{n=2}^{N} P(X_n|X_{n-1}) \tag{2.2}$$

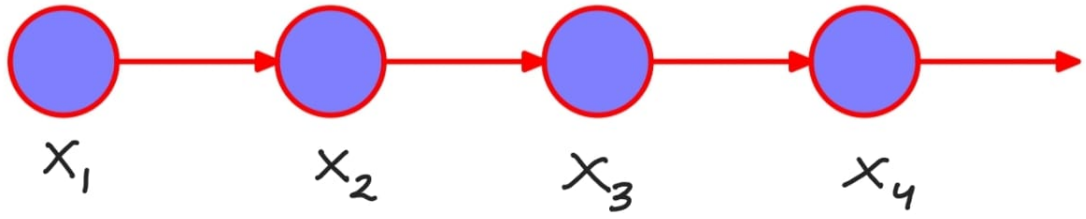


Figure 2.2: Graphical representation of a first-order Markov chain

i.e., the conditional distribution, depends only on the immediate predecessor, mathematically $P(X_n|X_{n-1}, X_{n-2}, \ldots, X_1) = P(X_n|X_{n-1})$; thus, if we use such a model to forecast the next observation in a sequence, the distribution of predictions will be determined solely by the value of the immediately preceding observation and will be independent of all previous observations.

The conditional distributions $P(X_n|X_{n-1})$ that define these models are often restricted to being equal, equivalent to the assumption of a stationary time series. The model is thus referred to as a homogeneous Markov chain. As an example, if the conditional distributions depend on parameters that are adjustable, then each of the conditional distributions in the sequence will have the exact same values for those parameters.

To put things together, let us consider an example using the elementary 3-state Markov model of students academic level as illustrated in figure (2.3). We suppose that a student can have any one of the following academic levels:
1st state (i=1): high academic achievement.
2nd state (i=2): average academic performance.
3rd state (i=3): low academic performance.

We assume that just one of the three aforementioned states represents the current

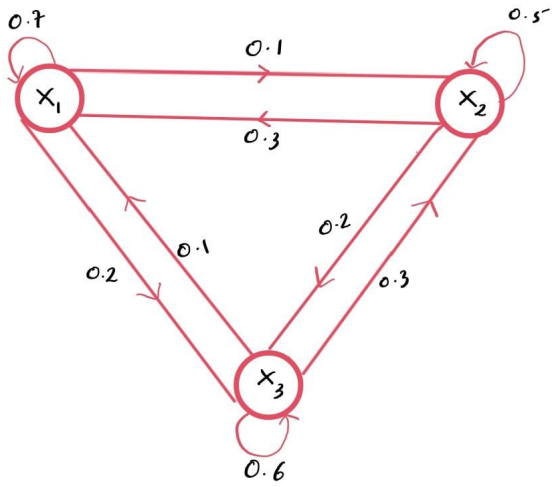


Figure 2.3: Graphical representation of a 3-state Markov model of students academic level

academic level of a student and that the matrix A of state transition probabilities is:

$$A = \{a_{ij}\} = \begin{bmatrix} 0.7 & 0.1 & 0.2 \\ 0.3 & 0.5 & 0.2 \\ 0.1 & 0.3 & 0.6 \end{bmatrix}$$

where each a_{ij} represents the probability of transitioning from state i to state j, with $1 \leq$ i, j $\leq N$, i.e.

$$a_{ij} = P\left[X_n = j | X_{n-1} = i\right], 1 \leq i, j \leq N,$$

Now, let's re-frame the question using the analogy of the student's academic performance: Suppose the student's academic level in the first year (t=1) is average. We want to determine the odds that the academic level forecast for the next seven years will be "low-high-average-average-high-low-high." In other words, we want to calculate the probability of the observation sequence O = {3, 1, 2, 2, 1, 3, 1} occurring at t=1,2,. . . ,8, given the initial state of average (2) on year 1. By applying the principles of the Markov model and using the state transition probability matrix A, we can calculate the odds or probability of the specified observation sequence occurring over the next seven years

as

$$
\begin{aligned}
P(X) &= P(X_1 = 2, X_2 = 3, X_3 = 1, \ldots, X_8 = 1) \\
&= P(X_1 = 2) \cdot P(X_2 = 3|X_1 = 2) \cdot P(X_3 = 1|X_2 = 3) \cdot P(X_4 = 2|X_3 = 1) \\
&\cdot P(X_5 = 2|X_4 = 2) \cdot P(X_6 = 1|X_5 = 2) \cdot P(X_7 = 3|X_6 = 1) \cdot P(X_8 = 1|X_7 = 3) \\
&= \pi_2 \cdot a_{23} \cdot a_{31} \cdot a_{12} \cdot a_{22} \cdot a_{21} \cdot a_{13} \cdot a_{31} \\
&= 1 \cdot (0.2)(0.1)(0.1)(0.5)(0.3)(0.2)(0.1) \\
&= 6 \times 10^{-6}
\end{aligned}
$$

where we use the notation $\pi_i = P(X_1 = i), 1 \leq i \leq N$ to denote the initial state probabilities.

2.3 Higher-Order Markov Chains

The associations between immediate neighboring observations are captured by first-order Markov chains but might not capture longer-term trends or patterns. To overcome this problem, we can extend the model to higher-order Markov chains, where each observation depends on several earlier observations. For instance, the joint distribution for a sequence of observations under a second-order Markov chain is influenced by the two previous observations.

Let $\{X_1, X_2, \ldots, X_N\}$ be a sequence of observations, then the joint distribution for the given sequence under the second-order Markov chain is given by:

$$
P(X_1, X_2, \ldots, X_N) = P(X_1) \cdot P(X_2|X_1) \prod_{n=3}^{N} P(X_n|X_{n-1}, X_{n-2}) \tag{2.3}
$$

Figure (2.4),taken from Bishop [3], represents this concept using a graphical model, where each observation acts as a node in a graph and the relationships between observations are represented by the edges of the graph.

Similarly, the joint distribution for the given sequence under an M'th-order Markov

chain is given by:

$$P(X_1, X_2, \ldots, X_N) = P(X_1) \cdot P(X_2|X_1) \ldots P(X_M|X_{M-1}, \ldots, X_1) \prod_{n=M+1}^{N} P(X_n|X_{n-M}, \ldots, X_{n-1}) \tag{2.4}$$

In a first-order Markov chain, where each observation is influenced by its immediate

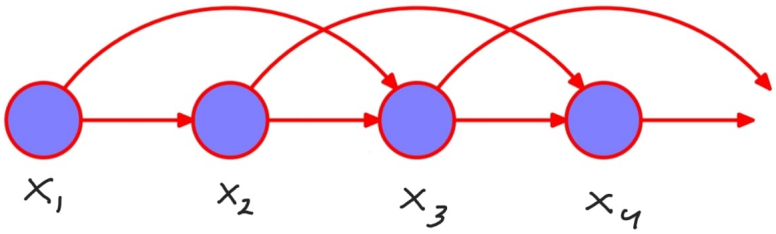


Figure 2.4: Graphical representation of a second-order Markov chain

preceding observation, the conditional distribution can be represented by a matrix of transition probabilities. Let A_{ij} represent the probability of transitioning from state j to state i, i.e., $P(X_n = i|X_{n-1} = j)$. The sum of transition probabilities for each state i should be equal to 1, denoting a valid probability distribution.

Therefore, if the observations are discrete variables with K possible states, the number of parameters required to represent the conditional distribution $P(X_n|X_{n-1})$ is $K(K-1)$. Each state has $K-1$ transitions to other states, and there are K states in total. Hence, the total number of parameters is $K(K-1)$.

If we extend the model to an M'th-order Markov chain, where each observation depends on the previous M observations, the joint distribution is built using conditionals $P(X_n|X_{n-M}, \ldots, X_{n-1})$, the number of parameters in this model increases to $K^M(K-1)$. Hence, as the order of the Markov chain increases, the model's number of parameters expands exponentially, making it computationally challenging for larger values of the order. Although Markov models offer a workable answer, they are not always able to capture the more complex relationships in sequential data. To address this, we introduce latent variables, which are hidden variables that capture the underlying factors or hidden states in the data. This brings us to the broader category of hidden Markov models, where more complex models are created from simpler components, allowing for more accurate modeling of the sequential data.

Therefore, we have considered the hidden Markov model as an alternative modeling technique to overcome this limitation.

2.4 Introducing Latent variables: State Space Models

We introduce additional latent variables Z_n for each observation X_n to develop models for sequential data that are not constrained by the Markov assumption to any order and can be defined with a limited number of free parameters. We now suppose that the latent variables form a Markov chain, giving birth to the graphical structure known as a state space model, which is illustrated in Figure (2.5). It fulfils the crucial conditional independence requirement that is Z_{n-1} and Z_{n+1} are independent given Z_n, i.e.

$$Z_{n+1} \perp\!\!\!\perp Z_{n-1} | Z_n. \tag{2.5}$$

To prove the conditional independence property stated in Equation (2.5), we use the **d-separation** property. D-separation is a criterion based on the graphical structure of a probabilistic graphical model that determines whether two sets of variables are independent given a third set. In our case, we can apply the d-separation property to the graphical structure of the state space model. The state space model consists of latent variables $\{Z_1, Z_2, \ldots, Z_N\}$ corresponding to observations $\{X_1, X_2, \ldots, X_N\}$. The graphical structure of the state space model is depicted in Figure (2.5), taken from Bishop [3].

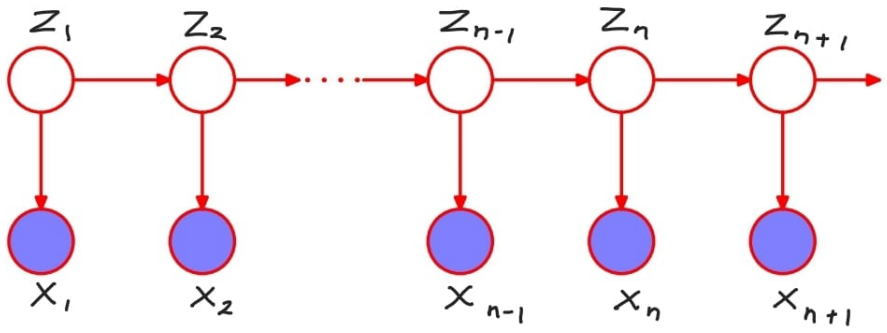


Figure 2.5: Graphical representation of a state space model

To evaluate the conditional independence property using d-separation, we follow

the below procedure:

- Examine the graphical structure and identify the variables involved in the conditional independence statement.
- Identify all possible paths between the variables given the conditioning set.
- For each path, check if it contains any blocked nodes. A blocked node is a node that, when conditioned upon, blocks the flow of influence along a path. A node is considered blocked if one of the following conditions is met:
  - The arrows along the path meet head-to-tail or tail-to-tail at the node, and the node itself is in the conditioning set.
  - The arrows along the path meet head-to-tail or tail-to-tail at the node, and the node itself is in the conditioning set.
- If all paths are blocked, the variables are conditionally independent given the conditioning set. If any path is unblocked, the variables are dependent given the conditioning set.

In the state space model, the variables involved are Z_{n+1}, Z_n, and Z_{n-1}. The path between Z_{n+1} and Z_{n-1} passes through Z_n, so the path is $Z_{n+1} \rightarrow Z_n \rightarrow Z_{n-1}$. In our case, Z_n is observed or conditioned upon. Therefore, according to the d-separation rules: the arrows along the path $Z_{n+1} \rightarrow Z_n \rightarrow Z_{n-1}$ meet head-to-tail at Z_n, and Z_n itself is in the conditioning set. Hence, Z_n blocks the path. Thus, the path $Z_{n+1} \rightarrow Z_n \rightarrow Z_{n-1}$ is blocked, thus we can conclude that Z_{n+1} and Z_{n-1} are conditionally independent given Z_n. Hence, we have proven the conditional independence property expressed in Equation (2.5) using the d-separation property and the graphical structure of the state space model.

To further illustrate the proof, we can refer to Figure (2.6), which displays the graphical representation of the state space model and visually demonstrates the conditional independence property step by step. In Figure (2.6), the variables Z_{n+1}, Z_n, and Z_{n-1} are represented as nodes, and the arrows indicate the direct connections between them. The path between Z_{n+1} and Z_{n-1} passes through Z_n. We first examine the graphical structure and identify the paths between Z_{n+1} and Z_{n-1} given Z_n. In this case,

we have the path $Z_{n+1} \rightarrow Z_n \rightarrow Z_{n-1}$. Next, we check if there exists any other path that passes through Z_n and connects Z_{n+1} and Z_{n-1}. In our graph, we can clearly see that the path $Z_{n+1} \rightarrow Z_n \rightarrow Z_{n-1}$ passes through Z_n. We remove the path between Z_{n+1} and Z_{n-1} since it passes through the observed variable Z_n. After removing the path, we check if there are any remaining paths between Z_{n-1} and Z_{n+1}. If no such path exist then we can conclude that Z_{n+1} and Z_{n-1} are conditionally independent given Z_n. In Figure (2.6), we can observe that there are no remaining paths between these variables. Thus Z_{n+1} and Z_{n-1} are conditionally independent given Z_n.

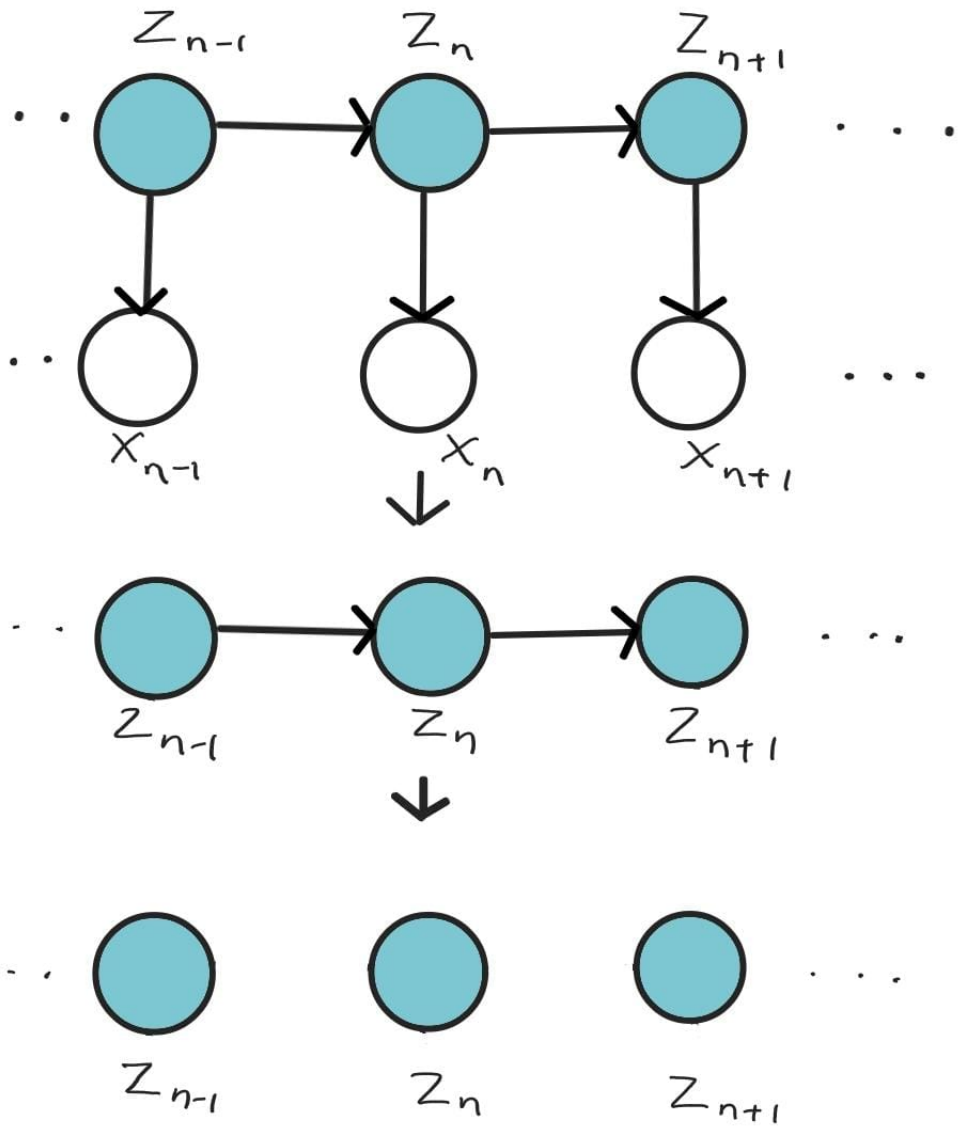


Figure 2.6: Graphical representation demonstrating the conditional independence property

The joint distribution for a sequence of N observations and corresponding latent variables in a state space model is given by:

$$P(X_1, \ldots, X_N, Z_1, \ldots, Z_N) = P(Z_1) \prod_{n=2}^{N} P(Z_n|Z_{n-1}) \prod_{n=1}^{N} P(X_n|Z_n) \tag{2.6}$$

Applying the d-separation criterion, we can observe that there is always a path connecting any two observed variables X_n and X_m via the latent variables Z_n (for $n = 1, 2, \ldots, N$). Importantly, this path is never blocked, meaning that there are no conditional independence properties among the observed variables given the latent variables. Consequently, the predictive distribution $P(X_{n+1}|X_1, \ldots, X_n)$ for the next observation X_{n+1}, given all previous observations, does not exhibit any conditional independence relationships. Thus, our predictions for X_{n+1} depend on all the preceding observations. However, we note that the observed variables in the state space model do not satisfy the Markov property in any order. In other words, the observed variables cannot be modelled as a Markov chain, as the dependencies among them are influenced by the latent variables.

When the latent variables in a state space model take on discrete values, the resulting model is known as a hidden Markov model (HMM). In conclusion, while Markov models provide a powerful framework for analysing sequential data due to their ability to capture dependencies and temporal dynamics. HMMs improve the models capabilities, which allow the representation of more complex relationships in sequential data. As they can accommodate both discrete and continuous variables, they have gained significant popularity in the field of sequential data modelling.

Chapter 3

Hidden Markov Models

As we have seen, when the latent variables in a state space model take on discrete values, they result in a hidden Markov model (HMM). HMMs are probabilistic models that find wide application in various domains, including speech recognition, natural language processing, and bioinformatics. They are particularly useful for modeling systems with hidden states that generate observable outputs.

In Chapter 13 of ”Pattern Recognition and Machine Learning” by Christopher M. Bishop [3], the mathematical principles underlying hidden Markov models are explored. The chapter delves into the probabilistic formulation of HMMs, highlighting the role of latent variables, the determination of emission probabilities, and the joint probability distribution. It also discusses the flexibility of HMMs in terms of emission distributions and the integration of discriminative models.

The foundational probabilistic model for HMMs is introduced, covering essential concepts such as latent variables, the 1-of-K coding scheme, and the conditional distribution of latent variables. The chapter further explores transition probabilities and the representation of the initial latent node. To aid understanding, visual representations in the form of state transition diagrams and lattice or trellis diagrams are presented. Emission probabilities and their adaptability to different data types are thoroughly examined, considering the diversity of observed outputs in real-world applications. Finally, the joint probability distribution and the parameter set governing the HMM are discussed, shedding light on the underlying statistical framework.

For those seeking further in-depth knowledge, it is worth noting that this chapter

draws inspiration from Chapter 13 of "Pattern Recognition and Machine Learning" by Christopher M. Bishop [3]. For readers desiring a more comprehensive understanding, we suggest to refer the original material, as it provides detailed explanations and examples on the subject matter.

3.1 Probabilistic Model for HMMs

HMMs rely on latent variables, which play a crucial role in capturing the underlying hidden states of a system. Hidden states are unobservable factors in the system that have an important influence on observable outcomes. They are denoted by the sequence $Z = \{Z_1, Z_2, \ldots, Z_N\}$, where N is the sequence length. Each hidden state Z_k is a K-dimensional binary vector represented by $Z_k = [z_{k,1}, z_{k,2}, \ldots, z_{k,K}]$, where $z_{k,i} \in \{0, 1\}$ for $i = 1$ to K and $z_{k,i} = 1$ only for one i.. This representation allows us to determine which component of the mixture is responsible for generating the observation X_n.

The conditional distribution of latent variables, $P(Z_n|Z_{n-1}, A)$, determines the probability distribution of Z_n based on the previous latent variable, Z_{n-1}. This distribution is established using transition probabilities, denoted as A_{ij}, which are elements of a transition matrix A. These probabilities capture the dynamics of latent variable transitions.

Transition probabilities, represented by A_{ij}, indicate the likelihood of transitioning from state i to state j. These probabilities are constrained to values between 0 and 1. Furthermore, each row in the transition matrix sums to 1, ensuring a valid probability distribution, i.e., mathematically, it can be seen as follows:

$$A_{ij} \equiv P(z_{n,j} = 1|z_{n-1,i} = 1), \tag{3.1}$$

subject to the constraints:

$$0 \leq A_{ij} \leq 1,$$

with the requirement that the sum of each row is equal to 1:

$$\sum_j A_{ij} = 1.$$

The conditional distribution of Z_n given Z_{n-1} and A can be expressed explicitly as:

$$P(Z_n|Z_{n-1}, A) = \prod_{j=1}^{K} \prod_{i=1}^{K} A_{ij}^{z_{n-1,i} z_{n,j}} \tag{3.2}$$

$A_{i,j}$ represents the transition probability from state i to state j, and $z_{n,j}$ represents the value of the $j-th$ component of Z_n.

The initial latent node, Z_1, is unique as it does not possess a parent node. As a result, it has a marginal distribution, $P(Z_1)$, which is represented by a probability vector, π. Each element of π, denoted as π_k, signifies the probability of $z_{1,k}$ being 1, mathematically, $\pi_k \equiv P(z_{1,k} = 1)$, so that

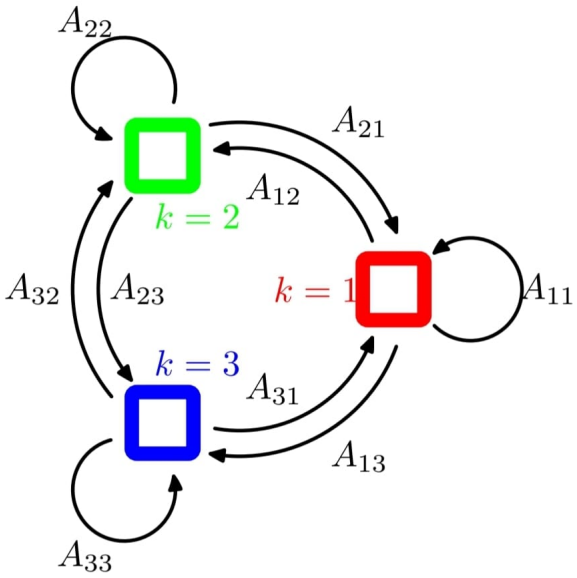
$$P(Z_1|\pi) = \prod_{k=1}^{K} \pi_k^{z_{1,k}} \tag{3.3}$$

where $\sum_k \pi_k = 1$.

Figure (3.1), taken from Bishop [3], shows a state transition diagram that represents the transition probabilities. In this diagram, states are depicted as nodes, and transition probabilities are depicted as directed edges. However, it is important to note that this representation does not constitute a probabilistic graphical model, as the nodes represent states of a single variable rather than distinct variables. An alternative representation of the state transitions is the lattice or trellis diagram. This diagram involves unfolding the state transition diagram over time, illustrating the transitions between latent states as shown in figure (3.2), taken from Bishop [3].

Observation symbols are the observable outputs of the system. They are denoted by the sequence $X = \{X_1, X_2, \ldots, X_N\}$, where N is the sequence length. Each observation symbol X_t at time t corresponds to a specific output associated with the hidden state Z_t. Observation symbols provide data that aids in inferring hidden states and understanding the system's behaviour. Emission probabilities, $P(X_n|Z_n, \phi)$, define the conditional distributions of observed variables given the latent variables.

Figure 3.1: State Transition Diagram for a latent variable with k=3 states



These probabilities are determined by the parameters ϕ, which govern the emission distributions. The emission probabilities can be represented as follows:

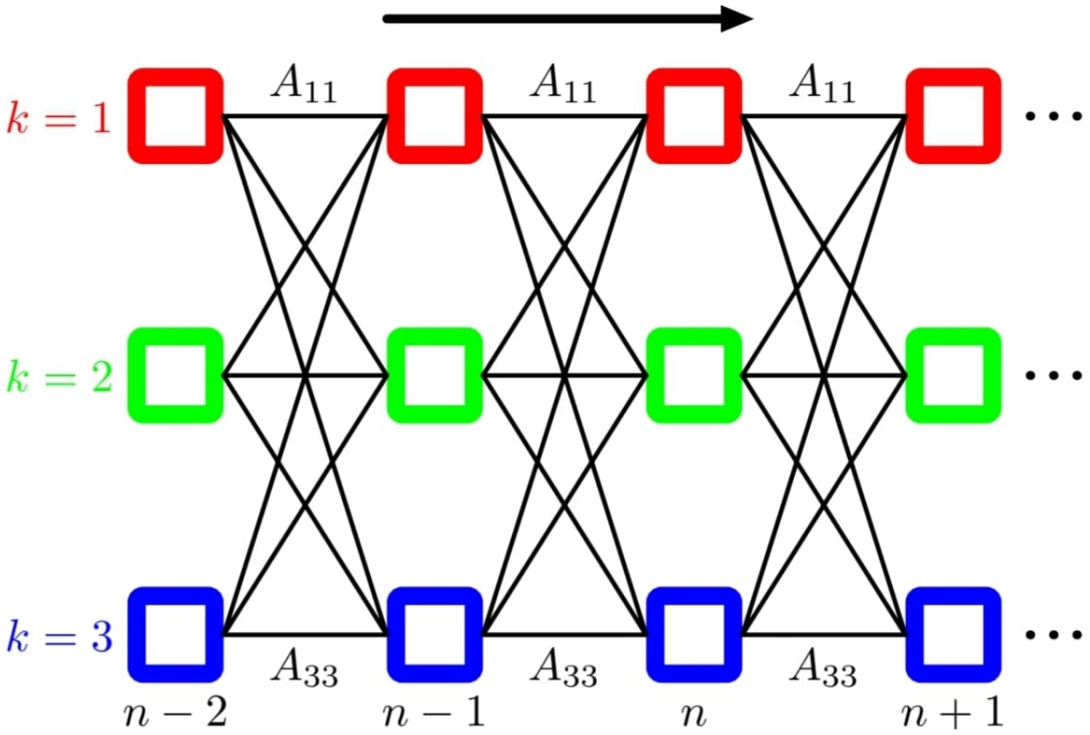
$$P(X_n|Z_n, \phi) = \prod_{k=1}^{K} P(X_n|\phi_k)^{z_{n,k}}. \tag{3.4}$$

The choice of emission distributions depends on the nature of the observed data. We will focus on homogeneous models, which assume that all conditional distributions regulating latent variables, represented by A, have the same parameters. Likewise, all emission distributions have the same parameters, ϕ. This assumption simplifies the model, but it may be expanded to handle more generic circumstances.

3.2 Joint Probability Distribution

We will first introduce the necessary notation and definitions to understand the joint probability distribution of latent and observed variables in an HMM. The joint probability distribution of latent and observed variables is given by $P(X, Z|\theta)$, where X represents observed variables $\{X_1, \ldots, X_N\}$ and Z denotes latent variables $\{Z_1, \ldots, Z_N\}$. The parameter set θ includes π, A, and ϕ, which govern the HMM. The joint probability

Figure 3.2: Lattice Diagram of latent variables where each column of this diagram corresponds to one of the latent variables



distribution of both latent and observed variables can be expressed as:

$$P(X, Z|\theta) = P(Z_1|\pi) \cdot \left[\prod_{n=2}^{N} P(Z_n|Z_{n-1}, A)\right] \cdot \prod_{i=1}^{N} P(X_i|Z_i, \phi) \tag{3.5}$$

The emission distributions used in HMMs are determined by the features of the observed data. Various distributions, such as discrete tables, Gaussians, and Gaussian mixtures, can be employed. Because of this adaptability, HMMs can handle a broad range of data types. HMMs are tractable models that can handle complicated data structures and capture sequential relationships effectively. Because of their mathematical formulation and methods, they are useful for a wide range of applications.

HMMs can also integrate discriminative models such as neural networks in addition to classic generative models such as Gaussian emission distributions. These models can simulate the emission density $P(X|Z)$ directly or offer a representation for $P(Z|X)$ that can be transformed into the needed emission density using Bayes' theorem.

3.3 Maximum Likelihood Estimation in Hidden Markov Models

Estimating the parameters of an HMM is essential for its successful application. In this chapter, we delve into the concept of maximum likelihood estimation in HMMs and introduce the Expectation Maximization (EM) algorithm as an efficient framework for parameter estimation.

Maximum Likelihood Estimation (MLE) is a popular approach for estimating the parameters of a statistical model based on observed data. In the case of HMMs, we aim to determine the parameters that maximize the likelihood of the observed data given the model. To compute the likelihood function, we need to marginalize over the latent variables in the joint distribution, i.e., if $X = \{X_1, \ldots, X_N\}$ is our observed data set and $Z = \{Z_1, \ldots, Z_N\}$ are its corresponding latent variables, then:

$$P(X|\theta) = \sum_{Z} P(X, Z|\theta) \tag{3.6}$$

However, the joint distribution does not factorize over time, leading to exponential growth in the number of terms to be summed. This creates computational difficulties when directly maximizing the likelihood function. To overcome this issue, we draw inspiration from the inference problem in simple chain models and apply a similar technique to HMMs. By leveraging the conditional independence properties of the graphical structure, we can reorder the summations and develop an algorithm with linear complexity instead of exponential.

Another difficulty in maximising the probability function is the requirement to sum across an increasingly large number of pathways via the lattice diagram. Direct maximization results in complex expressions with no closed-form solutions. Therefore, we turn to the EM algorithm to efficiently maximize the likelihood of HMMs.

3.3.1 EM Algorithm

To address the challenge of maximizing the likelihood function in hidden Markov models (HMMs), we employ the expectation maximization (EM) algorithm. The EM algorithm employs an iterative process to maximize the likelihood function in HMMs. It is important to note that: the EM algorithm begins with an initial set of model parameters denoted as θ'. These parameters act as a starting point for the algorithm's iterative process, where it aims to maximize the likelihood function in hidden Markov models (HMMs). In the E step, we utilize these parameter values to compute the posterior distribution of the latent variables, $P(Z|X, \theta')$. This posterior distribution is then employed to evaluate the expectation of the logarithm of the complete-data likelihood function, resulting in the formation of the function $Q(\theta, \theta')$, described in (Section 9.2) of Bishop [3] as:

$$Q(\theta, \theta') = \sum_{Z} P(Z|X, \theta') \cdot \log P(X, Z|\theta). \tag{3.7}$$

In order to implement the EM algorithm in HMMs, we introduce a notation for storing and representing the posterior distributions of latent variables. We use nonnegative numbers to store the marginal and joint posterior distributions of the latent variables.

Let $\gamma(Z_n)$ represent the marginal posterior distribution of a latent variable Z_n, and $\zeta(Z_{n-1}, Z_n)$ represent the joint posterior distribution of two successive latent variables.

$$\gamma(Z_n) = P(Z_n|X, \theta') \tag{3.8}$$

$$\zeta(Z_{n-1}, Z_n) = P(Z_{n-1}, Z_n|X, \theta') \tag{3.9}$$

To handle the latent variables in the EM algorithm, we adopt a convenient notation. We describe $\gamma(Z_n)$, using a set of K nonnegative integers that add to one for each value of n. Similarly, $\zeta(Z_{n-1}, Z_n)$ is kept in a $K \times K$ matrix of nonnegative values, which likewise adds up to one.

To expand further, we use $z_{n,k}$ to represent the conditional probability of the binary

variable $z_{n,k}$ equaling 1. Similarly, we use $\zeta(z_{n-1,j}, z_{n,k})$ to represent the conditional probability that both $z_{n-1,j}$ and $z_{n,k}$ are equal to 1. These probabilistic variables allow us to compute the expectations required in the EM algorithm. $E[z_{n,k}]$ is the chance that a binary random variable $z_{n,k}$ will take the value 1, which may be computed as the sum of all potential values of the latent variable Z weighted by $\gamma(Z)$ and $z_{n,k}$.

$$\gamma(z_{n,k}) = E[z_{n,k}] = \sum_{Z} \gamma(Z) \cdot z_{n,k} \tag{3.10}$$

Similarly, the expectation of the joint occurrence of $z_{n-1,j}$ and $z_{n,k}$, denoted as $E[z_{n-1,j}, z_{n,k}]$, can be obtained by summing over all possible values of the latent variable Z weighted by $\gamma(Z)$, $z_{n-1,j}$, and $z_{n,k}$.

$$\zeta(z_{n-1,j}, z_{n,k}) = E[z_{n-1,j} z_{n,k}] = \sum_{Z} \gamma(Z) \cdot z_{n-1,j} \cdot z_{n,k} \tag{3.11}$$

After substituting equation (2.6) to (3.7) and simplifying it using equations (3.10) and (3.11), we get:

$$Q(\theta, \theta') = \sum_{i=1}^{K} \gamma(z_{1,i}) \ln \pi_i + \sum_{n=2}^{N} \sum_{j=1}^{K} \sum_{i=1}^{K} \zeta(z_{n-1,j}, z_{n,i}) \ln(A_{ji}) + \sum_{n=1}^{N} \sum_{i=1}^{K} \gamma(z_{n,i}) \ln P(X_n|\phi_i) \tag{3.12}$$

In the M-step of the EM algorithm, we maximize the Q-function, in the equation (3.12), with respect to the parameters $\theta = \{\pi, A, \phi\}$, treating the $\gamma(Z_n)$ and $\zeta(Z_{n-1}, Z_n)$ as constants.

During the M-step, we update the initial state distribution by maximising the Q-function with respect to π, the initial state distribution. We use Lagrange multipliers to enforce the summation constraints associated with the probabilistic interpretation of π. The updated initial state distribution is given by:

$$\pi_i = \frac{\gamma(z_{1,i})}{\sum_{j=1}^{K} \gamma(z_{1,j})}, \tag{3.13}$$

where $\gamma(z_{1,i})$ represents the marginal posterior distribution of the initial state being i.

Next, we update the state transition matrix by maximising the Q-function with

respect to A, the state transition matrix. We use Lagrange multipliers to enforce the summation constraints associated with the probabilistic interpretation of A. The updated state transition matrix is given by:

$$A_{ij} = \frac{\sum_{n=2}^{N} \zeta(z_{n-1,i}, z_{n,j})}{\sum_{k=1}^{K} \sum_{n=2}^{N} \zeta(z_{n-1,i}, z_{n,k})}, \tag{3.14}$$

where $\zeta(z_{n-1,i}, z_{n,j})$ represents the joint posterior distribution of the transition from state i to state j.

Finally, we update the parameters of the emission distribution by maximizing the Q-function with respect to ϕ_i, the parameters of the i-th emission distribution. The specific update equations for the emission distribution depend on the type of emission distribution used in the HMM. For Gaussian emission densities, we have $P(X|\phi_i) = \mathcal{N}(X|\mu_i, \sum_i)$, and maximising $Q(\theta, \theta')$ with respect to μ_i we get the mean parameter μ_i as:

$$\mu_i = \frac{\sum_{n=1}^{N} \gamma(z_{n,i}) X_n}{\sum_{n=1}^{N} \gamma(z_{n,k})} \tag{3.15}$$

Similarly, maximising $Q(\theta, \theta')$ with respect to $\sum_i$ we get $\sum_i$ as:

$$\sum_i = \frac{\sum_{n=1}^{N} \gamma(z_{1,i})(X_n - \mu_i)(X_n - \mu_i)^T}{\sum_{n=1}^{N} \gamma(z_{n,k})} \tag{3.16}$$

Similar update equations can be derived for other types of emission distributions, such as Bernoulli.

The objective of the E step is to efficiently compute the values of $\gamma(Z_n)$ and $\zeta(Z_{n-1}, Z_n)$. We will now delve into a detailed discussion on how these quantities can be evaluated effectively.

3.3.2 The forward-backward algorithm

The E step of the EM algorithm in a hidden Markov model can be efficiently performed using the alpha-beta algorithm, also known as the forward-backward algorithm or the Baum-Welch algorithm. This algorithm utilises a two-stage message-passing approach to obtain the posterior distribution of the latent variables. Different variants of the

algorithm exist, but we will focus on the widely used alpha-beta algorithm to obtain exact marginals.

The evaluation of posterior distributions of latent variables in a hidden Markov model does not depend on the form of the emission density or the type of observed variables (continuous or discrete). We only need the values of $P(X_n|Z_n)$ for each value of Z_n at each time step. In this section and the next, we will omit the explicit dependence on the model parameters, as they remain fixed throughout. We can start by considering the conditional independence properties.

$$P(X|Z_n) = P(X_1, \ldots, X_n|Z_n) \cdot P(X_{n+1}, \ldots, X_N|Z_n) \tag{3.17}$$

$$P(X_1, \ldots, X_{n-1}|X_n, Z_n) = P(X_1, \ldots, X_{n-1}|Z_n) \tag{3.18}$$

$$P(X_1, \ldots, X_{n-1}|Z_{n-1}, Z_n) = P(X_1, \ldots, X_{n-1}|Z_{n-1}) \tag{3.19}$$

$$P(X_{n+1}, \ldots, X_N|Z_n, Z_{n+1}) = P(X_{n+1}, \ldots, X_N|Z_{n+1}) \tag{3.20}$$

$$P(X_{n+2}, \ldots, X_N|Z_{n+1}, X_{n+1}) = P(X_{n+2}, \ldots, X_N|Z_{n+1}) \tag{3.21}$$

$$P(X|Z_{n-1}, Z_n) = P(X_1, \ldots, X_{n-1}|Z_{n-1}) \cdot P(X_n|Z_n)P(X_{n+1}, \ldots, X_N|Z_n) \tag{3.22}$$

The conditional independence properties, which can be proven using d-separation, show that in a hidden Markov model, the observed node X_n is conditionally independent of all previous observed nodes given the corresponding latent variable Z_n. This is because all paths from X_{n-1} to X_n pass through Z_n, and these paths are head-to-tail. Alternatively, these properties can be directly derived from the joint distribution of the hidden Markov model using the rules of probability.

To evaluate the posterior distribution of the latent variable Z_n given the observed data $X_1, \ldots, X_N$, we calculate $\gamma(Z_n)$. We can use Bayes' theorem to express $\gamma(Z_n)$ as the product of the likelihood $P(X|Z_n)$ and the prior $P(Z_n)$, divided by the denominator $P(X)$.

$$\gamma(Z_n) = P(Z_n|X) = \frac{P(X|Z_n)P(Z_n)}{P(X)} \tag{3.23}$$

Using the conditional independence property (3.17) and the product rule of probability,

we can express $\gamma(Z_n)$ in terms of the joint probabilities and conditional probabilities:

$$\gamma(Z_n) = \frac{P(X_1, \ldots, X_n, Z_n)P(X_{n+1}, \ldots, X_N|Z_n)}{P(X)} \tag{3.24}$$

We define two quantities, $\alpha(Z_n)$ and $\beta(Z_n)$, as follows:

$$\alpha(Z_n) = P(X_1, \ldots, X_n, Z_n) \tag{3.25}$$

$$\beta(Z_n) = P(X_{n+1}, \ldots, X_N|Z_n), \tag{3.26}$$

$\alpha(Z_n)$ represents the joint probability of observing all the data up to time n and the value of Z_n, while $\beta(Z_n)$ represents the conditional probability of future data from time $n + 1$ to N given Z_n. Both $\alpha(Z_n)$ and $\beta(Z_n)$ are vectors of length K, where K is the number of possible settings for Z_n.

Now, we derive recursion relations to efficiently evaluate $\alpha(Z_n)$ and $\beta(Z_n)$. We utilise conditional independence properties as defined above, along with the sum and product rules:

$$\begin{aligned}
\alpha(Z_n) &= P(X_1, \ldots, X_n, Z_n) \\
&= P(X_1, \ldots, X_n|Z_n) \cdot P(Z_n) \\
&= P(X_n|Z_n) \cdot P(X_1, \ldots, X_{n-1}|Z_n) \cdot P(Z_n) \\
&= P(X_n|Z_n) \cdot P(X_1, \ldots, X_{n-1}, Z_n) \\
&= P(X_n|Z_n) \sum_{Z_{n-1}} P(X_1, \ldots, X_{n-1}, Z_{n-1}, Z_n) \\
&= P(X_n|Z_n) \sum_{Z_{n-1}} P(X_1, \ldots, X_{n-1}, Z_n|Z_{n-1}) \cdot P(Z_{n-1}) \\
&= P(X_n|Z_n) \sum_{Z_{n-1}} P(X_1, \ldots, X_{n-1}|Z_{n-1}) \cdot P(Z_n|Z_{n-1}) \cdot P(Z_{n-1}) \\
&= P(X_n|Z_n) \sum_{Z_{n-1}} P(X_1, \ldots, X_{n-1}, Z_{n-1}) \cdot P(Z_n|Z_{n-1})
\end{aligned}$$

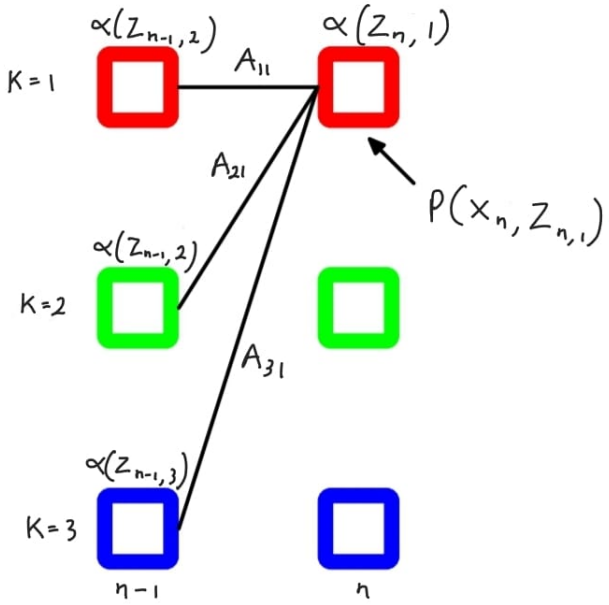
where the summation is over all possible values of Z_{n-1}. Using the equation (3.25) we

get:

$$\alpha(Z_n) = P(X_n|Z_n) \sum_{Z_{n-1}} \alpha(Z_{n-1}) \cdot P(Z_n|Z_{n-1}). \tag{3.27}$$

Let's take a closer look at this recursion relationship. The summation has K terms, and the right-hand side must be evaluated for each of the K values of Z_n. This indicates that the computing cost of each step of the α recursion scales like $O(K^2)$. Figure (3.3),taken from Bishop [3], shows a lattice diagram illustrating the forward recursion equation for $\alpha(Z_n)$.

Figure 3.3: Illustration of the forward recursion



To initiate this recursion, we require an initial condition given by:

$$\alpha(Z_1) = P(X_1, Z_1) = P(Z_1)P(X_1|Z_1) = \prod_{i=1}^{K} \{\pi_i P(X_1|\phi_i)\}^{z_{1,i}}. \tag{3.28}$$

Here, $\alpha(Z_{1,i})$ takes the value $\pi_i P(X_1|\phi_i)$ for $i = 1, \ldots, K$. By starting at the first node of the chain, we can progress along the chain and compute $\alpha(Z_n)$ for each latent node. Since each recursion step involves multiplying by a $K \times K$ matrix, the overall computational complexity of evaluating these quantities for the entire chain is $O(K^2N)$.

Similarly, we can derive a recursion relation for the quantities $\beta(z_n)$ by utilising the

conditional independence properties (13.27) and (13.28). We have:

$$\begin{aligned}
\beta(Z_n) &= P(X_{n+1}, \ldots, X_N | Z_n) \\
&= \sum_{Z_{n+1}} P(X_{n+1}, \ldots, X_N, Z_{n+1} | Z_n) \\
&= \sum_{Z_{n+1}} P(X_{n+1}, \ldots, X_N | Z_n, Z_{n+1}) \cdot P(Z_{n+1} | Z_n) \\
&= \sum_{Z_{n+1}} P(X_{n+1}, \ldots, X_N | Z_{n+1}) \cdot P(Z_{n+1} | Z_n) \\
&= \sum_{Z_{n+1}} P(X_{n+2}, \ldots, X_N | Z_{n+1}) \cdot P(X_{n+1} | Z_{n+1}) \cdot P(Z_{n+1} | Z_n).
\end{aligned}$$

This recursion relation allows us to compute $\beta(z_n)$ for each latent node by summing over the values of z_{n+1}. Using the equation (3.26), we get:

$$\beta(Z_n) = \sum_{Z_{n+1}} \beta(Z_{n+1}) \cdot P(X_{n+1} | Z_{n+1}) \cdot P(Z_{n+1} | Z_n). \tag{3.29}$$

We can employ a backward message passing algorithm to efficiently evaluate $\beta(Z_n)$ in terms of $\beta(Z_{n+1})$. At each step, we incorporate the effect of observation X_{n+1} through the emission probability $P(X_{n+1}|Z_{n+1})$, multiply by the transition matrix $P(Z_{n+1}|Z_n)$, and then marginalise out Z_{n+1}. This process is illustrated in Figure (3.4),taken from Bishop [3].

To initiate the recursion, we require a starting condition for $\beta(Z_N)$. By setting n = N in equation (3.24) and replacing $\alpha(Z_N)$ with its definition from equation (3.25), we obtain:

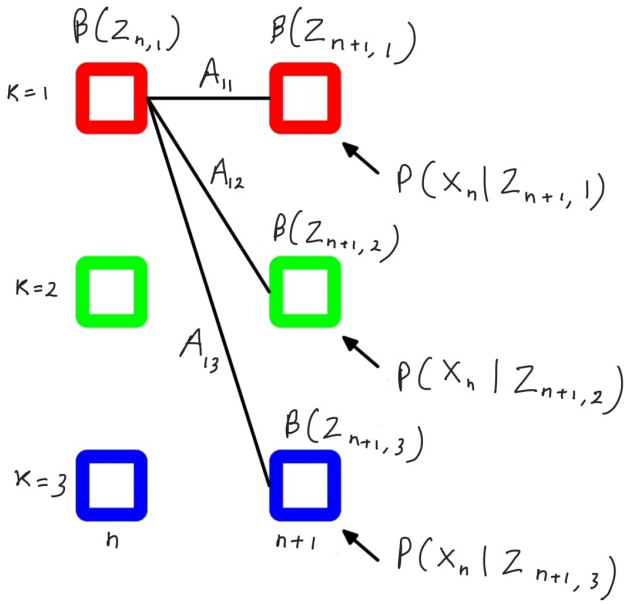
$$P(Z_N | X) = \frac{P(X, Z_N) \cdot \beta(Z_N)}{P(X)} \tag{3.30}$$

We observe that this equation holds true when $\beta(Z_N)$ is set to 1 for all possible settings of Z_N.

In the M-step equations, the quantity $P(X)$ cancels out. For example, the M-step equation for μ_i, as given by equation (3.15), takes the form:

$$\mu_i = \frac{\sum_{n=1}^{N} \gamma(z_{n,i}) \cdot X_n}{\sum_{n=1}^{N} \gamma(z_{n,i})} = \frac{\sum_{n=1}^{N} \alpha(z_{n,i}) \cdot \beta(z_{n,i}) \cdot X_n}{\sum_{n=1}^{N} \alpha(z_{n,i}) \cdot \beta(z_{n,i})} \tag{3.31}$$

Figure 3.4: Illustration of the backward recursion



However, it is useful to evaluate the likelihood function $P(X)$. By summing both sides of equation (3.24) over Z_n and considering the fact that the left-hand side represents a normalised distribution, we obtain:

$$P(X) = \sum_{Z_n} \alpha(Z_n) \cdot \beta(Z_n) \tag{3.32}$$

This allows us to monitor the value of the likelihood function throughout the EM optimisation process.

To evaluate the likelihood function, we can compute the sum of $\alpha(Z_N)$ over all possible values of Z_N. This gives us $P(X)$, the probability of observing the given data. We can conveniently calculate $P(X)$ by running the forward α recursion from the start to the end of the chain. However, it is important to note that we are required to use the assumption that $\beta(Z_N)$ is a vector of 1s.

$$P(X) = \sum_{Z_N} \alpha(Z_N) \tag{3.33}$$

Interpreting $P(X)$, each value of Z in the sum represents a particular choice of hidden state for each time step. By expressing the likelihood function as shown above, we reduce the computational cost from exponential to linear, summing contributions from

all paths passing through each state at each time step.

Next, let's consider the evaluation of $\zeta(Z_{n-1}, Z_n)$, which represents the conditional probabilities of the hidden states Z_{n-1} and Z_n given the observed data X. We can calculate $\zeta(Z_{n-1}, Z_n)$ directly using the results of the α and β recursions:

$$\begin{aligned}
\zeta(Z_{n-1}, Z_n) &= P(Z_{n-1}, Z_n|X) \\
&= \frac{P(X|Z_{n-1}, Z_n) \cdot P(Z_{n-1}, Z_n)}{P(X)} \\
&= \frac{P(X_1, \ldots, X_{n-1}|Z_{n-1}) \cdot P(X_n|Z_n) \cdot P(X_{n+1}, \ldots, X_N|Z_n) \cdot P(Z_n|Z_{n-1}) \cdot P(Z_{n-1})}{P(X)} \\
&= \frac{\alpha(Z_{n-1}) \cdot P(X_n|Z_n) \cdot P(Z_n|Z_{n-1}) \cdot \beta(Z_n)}{P(X)}
\end{aligned}$$

where we have made use of the conditional independence property stated earlier together with the definitions of $\alpha(Z_n)$ and $\beta(Z_n)$ given by (3.25) and (3.26).

As discussed earlier, to train a hidden Markov model using the EM algorithm, we start with an initial selection of the parameters $\theta' = (\pi, A, \phi)$. The A and π parameters can be initialised uniformly or randomly, while the parameters ϕ depend on the specific distribution. We run the forward and backward recursions to calculate $\gamma(Z_n)$ and $\zeta(Z_{n-1}, Z_n)$. Additionally, we can evaluate the likelihood function at this stage.

Once we have completed the E-step and obtained the values of $\gamma(Z_n)$ and $\zeta(Z_{n-1}, Z_n)$, we move on to the M-step to update the parameters θ'. We do this by maximising the expected complete-data log-likelihood, as explained earlier. We continue alternating between the E and M steps until we reach a point where the change in the likelihood function becomes very small.

It's important to note that the way we incorporate the observed data is through the conditional distributions $P(X_n|Z_n)$. However, the specific type or characteristics of the observed variables, as well as the exact form of the conditional distribution, do not affect the recursion process. As long as we can calculate the conditional distribution for each possible state of Z_n, we can apply the recursions. Also, since the observed variables are fixed, we can compute the values of $P(X_n|Z_n)$ based on Z_n at the beginning of the EM algorithm and keep them constant throughout the iterations. For a more detailed algorithmic description and implementation considerations, please refer to the appendix section.

Chapter 4

Methodology and Implementation

In this chapter, we will present the methodology that was followed to develop a predictive model for stock price prediction using hidden Markov models (HMMs). This methodology is based on the prediction approach proposed by Nguyet Nguyen in the paper titled ”Hidden Markov Model for Stock Trading” [2].

4.1 Determining the Optimal Number of Hidden States

In this section, we look at how to determine the optimal number of hidden states for our hidden Markov model for forecasting stock prices. The optimal number of hidden states must be chosen carefully since it directly influences the model's capacity to capture the underlying patterns and dynamics in stock market data.

To begin our analysis, we used the monthly Nifty-50 data from january 2000 to june 2021. The head of data is shown in table (4.1).

Open	High	Low	Close
1482.15	1671.15	1482.15	1546.20
1546.20	1818.15	1521.40	1654.80
1661.50	1773.85	1489.10	1528.45
1528.70	1636.95	1311.30	1406.55
1410.00	1436.60	1201.50	1380.45

Table 4.1: Nifty-50 monthly data from 31 January 2000 to 31 May 2000

4.1.1 Evaluation Metrics: AIC, BIC, HQC, and CAIC

To assess the performance and select the optimal number of hidden states, we have used four model selection criteria: Akaike Information Criterion (AIC), Bayesian Information Criterion (BIC), Hannan-Quinn Criterion (HQC), Bozdogan Consistent Akaike Information Criterion (CAIC)

The AIC, BIC, HQC, and CAIC are statistical metrics that balance model fit and complexity. To avoid overfitting, they penalise models with higher numbers of parameters. Lower values of these criteria indicate better model performance. We calculate the AIC, BIC, HQC, and CAIC values according to the following formulas:

$$AIC = -2 * \log(L) + 2 * k, \tag{4.1}$$

$$BIC = -2 * \log(L) + k * \log(N), \tag{4.2}$$

$$HQC = -2 * \log(L) + k * \log(\log(N)), \tag{4.3}$$

$$CAIC = -2 * \log(L) + k * (\log(N) + 1), \tag{4.4}$$

Where L is the likelihood function for the model, N is the number of observation points, and k is the number of estimated hidden parameters in the model, we assume that the distribution corresponding to each hidden state is Gaussian, hence the number of parameters. k is defined as k = $H^2 + 2H - 1$, where H is the number of states employed in the HMM.

4.1.2 Training Process

To train the HMM parameters, we have used historical observed data of fixed length T. The data consist of four components of stock prices: Open, Low, High, and Close prices, denoted as X_{open_t}, X_{low_t}, X_{high_t}, and X_{close_t}, respectively, where t represents the time index ranging from 1 to T. Mathematically, the sequence of observations can be seen as follows:

$$X = \{X_{open_t}, X_{low_t}, X_{high_t}, X_{close_t}, t = 1, 2, \ldots, T\}$$

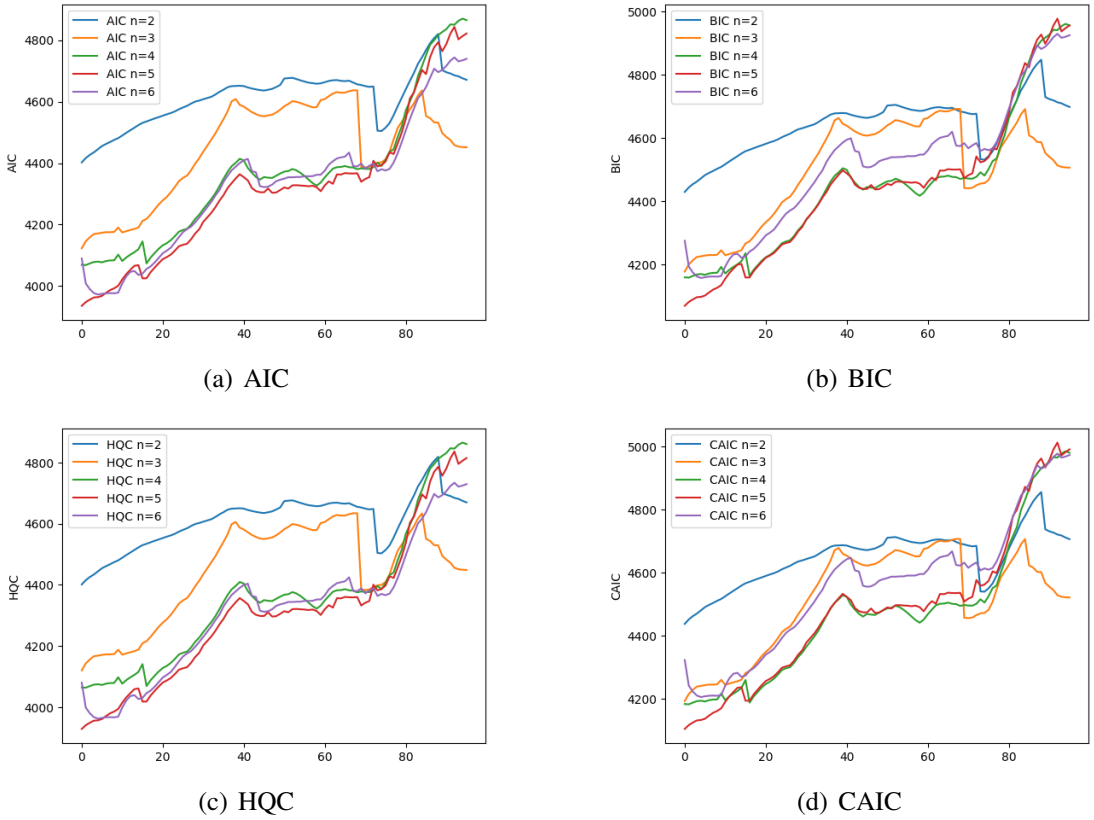
The data set was split into training and testing sets, with the training set comprising the initial period of fixed length T = 96 of the observed data. In this study, we utilised the Gaussian HMM, a type of HMM suitable for modelling continuous observations.

To evaluate the performance of different HMMs, we iteratively trained and tested models for various numbers of hidden states. For each iteration, we consider a block of data with a fixed length T = 96, representing a 8-years. Specifically, we focus on the Nifty-50 monthly prices from June 2005 to June 2013 as the first block. This block is used to calibrate the HMM parameters using the Baum-Welch algorithm. Next, we utilise the obtained parameters to calculate the likelihood (L) of the model by employing the forward algorithm. Finally, based on the likelihood, we compute the AIC, BIC, HQC, and CAIC values using the formulas (4.1) - (5.4).

To perform subsequent calibrations, we shift the ten-year data by one month, resulting in a new dataset from July 2005 to July 2013. We use the calibrated parameters from the previous calibration as the initial parameters for each iteration. This process is repeated 96 times, moving the data block forward each time. The final block of data covers the monthly prices from June 2013 to June 2021. For each model trained using a specific number of hidden states, we calculate the AIC, BIC, HQC, and CAIC values to pick the model with the best balance between goodness of fit and model complexity.

4.1.3 Results

After repeating the calibration process for each block of data and computing the AIC, BIC, HQC, and CAIC, we obtain a series of criterion values. These values are plotted in Figures (4.1), where a lower criterion value indicates a better model fit. From the results in Figures (4.1), we observe that the 5-state HMM outperforms the HMMs according to all four criteria. Therefore, we determine that the HMM with 5 hidden states provides the optimal model for stock price prediction and trading. However, it is important to note that the optimal number of states may vary for different stocks. Thus, we emphasise the importance of using these criteria to select the most appropriate number.



(a) AIC (b) BIC

(c) HQC (d) CAIC

Figure 4.1: AIC, BIC, HQC and CAIC for 96 HMM's parameter calibrations using Nifty-50 monthly prices.

4.2 Stock Price Prediction using HMM

We will now explain the process of predicting stock prices using hidden Markov models. This prediction method involves three main steps.

Step 1: Calibration of HMM Parameters

First, we will calibrate the parameters of the HMM using training data and calculate the likelihood of observing the data set. The parameters are determined based on the training data from a fixed time period, which we refer to as the training window D. Specifically, we use the training data from time T - D + 1 to T to calibrate the HMM's parameters (denoted by θ). In this process, we assume that the observation probability follows a Gaussian distribution. The initial HMM parameters for calibration are calculated using a Gaussian HMM. The training data consists of four sequences: open, low, high, and closing prices, denoted as:

$$X = \{X_{open_t}, X_{low_t}, X_{high_t}, X_{close_t}, \text{t} = \text{T - D + 1,T - D + 2,}\ldots\text{,T}\}$$

Step 2: Finding Similar Data

In the second step, we shift the data block one month back to obtain new observation data, $X_{new} = \{X_{open_t}, X_{low_t}, X_{high_t}, X_{close_t}\}$ for t = T - D, T - D + 1, ..., T - 1, and calculate the probability of observing this new data, $P(X_{new}|\theta)$. We continue moving the data block backward month by month until we find a dataset $X' = \{X'_{open_t}, X'_{low_t}, X'_{high_t}, X'_{close_t}\}$ for t = T - D + 1, T - D, ..., T, such that $P(X'|\theta)$ is similar to $P(X|\theta)$.

Step 3: Stock Price Prediction

In the final step, we predict the closing price of the stock at time T + 1, denoted as $X_{close_{T+1}}$. We calculate this prediction using the following formula:

$$X_{close_{T+1}} = X_{close_T} + ((X'_{close_{T+1}} - X'_{close_T}) * sign(P(X|\theta) - P(X'|\theta))), \tag{4.5}$$

where $X_{close_{T+1}}$ is the closing price of the stock at time T + 1 in the found dataset X', and sign() is the sign function. Similarly, for predicting for time T + 2, we will add the real observed data $X = \{X_{open_t}, X_{low_t}, X_{high_t}, X_{close_t}\}$ at time T + 1 to the dataset, and by following a similar process, we can predict the stock price for time T + 2 using new training data $X = \{X_{open_t}, X_{low_t}, X_{high_t}, X_{close_t}\}$ for t = T - D, T - D + 2, ..., T + 1. The calibrated HMM parameters (θ) from the first prediction are used as the initial parameters for subsequent predictions. We repeat this three-step prediction process for each subsequent prediction.

For convenience, we set the training window D equal to the out-of-sample forecast period. In practise, the length of the out-of-sample period can vary, but it is important to choose a proper length for the training window D based on the characteristics of the selected data, considering the efficiency of model simulations.

The out-of-sample data analysis was conducted to assess the performance of the HMM in predicting stock prices over a eight-year period (D = 96). In this study, Nifty-50 historical data from January 2000 to June 2013 was utilized to forecast stock prices from July 2013 to June 2021. As seen in Figure (5.1), the HMM demonstrated remarkable forecasting skills by efficiently capturing price variations throughout the 2020 - 2021 COVID crisis.

Chapter 5

Model Validation and Results

Due to the complex nature of financial markets, accurately predicting stock returns is a difficult task. Many researchers and practitioners rely on forecasting models to predict future stock returns, which can assist in making informed investment decisions. In this chapter, we introduce the historical average return model, which serves as a benchmark for evaluating the performance of other forecasting models. The model assumes that future returns will be equal to the average return observed in the historical data.

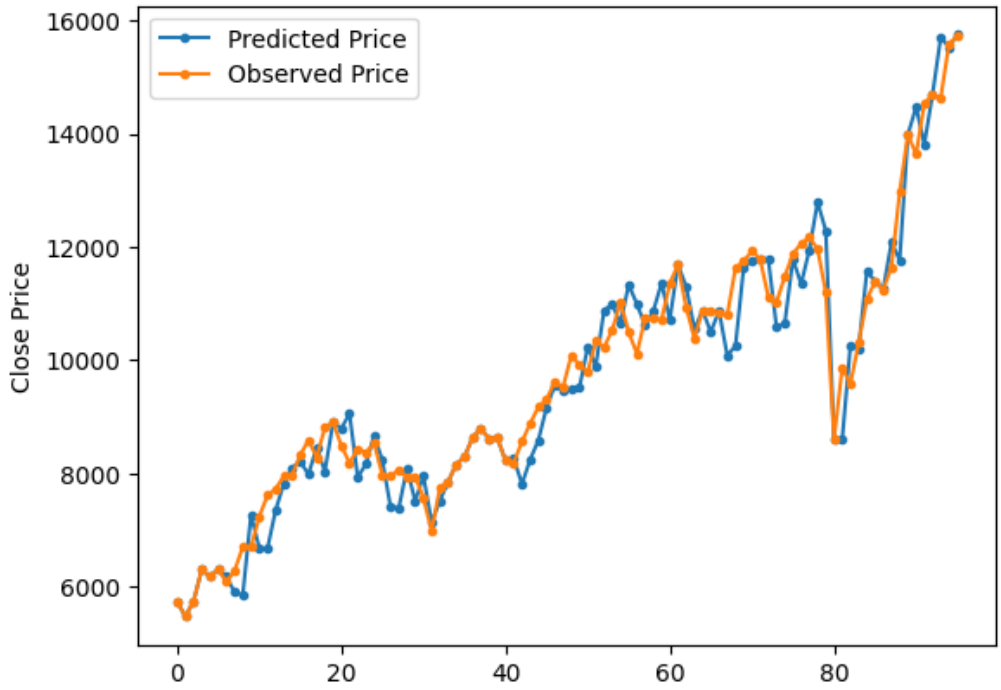


Figure 5.1: Predicted Nifty-50 monthly prices from July 2013 to June 2021 using five-state HMM.

5.1 Historical Average Model

To calculate the historical average return, we consider a time series of historical returns (R_t) for a given stock or market index. The historical average return ($\tilde{R}_t$) is computed as the arithmetic mean of the observed returns till time t:

$$\tilde{R}_{t+1} = \frac{1}{t}\sum_{i=1}^{t} R_i \tag{5.1}$$

where $\tilde{R}_i$ is the historical average returns of the observations, and the forecasted price at time $t + 1$ from this historical average return model is calculated as:

$$\tilde{P}_{t+1} = P_t(\tilde{R}_{t+1} + 1) \tag{5.2}$$

where P_t is the real observed price at time t.

5.2 Out of Sample R^2 Statistics

Evaluating the performance of these models is crucial to determining their effectiveness and identifying the most reliable approach. One commonly used method to assess forecasting models is the out-of-sample R^2 statistic. The out-of-sample R^2 statistic, introduced by Campbell and Thompson (2008), has become a popular measure to compare the performance of forecasting models. This statistic quantifies the reduction in mean squared predictive error (MSPE) between two models. By comparing the R^2 out of sample values (R^2_{OS}), researchers can determine if the desired model performs better than a competing model; here we compare it with the historical average return model.

To calculate the out-of-sample R^2_{OS}, the returns are divided into two sets: the in-sample data set and the out-of-sample data set. The first m points are allocated to the in-sample set, while the last q points constitute the out-of-sample set.

The out-of-sample R^2_{OS} is calculated by comparing the forecasted returns from a HMM model ($\hat{R}_{m+t}$) with the real returns (R_{m+t}) for each data point in the out-of-sample set. Mathematically, the out-of-sample R^2_{OS} is computed as the ratio of the sum of squared differences between the real returns (R_{m+t}), the forecasted return from our

HMM model ($\hat{R}_{m+t}$), and the forecasted returns from the competing model that is the historical average model ($\tilde{R}_{m+t}$) to the sum of squared differences between the real returns and the forecasted returns from the competing model, It can be expressed as:

$$R^2_{OSR} = 1 - \frac{\sum_{i=1}^{q}(R_{m+i} - \hat{R}_{m+i})^2}{\sum_{i=1}^{q}(R_{m+i} - \tilde{R}_{m+i})^2} \tag{5.3}$$

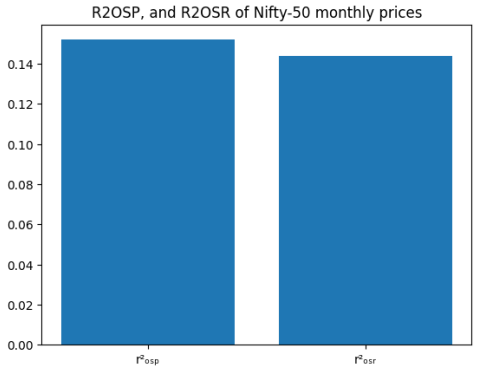
where, R^2_{OSR} represents the out-of-sample R^2 value for returns, m is the number of data points in the in-sample set, q is the number of data points in the out-of-sample set, R_{m+t} is the real return at time $m + t$, $\hat{R}_{m+t}$ is the forecasted return from the HMM model at time $m + t$, and $\tilde{R}_{m+t}$ is the forecasted return from the competing model at time $m + t$, which is calculated using (5.1).

The out-of-sample R^2 for stock prices based on predicted returns, denoted as R^2_{OSP}, can be calculated using the equation:

$$R^2_{OSP} = 1 - \frac{\sum_{i=1}^{q}(P_{m+i} - \hat{P}_{m+i})^2}{\sum_{i=1}^{q}(P_{m+i} - \tilde{P}_{m+i})^2} \tag{5.4}$$

where P_{m+t} represents the real stock price at time $m + t$,$\hat{P}_{m+t}$ is the forecasted price from the HMM model, and $\tilde{P}_{m+t}$ is the forecasted price based on the predicted return of the historical average return model as shown in (5.2). Figure (5.2) reveals that the

Figure 5.2: R^2_{OSP} and R^2_{OSR} for Nifty-50



computed R^2_{OSP} and R^2_{OSR} are positive, which indicates that the desires model performs better in terms of predicting accuracy than the HAR model. Based on the interpretation of these positive R^2_{OSP} and R^2_{OSR} values, we may conclude that the desired model outperforms the opposing model. This outcome gives evidence to the idea that the

desired model is more successful at capturing and forecasting stock market returns and prices.

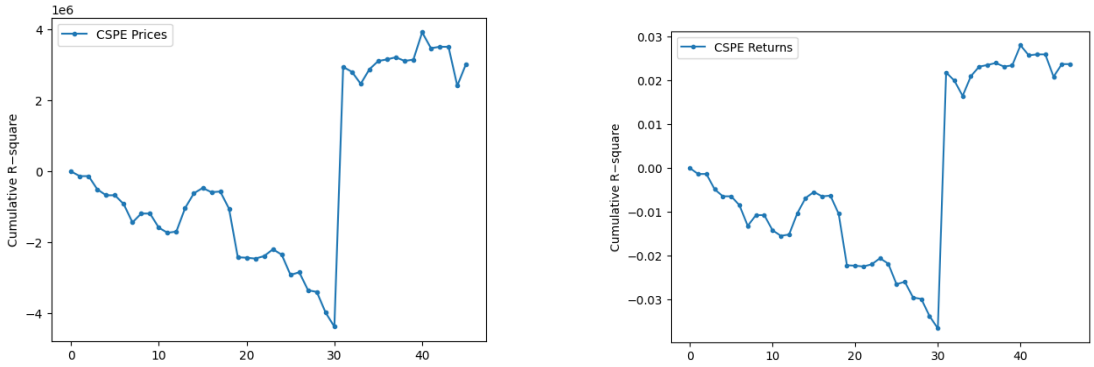
5.3 Cumulative Squared Predictive Errors (CSPEs)

While R^2_{OSP} and R^2_{OSR} compare the performance of two models on the whole out-of-sample forecasting period, they fail to provide insight into each model's efficiency for individual point predictions. To solve this, we use the cumulative squared predictive errors (CSPEs) introduced by Zhu and Zhu (2013) to compare the performance of the two models after each prediction.

The CSPE statistic at time $m + t$, denoted by $CSPE_t$ is calculated as:

$$CSPE_t = \sum_{i=m+1}^{t} [(R_i - \tilde{R}_i)^2 - (R_i - \hat{R}_i)^2] \tag{5.5}$$

The CSPE statistic reflects the difference in squared errors between the predicted values from the historical average model ($\tilde{R}_i$)and the HMM ($\hat{R}_i$) model. By observing the trend of the cumulative squared predictive errors, we can determine the relative performance of the two models. An increasing function indicates that the HMM outperforms the historical average model, while a decreasing function suggests that the historical average model performs better within that time interval. Additionally, by replacing the return prices in the formula, we can calculate the CSPE for prices.



(a) CSPE of forecasted price

(b) CSPE of forecasted return

Figure 5.3: CSPE of Nifty-50 monthly forecasted price and forecasted returns

Figures (5.3) show the CSPE for both predicted prices and returns, indicating that

while the Historical Average Return (HAR) model initially outperforms the Hidden Markov Model (HMM), the Cumulative Squared Predictive Errors (CSPEs) exhibit positive values and an upward trend after a certain period of the out-of-sample period. Consequently, we can infer that the HMM model outperforms the HAR model in terms of out-of-sample predictions.

5.4 Evaluating Performance: HMM vs. Historical Average Model in Stock Price Prediction

In this section, we compare the performance of the HMM and the HAR model using four standard error estimators: Absolute Percentage Error (APE), Average Absolute Error (AAE), Average Relative Percentage Error (ARPE), and Root-Mean-Square Error (RMSE). The error estimators are calculated using the following formulas:

$$APE = \frac{1}{\bar{R}} \sum_{i=i}^{T} \frac{|R_i - R'_i|}{T}, \tag{5.6}$$

$$AAE = \sum_{i=i}^{T} \frac{|R_i - R'_i|}{T}, \tag{5.7}$$

$$ARPE = \frac{1}{T} \sum_{i=i}^{T} \frac{|R_i - R'_i|}{T}, \tag{5.8}$$

$$RMSE = \sqrt{\frac{1}{T} \sum_{i=i}^{T} (R_i - R'_i)^2}, \tag{5.9}$$

Here, T represents the number of simulated points, R_i represents the real stock price (or stock return), R'_i represents the estimated price (or return), and $\bar{R}$ represents the mean of the sample.

We compute the prediction errors for both the Hidden Markov Model (HMM) and the Historical Average Return (HAR) models using the error estimators, taking into

account the anticipated returns (as per Equation (5.3)) and expected prices (as per Equation (5.4)). We use the R^2_{OS} statistic to compare the efficiency of the HMM with the HAR model and define the efficiency metric as:

$$\eta = 1 - \frac{E_{HMM}}{E_{HAR}}, \tag{5.10}$$

where η denotes the efficiency, E_{HMM} denotes the error of the HMM, and E_{HAR} denotes the error of the HAR model. By applying the equation (5.6) - (5.9), we calculate the errors of the two models and compare the HMM's efficiency to that of the HAR model. An efficiency number greater than zero shows that the HMM outperforms the HAR model. Table (5.1) displays the acquired results.

Table 5.1: Error Estimators and Efficiency Comparison

Error Estimators	APE	AAE	ARPE	RMSE
Predicted Return				
HMM	3.51908	0.04045	0.00087	0.05411
HAR	3.60469	0.04141	0.00090	0.05868
Efficiency	0.02374	0.02301	0.02301	0.07784
Predicted Price				
HMM	0.03820	449.47391	9.77117	595.54603
HAR	0.03921	460.85388	10.01856	648.51926
Efficiency	0.02572	0.02469	0.02469	0.08168

All error estimators show a positive efficiency, showing that the HMM model outperforms the HAR model, this data clearly supports the superiority of the HMM model over the HAR model in terms of prediction accuracy.

Chapter 6

Conclusion

6.1 Summary of Findings

In this thesis, we explored the application of a Hidden Markov Model (HMM) for stock price prediction. The study aimed to evaluate the performance of the HMM model in forecasting stock prices and compare it to the Historical Average Return (HAR) model, which served as a benchmark. The analysis was conducted on Nifty-50 historical data from January 2000 to June 2021, with the out-of-sample period spanning from July 2013 to June 2021.

The three-step prediction process of the HMM model involved data calibration, finding similar data, and stock price prediction. The HMM model demonstrated remarkable forecasting skills, capturing price variations throughout the challenging 2020-2021 COVID crisis. By comparing the HMM model's predictions with the actual stock prices, we assessed its accuracy and effectiveness.

To evaluate the performance of the HMM model, we employed various evaluation metrics. The out-of-sample R^2 statistic was used to compare the predictive power of the HMM model with that of the HAR model. The results indicated that the HMM model outperformed the HAR model, providing more accurate predictions of stock returns and prices.

We also examined the Cumulative Squared Predictive Errors (CSPEs) to assess the relative performance of the HMM and HAR models over time. The CSPE analysis revealed that while the HAR model initially outperformed the HMM model, the HMM

model exhibited positive CSPE values and an upward trend after a certain period in the out-of-sample period. This indicated that the HMM model surpassed the HAR model in terms of predictive accuracy.

Furthermore, we compared the prediction errors of the HMM and HAR models using various error estimators, including Absolute Percentage Error (APE), Average Absolute Error (AAE), Average Relative Percentage Error (ARPE), and Root-Mean-Square Error (RMSE). The calculations consistently showed that the HMM model had lower prediction errors and higher efficiency compared to the HAR model.

6.2 Contributions and Implications

The findings of this study have significant implications for stock market prediction and investment decision-making. The application of the HMM model provided more accurate forecasts of stock returns and prices compared to the traditional HAR model, which relied solely on historical average returns. This indicates that incorporating hidden states and probabilistic modeling, as done in the HMM, can enhance the predictive capabilities of stock price models.

The superior performance of the HMM model in capturing price variations during the COVID crisis highlights its robustness in volatile and unpredictable market conditions. This suggests that the HMM model could be particularly valuable for risk management and portfolio optimization strategies, where accurate predictions are crucial.

The evaluation metrics used in this study, such as the out-of-sample R^2 statistic, CSPE analysis, and error estimators, provide researchers and practitioners with valuable tools for comparing the performance of different stock price prediction models. These metrics enable a comprehensive assessment of model effectiveness and can guide the selection of appropriate forecasting models in real-world scenarios.

Appendix A

A.1 Algorithms

A.1.1 Forward Algorithm

Input: Observation sequence $X = (X_1^{(l)}, X_2^{(l)}, \ldots, X_N^{(l)})$

Output: Joint probability $P(X, Z)$

Initialization: Set the initial probability $\alpha_1^l(i) = \pi_i \cdot P(X_1^{(l)}|Z_i)$ for each state i;

for *each sequence* $l = 1, 2, \ldots, L$ **do**

  for $t = 2, 3, \ldots, N$ **do**

    for $i = 1, 2, \ldots, K$ **do**

      $\alpha_t^l(i) = \sum_{j=1}^{K} \alpha_{t-1}^l(j) \cdot P(Z_i|Z_j) \cdot P(X_t^{(l)}|Z_i)$;

    end

  end

  Calculate: Compute the probability of observation $P(X^{(l)}|\theta) = \sum_{i=1}^{K} \alpha_T^l(i)$;

  Update: Update the overall probability $P(X|\theta) = P(X|\theta) \cdot P(X^{(l)}|\theta)$;

end

Output: The final probability $P(X|\theta)$;

Algorithm 1: Forward Algorithm

A.1.2 Backward Algorithm

Input: Observation sequence $X = (X_1^{(l)}, X_2^{(l)}, \ldots, X_N^{(l)})$

Output: Joint probability $P(X, Z)$

Initialization: Set the backward probabilities $\beta_N^l(i) = 1$ for each state i;

for *each sequence* $l = 1, 2, \ldots, L$ **do**

  for $t = N-1, N-2, \ldots, 1$ **do**

    for $i = 1, 2, \ldots, K$ **do**

      $\beta_t^l(i) = \sum_{j=1}^{K} \beta_{t+1}^l(j) \cdot P(Z_j|Z_i) \cdot P(X_{t+1}^{(l)}|Z_j)$;

    end

  end

  Calculate: Compute the probability of observation $P(X^{(l)}|\theta) = \sum_{i=1}^{K} \beta_1^l(i)$;

  Update: Update the overall probability $P(X|\theta) = P(X|\theta) \cdot P(X^{(l)}|\theta)$;

end

Output: The final probability $P(X|\theta)$;

Algorithm 2: Backward Algorithm

References

[1] Lawrence R. Rabiner, *"A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition"*, Proceedings of the IEEE, Vol. 77, No. 2 (February 1989)

[2] Nguyet Nguyen*"Hidden Markov Model for Stock Trading"* International Journal of Financial Studies, (March 2018).

[3] Christopher M. Bishop *"Pattern Recognition and Machine Learning "*, Springer, (August 17, 2006)

[4] Xiaolin Li, Marc Parizeau, Réjean Plamondon *"Training Hidden Markov Models with Multiple Observations- A combinatorial Method"*, IEEE Transactions on Pattern Analysis and Machine Intelligence , Vol. 22, No. 4 (April, 2000)