PREDICT TREND OF UNEMPLOYMENT RATE BY OBSERVING THE NASDAQ COMPOSITE INDEX USING HIDDEN MARKOV MODEL

by

Li Liu, Bachelor of Science

A Thesis Presented in Partial Fulfillment
of the Requirements of the Degree
Master of Science

COLLEGE OF ENGINEERING AND SCIENCE
LOUISIANA TECH UNIVERSITY

August 2021

LOUISIANA TECH UNIVERSITY

GRADUATE SCHOOL

June 23, 2021
Date of thesis defense

We hereby recommend that the thesis prepared by

Li Liu

entitled **PREDICT TREND OF UNEMPLOYMENT RATE BY OBSERVING THE NASDAQ COMPOSITE INDEX USING HIDDEN MARKOV MODEL**

be accepted in partial fulfillment of the requirements for the degree of

Master of Science in Mathematics

Xiyuan Liu
Supervisor of Thesis Research

David Irakiza
Head of Mathematics & Statistics

Thesis Committee Members:
Xiyuan Liu
Songming Hou
Jonathan Bruce Walters

Approved:

Hisham Hegab
Dean of Engineering & Science

Approved:

Ramu Ramachandran
Dean of the Graduate School

GS Form 13
(01/20)

ABSTRACT

In this paper, we present how to use Hidden Markov Models (HMM) to predict the change in Unemployment Rate. By observing the NASDAQ price difference, we will predict the Unemployment Rate will rise or fall in the next month using the Hidden Markov Model. We will use the NASDAQ price and the unemployment rate from 2016 to 2020 monthly data for this paper. When we check the relation between the NASDAQ price and the unemployment rate, we find out that whenever the NASDAQ price goes up, the unemployment rate will drop for a time. So we are interested in how we can build Hidden Markov Models (HMM) by using the data we have. Furthermore, we want to see if our model is more accurate than other models that predict the unemployment rate.

APPROVAL FOR SCHOLARLY DISSEMINATION

The author grants to the Prescott Memorial Library of Louisiana Tech University the right to reproduce, by appropriate methods, upon request, any or all portions of this Thesis. It is understood that "proper request" consists of the agreement, on the part of the requesting party, that said reproduction is for his personal use and that subsequent reproduction will not occur without written approval of the author of this Thesis. Further, any portions of the Thesis used in books, papers, and other works must be appropriately referenced to this Thesis.

Finally, the author of this Thesis reserves the right to publish freely, in the literature, at any time, any or all portions of this Thesis.

Author _____

Date _____

DEDICATION

This thesis is dedicated to Dr. Xiyuan Liu for his kind care and careful guidance.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENT

I would like to express my most sincere gratitude to my thesis advisor, Professor Xiyuan Liu! Mr. Liu has not only taught me by example in academics, but also by his high moral character. The writing of this paper is directly due to his careful guidance. I would like to give my sincere gratitude to Dr. Songming Hou for his kind support. I am grateful to all my teachers for their guidance and selfless help in my professional studies during my college years. I would like to thank all the teachers who have reviewed this thesis. Your guidance and criticism are my motivation for further study and research.

In addition, I must thank my parents. I cannot repay the kindness of raising me. As their child, I have adhered to their simple, tough character. This is why I have enough confidence and ability to overcome the difficulties and hardships on the road ahead. It is also because of their hard work day and night that I had the opportunity to complete my education as I wished and then get the opportunity to develop further.

Also, I wish to thank my girlfriend, Yun Liu, for her encouragement and care. Without their trust, I could not make such significant progress.

CHAPTER 1

INTRODUCTION

During the Covid-19 pandemic, the US Unemployment Rate rose perpendicularly, it became 14.8 in April 2020, which is the highest rate in recent ten years. Also, we note that along with the rise in the unemployment rate, the NASDAQ correspondingly began to fall severely. Hence, the question arises if there is a Hidden Markov Model that can be built between the unemployment rate and NASDAQ. When the NASDAQ shows certain characteristics, then the unemployment rate will rise or fall.

In general, a declining unemployment rate represents a healthy overall economic development and is conducive to currency appreciation; a rising unemployment rate represents a recession and is not conducive to currency appreciation. Suppose the unemployment rate is analyzed together with the inflation indicator of the same period. In that case, it is possible to know whether the economy is overheating at that time, whether it will constitute pressure to raise interest rates, or whether it is necessary to stimulate the development of the economy by cutting interest rates. Therefore, by predicting the rise or fall of the unemployment rate, we can predict socio-economic development. If we need to predict the rise or fall of the unemployment rate, we need to build a model to analyze the data and then predict the direction of the unemployment rate. Because of our needs, Hidden Markov Models came to our attention.

Hidden Markov Models had been used for the stock market prediction for years. The Hidden Markov Model can model hidden state transitions based on ordered observational data. The problem of stock forecasting can be seen to follow the same pattern. The price of a stock depends on many factors that are usually invisible to investors (hidden variables) (Nguyen 2018). Transitions between the underlying factors vary with company policies and decisions, financial conditions, and management decisions, all of which affect the price of a stock (observed data). Therefore, the Hidden Markov Model is naturally suited to price forecasting problems.

For our topic, we will use the NASDAQ price's monthly difference be the observations and the fall or rise of the unemployment rate be the hidden state, so we can use Hidden Markov Models to predict the unemployment rate trends by observing the NASDAQ price.

We divide the entire data set into two categories. The first data set is the training set that will be used to train the model. The second data set is the test set, which is used to provide an unbiased evaluation of the model. Separating the training data set from the test set can prevent overfitting. So, in this case, we divide the data set into two parts: one for training the model and one for evaluating the model.

The rest of the paper is organized as follows. In Section 2, we introduce the history and relationships of the Markov chain and the Hidden Markov Model. In Section 3, we give the details of the methodology of the Hidden Markov Model. In Section 4, we show the data description and prepossessing. Finally, in Section 5, we discuss the results and our conclusions.

CHAPTER 2

LITERATURE REVIEW

In January 1913, the Russian mathematician A.A. Markov spent hours sifting through patterns of vowels and consonants from Pushkin's novel to see if he could apply probability theory to poetry, which is now known as the Markov Chain (Basharin 2004).

Markov Chain makes the assumption of predicting the future state in a sequence, and the only factor that matters is the current state. All states before the current state do not effect on the future except through the current state (Diaconis 2009). For example, if we want to predict tomorrow's weather, it only depends on today's weather and not on yesterday's weather (Khiatani 2018).

Adding the observations and hidden states into a Markov Chain makes it a Hidden Markov Model. In the 1960s, Leonard E. Baum and other authors described the Hidden Markov Model in a series of statistical papers (Mor 2021).

Hidden Markov models are probabilistic models about time series, describing the process of generating a random sequence of unobservable states from a hidden Markov chain and then generating a random series of observations by generating associate degree observation for every status.

Since the 1980s, Hidden Markov models have been applied to speech recognition with significant success. In the 1990s, Hidden Markov Model was also introduced to computer text recognition and the core technology of mobile communication, multi-user

detection, and it has been applied in the fields of bioinformatics and troubleshooting. One of the Hidden Markov Model usages is to use the model to predict the stock market's tendency (Hassan 2005). Based on the stock market’s performance, the tendency can be set to bear market or bull market or ordinary, which is the hidden state. We can build a Hidden Markov Model to predict if the market will be a bear market, a bull market, or normal (Somani 2014).

The sequence of states randomly generated by the Hidden Markov Chain is called the state sequence; each state generates one observation, and the resulting random sequence of observations is called the observation sequence. Each position of the sequence can be considered as a moment.

CHAPTER 3

METHODOLOGY

3.1 Markov Chain

A Markov Chain is a discrete-time stochastic process in mathematics with Markov properties. In this process, given current knowledge or information, the past (the historical state before the present) is irrelevant for predicting the future (the future state after the present).

Markov processes with discrete time and state are called Markov chains, abbreviated as $q_i = q(i)$, $i = 0,1,2,\cdots$.

The Markov chain is a sequence of random variables q_1, q_2, q_3, ... The range of these variables, the set of all their possible values, is called the state space, and the values of q_i are the states at time n. If the conditional probability distribution of q_i with respect to past states is only a function of q_{i-1}, then

$$P(q_i = a|q_1, \cdots, q_{i-1}) = P(q_i = a|q_{i-1}) \qquad \textbf{Eq. 3-1}$$

Here q is the state in the process. This constant equation above can be regarded as a Markov property. The change in the state of a random variable in a Markov chain with time step is called evolution or transition. There are two ways to describe the structure of Markov chains: (1) transfer matrices and transfer diagrams (shown in Figure 3-1) and (2) the properties that Markov chains exhibit during the transfer process are defined.

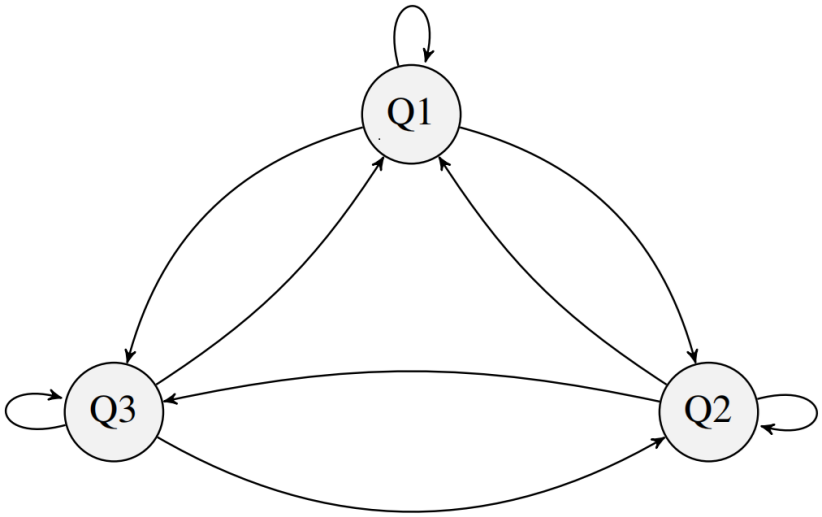

Figure 3-1: Markov Chain

In Figure 3-1, a graphic explanation of how the Markov Chain works is given. Assume that the current Markov chain has three states: $Q1, Q2, Q3$. Each state is transformed to the next state with a certain probability. For example, the state from $Q1$ is transformed to $Q2$ with a certain probability. Markov chain is such an arbitrary process that its future state distribution depends only on the present and has nothing to do with the past. In fact, it is a random variable that varies with time according to the Markov property.

3.2 Hidden Markov Model

The Hidden Markov Model is a kind of Markov chain whose states cannot be observed directly but can be observed by a sequence of observation vectors, each of which is expressed as various states by some probability density distribution, and each observation vector is generated by a sequence of states with corresponding probability density distribution. Therefore, the Hidden Markov Model is a dual stochastic process of

Hidden Markov Chains with a certain number of states and the set of displayed stochastic functions.

To have a better understanding of how the Hidden Markov Model works, let Q be the set of all possible states, V be the set of all possible observations, Y be a sequence of states and X is the corresponding sequence of observations.

$$Q = (q_1, q_2, \cdots, q_N), V = (v_1, v_2, \cdots, v_M) \quad \textbf{Eq. 3-2}$$

$$Y = (y_1, y_2, \cdots, y_T), X = (x_1, x_2, \cdots, x_T) \quad \textbf{Eq. 3-3}$$

Here, N is the number of possible states, M is the number of possible observations and T is the length of I. The state q is invisible and the observation v is visible.

Let A be a state transfer probability matrix:

$$A = [a_{ij}]_{N*N} \quad \textbf{Eq. 3-4}$$

where

$$a_{ij} = P(y_{t+1} = q_j \,|y_t = q_i), i = 1,2,\cdots,N;\ j = 1,2,\cdots,N \quad \textbf{Eq. 3-5}$$

Matrix A is the probability of transferring to state q_j at moment $t + 1$ under the condition of being in state q_i at moment t.

Let B be the observation probability matrix

$$B = [b_j(k)]_{N*M} \quad \textbf{Eq. 3-6}$$

where

$$b_j(k) = P(x_t = v_j \,|y_t = q_j), k = 1,2,\cdots,M;\ j = 1,2,\cdots,N \quad \textbf{Eq. 3-7}$$

Matrix B is the probability of getting an observation v_k conditional on moment t being in state q_j. We call matrix B the emission probability matrix.

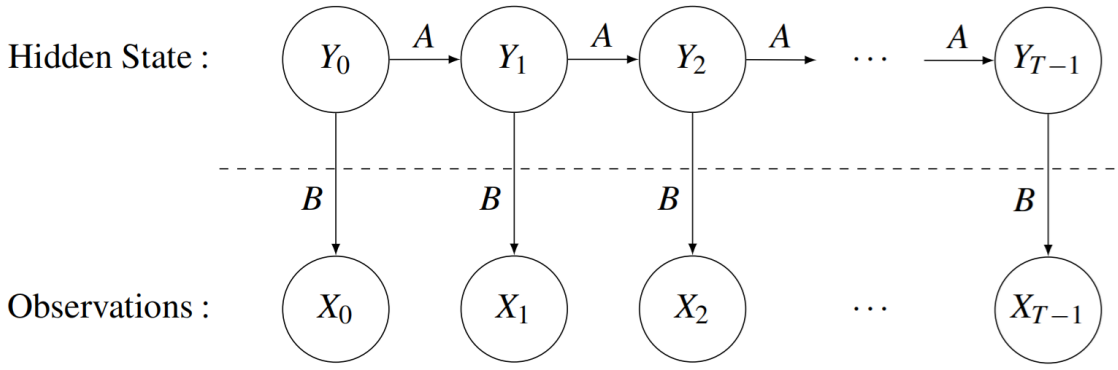

Figure 3-2: Hidden Markov Model

Let π be the initial state probability vector

$$\pi = (\pi_i) \qquad \textbf{Eq. 3-8}$$

where

$$\pi_i = P(y_1 = q_i), i = 1, \cdots, N \qquad \textbf{Eq. 3-9}$$

The Hidden Markov Model is determined by the initial state probability vector π, the state transfer probability matrix A, and the observation probability matrix B. π and A determine the state sequence, and B determines the observation sequence. Thus, the Hidden Markov Model can be expressed in ternary notation:

$$\lambda = (A, B, \pi) \qquad \textbf{Eq. 3-10}$$

These are the three elements of the Hidden Markov Model.

Given model $\lambda = (A, B, \pi)$ and observation sequence $O = (o_1, o_2, \cdots, o_T)$, calculate the probability of occurrence of observation sequence O under model λ:

$$P(O|\lambda) \qquad \textbf{Eq. 3-11}$$

If we already know the initial value π, and the observation sequence O, then use the forward and backward algorithm to estimate the Transition and Emission

Probabilities (Yu 2003). Then we can use the Viterbi Algorithm to predict the hidden states which generated the visible sequence.

In Hidden Markov Model, set Y be the hidden state and X be the observations. If we want to find the probability of $x_1 = o_1, x_2 = o_1, x_3 = o_2$, which is $P(x_1 = o_1, x_2 = o_1, x_3 = o_2)$. Here we put the hidden state in it, and we can use integrals to cancel it out. Here we can get:

$$P(x_1 = o_1, x_2 = o_1, x_3 = o_2)$$

$$= \sum_{y_1=1}^{N} \sum_{y_2=1}^{N} \sum_{y_3=1}^{N} P(x_1, x_2, x_3, y_1, y_2, y_3) \qquad \textbf{Eq. 3-12}$$

$P(x_1, x_2, x_3, y_1, y_2, y_3)$ can be written as:

$$P(x_3|x_1, x_2, y_1, y_2, y_3) * P(x_1, x_2, y_1, y_2, y_3) \qquad \textbf{Eq. 3-13}$$

By the Markov property:

$$P(x_3|x_1, x_2, y_1, y_2, y_3) * P(x_1, x_2, y_1, y_2, y_3)$$

$$= P(x_3|y_3) * P(x_1, x_2, y_1, y_2, y_3) \qquad \textbf{Eq. 3-14}$$

Repeat the process above, we can get:

$$P(x_3|y_3) * P(x_1, x_2, y_1, y_2, y_3)$$

$$= P(x_3|y_3) * P(y_3|y_2) * P(x_2|y_2) * P(y_2|y_1) * P(x_1|y_1) * P(y_1) \qquad \textbf{Eq. 3-15}$$

$P(x_i|y_i)$ is the emission probability, $P(y_{i+1}|y_i)$ is the transition probability, $P(y_1)$ is the initial value. So, we can find the probability of $x_1 = o_1, x_2 = o_1, x_3 = o_2$ with the three elements of the Hidden Markov Model.

Similarly, we can find:

$$P(x_1, x_2, \cdots, x_t\ |\lambda)$$

$$= \sum_{y_1=1}^{N} \sum_{y_2=1}^{N} \sum_{y_3=1}^{N} P(x_1, x_2, \cdots, x_t, y_1, y_2, \cdots, y_t)$$

$$= \sum_{y_1=1}^{N} \sum_{y_2=1}^{N} \sum_{y_3=1}^{N} P(q_1)P(x_1|y_1)P(y_2|y_1) \cdots P(x_t|y_t)P(y_t|y_{t-1}) \qquad \textbf{Eq. 3-16}$$

Set $a_{i,j} = P(y_t = j|y_{t-1} = i), b_j(x_t) = P(x_t|y_t = j)$. We can get from the above equation:

$$\sum_{y_1=1}^{N} \sum_{y_2=1}^{N} \sum_{y_3=1}^{N} \mathrm{P(y_1)P(x_1|y_1)P(y_2|y_1)} \cdots \mathrm{P(x_t|y_t)P(y_t|y_{t-1})}$$

$$= \sum_{y_1=1}^{N} \sum_{y_2=1}^{N} \sum_{y_3=1}^{N} \pi(\mathrm{y_1}) b_1(x_1) \prod_{\mathrm{t=2}}^{\mathrm{T}} a_{y_{t-1},y_t} b_{y_t(x_t)} \qquad \textbf{Eq. 3-17}$$

3.3 Forward-Backward Algorithm

Now we understand how Hidden Markov Model worked, but how can we get the 3 elements we need for Hidden Markov Model? More precisely, how can we calculate the transition and emission probability? Cause in the real life, we have no way to get them directly. So, we need to use the forward-backward algorithm.

Forward probability is the probability of being in state i at moment t and observing y_1 to y_t give that the Hidden Markov Model λ. Let $\alpha = P(x_1, x_2, \cdots, x_t, y_t = i|\lambda)$ which is $P(Y|\lambda) = \sum_{i=1}^{N} \alpha_{i(T)}$ This is the probability of partial sequence $x_1, \cdots, x_t$ and ending up in state i at time t.

Let $T = 1$, then we got:

$$\alpha_i(1) = P(x_1, y_1)$$

$$= P(x_1|y_1)P(y_1)$$

$$= b_i(x_1)\pi(y_1) \quad \textbf{Eq. 3-18}$$

Then $T = 2$, we got:

$$\alpha_j(2) = P(x_1, x_2, y_2)$$

$$= \sum_{i=1}^{N} P(x_1, x_2, y_1 = i, y_2 = j)$$

$$= \sum_{i=1}^{N} P(x_2|y_2)P(y_2|y_1)P(x_1, y_1)$$

$$= b_j(x_2) \sum_{i=1}^{N} a_{i,j}\, \alpha_i(1) \quad \textbf{Eq. 3-19}$$

From above, we can get that:

$$\alpha_j(T) = b_j(x_T)\left[\sum_{i=1}^{N} a_{i,j}\, \alpha_i(T-1)\right]$$

$$= P(x_1, \cdots, x_t, y_T = j) \quad \textbf{Eq. 3-20}$$

$$P(x_1, \cdots, x_T) = \sum_{j=1}^{N} \alpha_j(T) \quad \textbf{Eq. 3-21}$$

This is called the Forward Algorithm. There is another algorithm called the Backward Algorithm which is the time-reversed version of the Forward Algorithm. We need to find the probability of hidden state Y_i at time t, to generate the remaining part of the sequence of the visible symbol X^T.

Similarly, we let $\beta = P(x_{t+1}, x_{t+2}, \cdots, x_T|y_t = i, \lambda)$. This is at time t when the state is q_i, the probability of partial sequence of $x_{t+1}, \cdots, x_T$.

Using the same process, we did for the forward algorithm, we can also get the backward algorithm:

$$\beta_i(t) = P(x_{t+1} \cdots x_T \,|y_t = q_i)$$

$$= \sum_{j=0}^{N} P\big(x_{t+1} \cdots x_T, y_{t+1} = q_j \,\big|y_t = q\_i\big)$$

$$= \sum_{j=0}^{N} P\big(x_{t+2} \cdots x_T \big| x_{t+1}, y_{t+1} = q_j, = q_i\big) P\big(x_{t+1}, t_{t+1} = q_j \big| y_t = q_i\big)$$

$$= \sum_{j=0}^{N} P\big(x_{t+2} \cdots x_T \big| x_{t+1}, y_{t+1} = q_j, y_t = q_i\big) P\big(x_{t+1} \big| t_{t+1} = q_j, y_t = q_i\big)$$

$$P\big(y_{t+1} = q_j \big| y_t = q_i\big)$$

$$= \sum_{j=0}^{N} P\big(x_{t+2} \cdots x_T \big| y_{t+1} = q_j\big) P\big(x_{t+1} \big| t_{t+1} = q_j\big) P\big(y_{t+1} = q_j \big| y_t = q_i\big)$$

$$= \sum_{j=0}^{N} \beta_j(t+1) b_j(x_{t+1}) a_{i,j} \qquad \textbf{Eq. 3-22}$$

$$P(x_1, \cdots, x_T) = \sum_{j=1}^{N} \beta_j(1) \pi_j b_j(x_1) \qquad \textbf{Eq. 3-23}$$

Next, we can find out the relationship between the forward and backward algorithm:

$$P(y_t = q_i, X|\lambda)$$

$$= P(X|y_t = q_i, \lambda) P(y_t = q_i|\lambda)$$

$$= P(x_1 \cdots x_t, x_{t+1} \cdots x_T | y_t = q_i, \lambda)$$

$$= P(x_1 \cdots x_t | y_t = q_i, \lambda) P(x_{t+1} \cdots x_T | y_t = q_i, \lambda) P(y_t = q_i|\lambda)$$

$$= P(x_1 \cdots x_t | y_t = q_i, \lambda) P(x_{t+1} \cdots x_T | y_t = q_i, \lambda)$$

$$= \alpha_i(t) \beta_i(t) \qquad \textbf{Eq. 3-24}$$

From the conclusion we can easily see that at moment t if the state is known to be i, then this can be blocked before and after t and finally transformed into forward and backward probabilities.

Then we can find the probability of a single state:

$$\gamma_t(i) = P(y_t = q_i|X, \lambda)$$

$$= \frac{P(y_t = q_i, X|\lambda)}{P(X|\lambda)}$$

$$= \frac{\alpha_i(t)\beta_i(t)}{P(X|\lambda)}$$

$$= \frac{\alpha_i(t)\beta_i(t)}{\sum_{i=1}^{N}\alpha_i(t)\beta_i(t)} \quad \textbf{Eq. 3-25}$$

Think further: what if the joint probability of state i at moment t and state j at moment $\mathrm{t}+1$ is required? We would get the following:

$$\xi\_t(i,j) = P(y\_t = q\_i, y\_\{t+1\} = q\_j|X, \lambda))$$

$$= \frac{P(y_t, y_{t+1}, X|\lambda)}{P(X|\lambda)}$$

$$= \frac{P(y_t, y_{t+1}, X|\lambda)}{\sum_{i=1}^{N}\sum_{j=1}^{N}P(y_t, y_{t+1}, X|\lambda)}$$

$$= \alpha_i(t)a_{ij}b_j(x_{t+1})\beta_{t+1}(j) \quad \textbf{Eq. 3-26}$$

So we can have the marginal distribution $P(y_t = i, y_{t+1} = j|X)$

$$P\big(y_t = q_i, y_{t+1} = q_j\big|X\big) = \frac{\alpha_i(t)a_{ij}b_j(x_{t+1})\beta_{t+1}(j)}{P(X)} \quad \textbf{Eq. 3-27}$$

3.4 Viterbi Algorithm for Prediction

After using the forward-backward algorithm to get the transition matrix and emission matrix we need for the Hidden Markov model, we can use the Viterbi

Algorithm to predict the state. The Viterbi Algorithm is actually a dynamic programming solution to the Hidden Markov Model prediction problem, dynamic programming is used to find the probabilistic maximum path (optimal path). A path corresponds to a sequence of states (Churbanov 2008).

According to the dynamic programming principle, the optimal path has the property that if the optimal path passes through node A at moment t, then the partial path of this path from node A to endpoint B must be optimal for all partial paths from A to B. Based on this principle, we simply start from moment $t = 1$ and recursively calculate the maximum probability of each partial path with state i at moment t until we obtain the maximum probability of each path with state i at moment $t = T$. The maximum probability at moment $t = T$ is the probability of the optimal path P. The endpoint B of the optimal path is also obtained at the same time. After that, starting from the endpoint B, the previous nodes are obtained step by step from back to front.

$$\phi_t(i) = \max_{i_1,\cdots,i_{t-1}} P\left(x_1 x_2 \cdots x_t, y_1 y_2 \cdots y_{t-1}, y_t = q_i\right) \quad \textbf{Eq. 3-28}$$

which represents the highest probability that the first t observations are on a single path ending in state q_i.

Similarly, we can find that:

$$\phi_{t+1}(j) = \max_{i_1 \cdots i_t} P\left(x_1 x_2 \cdots x_{t+1}, y_1 y_2 \cdots y_t, y_{t+1} = q_j\right)$$

$$= \max_{1 \le i \le N} \phi_t\,(i) a_{ij} b_j x(t+1) \quad \textbf{Eq. 3-29}$$

So, we need to maximize $\phi_t(i)$ at each time step t to get the hidden states sequence.

$$argmax_t \phi_t(i) \quad \textbf{Eq. 3-30}$$

Then we will get the last hidden state, trace back the most likely hidden path. That is how Viterbi Algorithm works for the Hidden Markov Model.

CHAPTER 4

DATA DESCRIPTION

4.1 Data Definition

Once we have established the methodology, we begin to collect the necessary information. We collect NASDAQ from Yahoo Finance starting from Jan. 2016 to Dec. 2020. We also collected data on the unemployment rate for Federal Reserve Economic Data (FRED) on the same date. Since the NASDAQ price did not have the record every day due to the holiday and other situations, the daily dataset will have lots of missing data; this will have a significant impact on model parameters, data analysis, and forecasting, so we decide to use the monthly average price as a reference. Also, since the unemployment rate is recorded monthly, no modifications are needed for the data required.

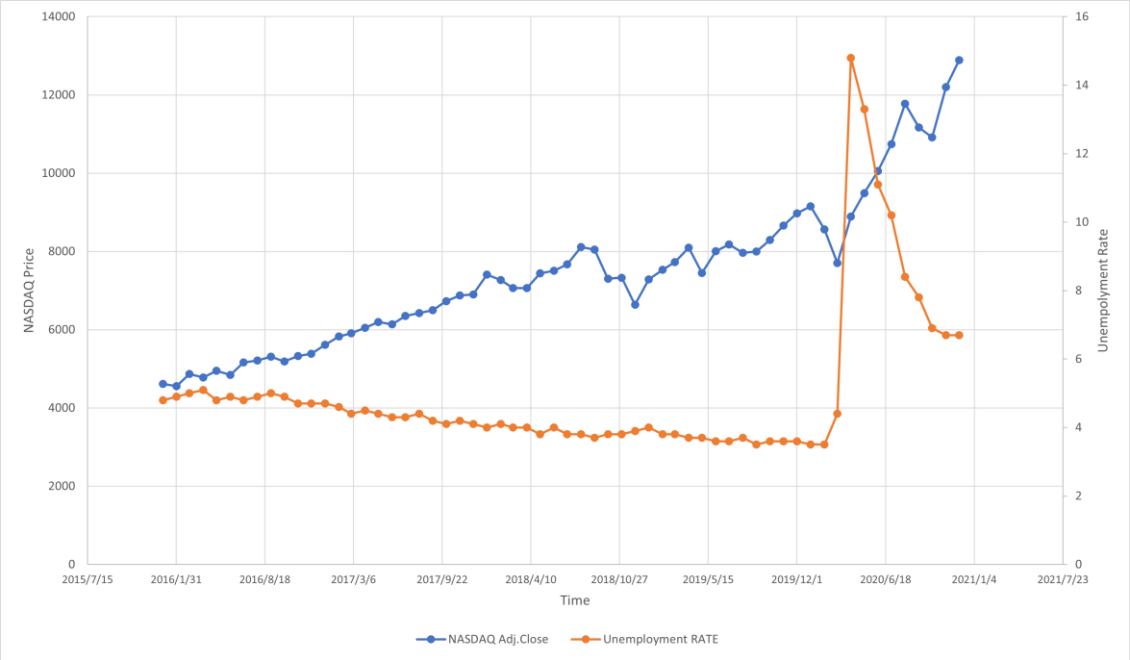

Figure 4-1 NASDAQ and Unemployment Rate Trend

Next, let us take a look at the NASDAQ and unemployment rate datasets' trend. The above is a graph showing the trend in NASDAQ data and the unemployment rate from 2016 to 2020.

From the Figure 4-1, we can see that the NASDAQ and the unemployment rate change in opposite trends, when the NASDAQ rises, the unemployment rate falls, and when the NASDAQ rises, the unemployment rate rises.

From this, we can see that the change in the NASDAQ (upward or downward) is a very good variable for predicting the evolution of the unemployment rate (upward or downward).

4.2 Data Pre-process

For data pre-processing, there are several features of the NASDAQ data:

- Date: The current month.

- Open: The price when the stock market opened on the first day of the current month.
- High: The highest price of the current month.
- Low: The lowest price of the current month.
- Close: The price when the stock market closed, the price is adjusted for splits.
- Adjusted Close: Adjusted close price adjusted for both dividends and splits.
- Volume: The monthly traded volume of the NASDAQ.

For our model, we will use the adjusted closing price for the observation feature. Since it is adjusted for both dividends and splits, the data will be more accurate for the analysis.

To use the Hidden Markov Model to predict the trend of the unemployment rate, we need to turn the data into different levels. Here we will use the difference between two adjacent months. For the NASDAQ price data, if the current month's data is subtracted from the previous month's data and the result is a negative number, set this to "F"(Fall), which means that the NASDAQ is decreasing from the previous month; if the result is a positive number, set this to "R"(Rise), which means that the NASDAQ is increasing from the previous month to this month. Similarly, for the unemployment rate data, let the current month's data subtract the previous month's data; and if the result is a negative number, set this as "D"(Down), which means that the unemployment rate is decreasing from the previous month to this month; if the result is a positive number, set this as "U"(Up), which means that the unemployment rate is increasing from the previous month.

After defining the fall/rise for NASDAQ and up/down for unemployment rate, we sort out the feature and data we need from the NASDAQ, which are the date and the Adjusted Close, then taking the difference for each month to month, we get the modified data shown below:

Table 4-1 NASDAQ trend

Date	**Trend**	**Date**	**Trend**	**Date**	**Trend**
2016/2/1	Fall	2017/10/1	Rise	2019/6/1	Rise
2016/3/1	Rise	2017/11/1	Rise	2019/7/1	Rise
2016/4/1	Fall	2017/12/1	Rise	2019/8/1	Fall
2016/5/1	Rise	2018/1/1	Rise	2019/9/1	Rise
2016/6/1	Fall	2018/2/1	Fall	2019/10/1	Rise
2016/7/1	Rise	2018/3/1	Fall	2019/11/1	Rise
2016/8/1	Rise	2018/4/1	Rise	2019/12/1	Rise
2016/9/1	Rise	2018/5/1	Rise	2020/1/1	Rise
2016/10/1	Fall	2018/6/1	Rise	2020/2/1	Fall
2016/11/1	Rise	2018/7/1	Rise	2020/3/1	Fall
2016/12/1	Rise	2018/8/1	Rise	2020/4/1	Rise
2017/1/1	Rise	2018/9/1	Fall	2020/5/1	Rise
2017/2/1	Rise	2018/10/1	Fall	2020/6/1	Rise
2017/3/1	Rise	2018/11/1	Rise	2020/7/1	Rise
2017/4/1	Rise	2018/12/1	Fall	2020/8/1	Rise
2017/5/1	Rise	2019/1/1	Rise	2020/9/1	Fall
2017/6/1	Fall	2019/2/1	Rise	2020/10/1	Fall
2017/7/1	Rise	2019/3/1	Rise	2020/11/1	Rise
2017/8/1	Rise	2019/4/1	Rise	2020/12/1	Rise
2017/9/1	Rise	2019/5/1	Fall		

Also, after taking the same steps for the unemployment rate data, we get the following data:

Table 4-2 Unemployment rate trend

Date	Trend	Date	Trend	Date	Trend
2016/2/1	Up	2017/10/1	Down	2019/6/1	Down
2016/3/1	Up	2017/11/1	Up	2019/7/1	Down
2016/4/1	Up	2017/12/1	Down	2019/8/1	Up
2016/5/1	Down	2018/1/1	Down	2019/9/1	Down
2016/6/1	Up	2018/2/1	Up	2019/10/1	Up
2016/7/1	Down	2018/3/1	Down	2019/11/1	Down
2016/8/1	Up	2018/4/1	Down	2019/12/1	Down
2016/9/1	Up	2018/5/1	Down	2020/1/1	Down
2016/10/1	Down	2018/6/1	Up	2020/2/1	Down
2016/11/1	Down	2018/7/1	Down	2020/3/1	Up
2016/12/1	Down	2018/8/1	Down	2020/4/1	Up
2017/1/1	Down	2018/9/1	Down	2020/5/1	Down
2017/2/1	Down	2018/10/1	Up	2020/6/1	Down
2017/3/1	Down	2018/11/1	Down	2020/7/1	Down
2017/4/1	Up	2018/12/1	Up	2020/8/1	Down
2017/5/1	Down	2019/1/1	Up	2020/9/1	Down
2017/6/1	Down	2019/2/1	Down	2020/10/1	Down
2017/7/1	Down	2019/3/1	Down	2020/11/1	Down
2017/8/1	Up	2019/4/1	Down	2020/12/1	Down
2017/9/1	Down	2019/5/1	Down		

After preprocessing the data, we combine the two sets of data using the NASDAQ index rise and fall as our observations, so that the unemployment rate rises and falls as the hidden states of the Hidden Markov Model. We then get the data set below:

Table 4-3 Combine table

Date	Obs.	State	Date	Obs.	State	Date	Obs.	State
2016/2/1	Fall	Up	2017/11/1	Rise	Up	2019/8/1	Fall	Up
2016/3/1	Rise	Up	2017/12/1	Rise	Down	2019/9/1	Rise	Down
2016/4/1	Fall	Up	2018/1/1	Rise	Down	2019/10/1	Rise	Up
2016/5/1	Rise	Down	2018/2/1	Fall	Up	2019/11/1	Rise	Down
2016/6/1	Fall	Up	2018/3/1	Fall	Down	2019/12/1	Rise	Down
2016/7/1	Rise	Down	2018/4/1	Rise	Down	2020/1/1	Rise	Down
2016/8/1	Rise	Up	2018/5/1	Rise	Down	2020/2/1	Fall	Down
2016/9/1	Rise	Up	2018/6/1	Rise	Up	2020/3/1	Fall	Up
2016/10/1	Fall	Down	2018/7/1	Rise	Down	2020/4/1	Rise	Up
2016/11/1	Rise	Down	2018/8/1	Rise	Down	2020/5/1	Rise	Down
2016/12/1	Rise	Down	2018/9/1	Fall	Down	2020/6/1	Rise	Down
2017/1/1	Rise	Down	2018/10/1	Fall	Up	2020/7/1	Rise	Down
2017/2/1	Rise	Down	2018/11/1	Rise	Down	2020/8/1	Rise	Down
2017/3/1	Rise	Down	2018/12/1	Fall	Up	2020/9/1	Fall	Down
2017/4/1	Rise	Up	2019/1/1	Rise	Up	2020/10/1	Fall	Down
2017/5/1	Rise	Down	2019/2/1	Rise	Down	2020/11/1	Rise	Down
2017/6/1	Fall	Down	2019/3/1	Rise	Down	2020/12/1	Rise	Down
2017/7/1	Rise	Down	2019/4/1	Rise	Down			
2017/8/1	Rise	Up	2019/5/1	Fall	Down			
2017/9/1	Rise	Down	2019/6/1	Rise	Down			
2017/10/1	Rise	Down	2019/7/1	Rise	Down			

The above steps complete the preprocessing of the data. Next, we will start training the parameters needed for the corresponding data, build up the Hidden Markov Model, and then use the model to predict the existing data.

4.3 Model Training

Before starting to build the model, we must separate the data to the training dataset and the test dataset. Generally, when we use data for the model training, we may choose the training data randomly so the model can be more accurate; however, in our case, because the Hidden Markov Model is time-related, therefore we cannot choose the

training data randomly. Thus, we use the data from 2016 to 2019 for the training dataset and the whole 2020 years' data as the testing dataset.

Following this, we begin to form the parameters of the Hidden Markov model: the transition probability and the emission probability. The transition probability matrix represents the probability that the unemployment rate will rise or fall in the next month when the previous month's state is rising or falling. In our initial Hidden Markov Model, we set the probability be 0.5 for each state. This means that the probability for the next month's state is a drop in the unemployment rate given that this month's state is an increase in the unemployment rate is 0.5. The initial transition probability is:

Table 4-4 Transition probability

	Down	**Up**
Down	0.5	0.5
Up	0.5	0.5

The emission probability matrix represents the probability that when the unemployment rate is rising, the probability of the NASDAQ Price to rise or fall. Take our model as an example, we set that when the unemployment rate is rising, the probability of the NASDAQ Price rise is 0.3, and when the unemployment rate is falling, the probability of the NASDAQ Price fall is 0.4. The initial emission probability is:

Table 4-5 Emission probability

	Down	**Up**
Down	0.4	0.6
Up	0.7	0.3

This setup makes sense for our research; the particular reason for the circumstance is, as we can see from the previous graph of the trend the NASDAQ and unemployment rate trend, when the unemployment rate rises, the probability of the NASDAQ rising is very low, and similarly, when the NASDAQ falls, the probability of the unemployment rate rising is also very low. This allows us to temporarily establish a negative correlation between the two.

After setting the initial value for the parameters, we can use the forward-backward algorithm to maximize the transition probability and the emission probability. The algorithm is shown above in the methodology section.

4.4 Forecasting

After determining the parameters of the Hidden Markov Model, we can use the new model to make predictions for our testing dataset which is to predict the trend of 2020 year's unemployment rate. We can then compare the result of our prediction and the actual situation.

From the training process, we get our new transition probability matrix:

Table 4-6 Transition probability

	Down	**Up**
Down	0.648798	0.351202
Up	0.868314	0.131686

Also, we get the new emission probability matrix:

Table 4-7 Emission probability

	Down	**Up**
Down	0.072929	0.927071
Up	0.68403	0.31597

With the model determined, we can now use Viterbi algorithm to forecast the monthly trend of the unemployment rate in 2020, before we use the new Hidden Markov Model for the 2020 dataset, we need to use it on the training dataset first to check the accuracy rate of the prediction. After the prediction of the training set, we get the result in the below table:

Table 4-8 Training set prediction with actual result

Date	**Prediction**	**Actual**	**Date**	**Prediction**	**Actual**
2016/2/1	Up	Up	2018/2/1	Up	Up
2016/3/1	Down	Up	2018/3/1	Down	Down
2016/4/1	Up	Up	2018/4/1	Down	Down
2016/5/1	Down	Down	2018/5/1	Down	Down
2016/6/1	Up	Up	2018/6/1	Down	Up
2016/7/1	Down	Down	2018/7/1	Down	Down
2016/8/1	Down	Up	2018/8/1	Down	Down
2016/9/1	Down	Up	2018/9/1	Down	Down
2016/10/1	Up	Down	2018/10/1	Up	Up
2016/11/1	Down	Down	2018/11/1	Down	Down
2016/12/1	Down	Down	2018/12/1	Up	Up
2017/1/1	Down	Down	2019/1/1	Down	Up
2017/2/1	Down	Down	2019/2/1	Down	Down
2017/3/1	Down	Down	2019/3/1	Down	Down
2017/4/1	Down	Up	2019/4/1	Down	Down
2017/5/1	Down	Down	2019/5/1	Up	Down
2017/6/1	Up	Down	2019/6/1	Down	Down
2017/7/1	Down	Down	2019/7/1	Down	Down
2017/8/1	Down	Up	2019/8/1	Up	Up
2017/9/1	Down	Down	2019/9/1	Down	Down
2017/10/1	Down	Down	2019/10/1	Down	Up
2017/11/1	Down	Up	2019/11/1	Down	Down
2017/12/1	Down	Down	2019/12/1	Down	Down
2018/1/1	Down	Down			

From the prediction result and the actual result of the unemployment rate in the training dataset, we can calculate the accuracy rate of the current Hidden Markov Model. By using the confusion matrix, we can get the table:

Table 4-9 Confusion matrix for training set prediction

Prediction	**Down**	**Up**
Down	26	9
Up	5	7

From the table, we can get that the true positive is 26, true negative is 7, false positive is 9, and the false negative is 5. We can now get the accurate rate of the prediction:

$$\frac{TP + TN}{TP + TN + FP + FN}$$

$$= \frac{26 + 7}{26 + 7 + 9 + 5}$$

$$= 0.7021$$

Now, we can use the new Hidden Markov Model to predict the unemployment rate for the 2020 dataset. After running the code, we get the results table below.

From the table below, we have the predicted results of 2020:

Table 4-10 Prediction for 2020

Date	Prediction
2020/1/1	Down
2020/2/1	Up
2020/3/1	Up
2020/4/1	Down
2020/5/1	Down
2020/6/1	Down
2020/7/1	Down
2020/8/1	Down
2020/9/1	Up
2020/10/1	Up
2020/11/1	Down
2020/12/1	Down

Now we can now compare the results with the real situation:

Table 4-11 2020 prediction and actual situation

Date	Prediction	Actual
2020/1/1	Down	Down
2020/2/1	Up	Down
2020/3/1	Up	Up
2020/4/1	Down	Up
2020/5/1	Down	Down
2020/6/1	Down	Down
2020/7/1	Down	Down
2020/8/1	Down	Down
2020/9/1	Up	Down
2020/10/1	Up	Down
2020/11/1	Down	Down
2020/12/1	Down	Down

Next, let us find the accuracy rate of the testing set. We get the below table of the prediction of 2020 by using the confusion matrix function:

Table 4-12 Confusion matrix for testing set prediction

Prediction	Down	Up
Down	7	1
Up	3	1

The result showed us that the true positive is 7, true negative is 1, false positive is 1, and the false negative is 3. Then the accuracy rate of the testing set prediction is:

$$\frac{TP + TN}{TP + TN + FP + FN}$$

$$= \frac{7 + 1}{7 + 1 + 1 + 3}$$

$$= 0.6667$$

We can see that our model's prediction accuracy rate for the testing set is around 67\%. The accuracy rate is close enough to the training set accuracy rate. Our Hidden Markov Model is based on time-related, so when external factors intervene, the model will have a certain prediction error. Since we only have 12 months to predict, that may make the accuracy rate easily affected.

So we try to use more data for the training and testing datasets. We then choose the NASDAQ price and unemployment rate data from 2009 to 2021 for our model. With the same procedure we did in the Data Pre-process section, we get our new data sets from 2009 to 2021. Then split the dataset to training and testing datasets. As before, since the Hidden Markov Model is related to time series, so we cannot split the data randomly, we have to choose the data by timeline. So we use the data from 2009 to 2018 be the training dataset to train our model and get the parameters, then use the data from 2019 to 2021 for the testing dataset for the new model prediction.

After training the model, we get the transition probability matrix and the emission probability for our Hidden Markov Model:

Table 4-13 Transition probability

	Down	**Up**
Down	0.5765922	0.4234078
Up	0.6261266	0.3738734

Table 4-14 Emission probability

	Down	**Up**
Down	0.2321136	0.7678864
Up	0.5711989	0.4288011

With the new fitted model, we start to make a prediction with the training set. Then, we get the result compare with the actual result:

Table 4-15 Prediction vs actual results for 2009-2018 data

Date	**Actual**	**Prediction**	**Date**	**Actual**	**Prediction**	**Date**	**Actual**	**Prediction**
2009/2/1	Up	Up	2012/6/1	Down	Down	2015/10/1	Down	Down
2009/3/1	Up	Up	2012/7/1	Down	Down	2015/11/1	Up	Down
2009/4/1	Up	Up	2012/8/1	Down	Down	2015/12/1	Down	Up
2009/5/1	Up	Down	2012/9/1	Down	Down	2016/1/1	Down	Down
2009/6/1	Up	Down	2012/10/1	Down	Down	2016/2/1	Up	Down
2009/7/1	Down	Up	2012/11/1	Down	Down	2016/3/1	Up	Down
2009/8/1	Up	Down	2012/12/1	Up	Down	2016/4/1	Up	Up
2009/9/1	Up	Up	2013/1/1	Up	Down	2016/5/1	Down	Down
2009/10/1	Up	Up	2013/2/1	Down	Down	2016/6/1	Up	Up
2009/11/1	Down	Down	2013/3/1	Down	Down	2016/7/1	Down	Down
2009/12/1	Down	Down	2013/4/1	Up	Up	2016/8/1	Up	Down
2010/1/1	Down	Up	2013/5/1	Down	Down	2016/9/1	Up	Up
2010/2/1	Down	Down	2013/6/1	Down	Down	2016/10/1	Down	Up
2010/3/1	Up	Down	2013/7/1	Down	Up	2016/11/1	Down	Down
2010/4/1	Down	Up	2013/8/1	Down	Up	2016/12/1	Down	Down
2010/5/1	Down	Up	2013/9/1	Down	Down	2017/1/1	Down	Down

Table 4-15 Continued

2010/6/1	Down	Up	2013/10/1	Down	Down	2017/2/1	Down	Down
2010/7/1	Down	Down	2013/11/1	Down	Down	2017/3/1	Down	Up
2010/8/1	Up	Up	2013/12/1	Down	Down	2017/4/1	Up	Up
2010/9/1	Down	Down	2014/1/1	Down	Up	2017/5/1	Down	Up
2010/10/1	Down	Down	2014/2/1	Up	Down	2017/6/1	Down	Down
2010/11/1	Up	Down	2014/3/1	Down	Up	2017/7/1	Down	Down
2010/12/1	Down	Down	2014/4/1	Down	Down	2017/8/1	Up	Down
2011/1/1	Down	Down	2014/5/1	Up	Down	2017/9/1	Down	Down
2011/2/1	Down	Down	2014/6/1	Down	Down	2017/10/1	Down	Up
2011/3/1	Down	Up	2014/7/1	Up	Down	2017/11/1	Up	Down
2011/4/1	Up	Down	2014/8/1	Down	Down	2017/12/1	Down	Up
2011/5/1	Down	Up	2014/9/1	Down	Up	2018/1/1	Down	Down
2011/6/1	Up	Up	2014/10/1	Down	Down	2018/2/1	Up	Up
2011/7/1	Down	Up	2014/11/1	Up	Down	2018/3/1	Down	Down
2011/8/1	Down	Up	2014/12/1	Down	Down	2018/4/1	Down	Down
2011/9/1	Down	Up	2015/1/1	Up	Up	2018/5/1	Down	Down
2011/10/1	Down	Down	2015/2/1	Down	Down	2018/6/1	Up	Up
2011/11/1	Down	Down	2015/3/1	Down	Down	2018/7/1	Down	Down
2011/12/1	Down	Up	2015/4/1	Down	Up	2018/8/1	Down	Down
2012/1/1	Down	Down	2015/5/1	Up	Down	2018/9/1	Down	Up
2012/2/1	Down	Down	2015/6/1	Down	Up	2018/10/1	Up	Down
2012/3/1	Down	Up	2015/7/1	Down	Down	2018/11/1	Down	Down
2012/4/1	Down	Up	2015/8/1	Down	Down	2018/12/1	Up	Up
2012/5/1	Down	Up	2015/9/1	Down	Down			

Next, we need to use the prediction results and the confusion matrix function to find the accuracy rate for the prediction. After the run the code, we get the table:

Table 4-16 Confusion matrix for training set prediction

Prediction	**Down**	**Up**
Down	55	20
Up	28	16

The table showed us that the true positive is 55, true negative is 16, false positive is 20, and the false negative is 28. With the number we have, we can find that the accuracy rate of the prediction is:

$$\frac{TP + TN}{TP + TN + FP + FN}$$

$$= \frac{55 + 16}{55 + 16 + 20 + 28}$$

$$= 0.5966$$

Next, we start to use the fitted model to predict the testing set we have, which is the 2019 to 2021 set. After the prediction, we get the following predictions to compare to the actual results in the following table:

Table 4-17 Prediction vs actual result for 2019 - 2021

Date	Actual	Prediction	Date	Actual	Prediction
2019/1/1	Up	Down	2020/3/1	Up	Up
2019/2/1	Down	Down	2020/4/1	Up	Down
2019/3/1	Down	Up	2020/5/1	Down	Down
2019/4/1	Down	Down	2020/6/1	Down	Down
2019/5/1	Down	Up	2020/7/1	Down	Down
2019/6/1	Down	Down	2020/8/1	Down	Down
2019/7/1	Down	Down	2020/9/1	Down	Up
2019/8/1	Up	Down	2020/10/1	Down	Up
2019/9/1	Down	Up	2020/11/1	Down	Down
2019/10/1	Up	Down	2020/12/1	Down	Down
2019/11/1	Down	Down	2021/1/1	Down	Down
2019/12/1	Down	Down	2021/2/1	Down	Down
2020/1/1	Down	Down	2021/3/1	Down	Down
2020/2/1	Down	Up	2021/4/1	Up	Down

The confusion matrix for the testing set prediction is the next value we need to calculate. After the code running, we get the matrix:

Table 4-18 Confusion matrix for testing set prediction

Prediction	Down	Up
Down	16	5
Up	6	1

The table showed us that the true positive is 16, true negative is 1, false positive is 5, and the false negative is 6. With the number we have, we can find that the accuracy rate of the prediction is:

$$\frac{TP + TN}{TP + TN + FP + FN}$$

$$= \frac{16 + 1}{16 + 1 + 5 + 6}$$

$$= 0.6071$$

After completing the models that were formed after analyzing the two sets of data, we obtained the prediction accuracy of the two models:

Table 4-19 Prediction accuracy

Accuracy	Training	Testing
Model I	0.7021	0.6667
Model II	0.5966	0.6071

The accuracy rate shows that the Model I is more accurate than the other model, which means that using more data for the training will not make the prediction more accurate. The reason for this phenomenon may be due to the time-related properties of the Hidden Markov Model and the Markov Chain assumption. Since the Hidden Markov Model is based on the Markov Chain, it means the current state is influenced only by the state of the previous moment, which means the if we use more data for the model fitting,

the model may be not accurate. The same situation happened in some time-series research: the more data used, the less accurate the prediction is for the time series model (Brown 1987).

Now, we have all the results and data we need for our conclusion.

CHAPTER 5

CONCLUSION AND FUTURE WORK

5.1 Conclusion

So far, we have predicted the unemployment rate's trend for all 12 months of 2020, with the Hidden Markov Model trained by the data from 2016 to 2019. Since we had some doubts about the accuracy rate of this model, we decided to use more data to train the model. Then we used the data from 2009 to 2018 as the training set for the model fitting, then used the new Hidden Markov Model to predict the unemployment rate for 2019 to 2021 and compare to the actual result, then got the accuracy rate for the new model.

From the result, we can see that the new model's accuracy rate is lower than the old model we trained, for both the training set and testing set. From this result, we can figure out the reason is that the model is over-fitting. Since the Hidden Markov Model is time related, and by the Markov Assumption, we can conjecture that our model is time-sensitive or even current time sensitive. This means that if we use too much data to train the model, our model will be less accurate. We need to use recent years' data for the training to have an accurate model.

The other thought is that since we only have 12 months to predict and we have a big impact of the coronavirus. In September and October 2020, the NASDAQ price was falling. The particular reason for this circumstance is that the coronavirus hit America

very hard in the previous months, so the stock market price kept falling, but in September and October, the increase in coronavirus cases slowed, so many companies started back to work; therefore, more person were getting hired than the previous months and that makes the unemployment rate start to drop.

For our research, we use the model trained by the 2016 to 2019 data to predict the 12 months' unemployment rate in 2020. If we exclude the effects of the pandemic, our prediction results are closer to the reality, and there 8 prediction results are the same with the actual situation; for the different predictions, maybe the effects of Covid19 on the economy caused the prediction failures or maybe our model training can be even more accurate. This leads to the possibility of future work in our study.

5.2 Future Work

In the future, after the pandemic, the NASDAQ price and unemployment rate will be more stable and normalized, so that will make our model more accurate. Therefore, the prediction will be more accurate based on the NASDAQ price.

Also, the other way that may make the model more precise is to use less current data. Like we stated earlier, our model is time-sensitive or current time-sensitive. So in the future, we can use less current data to train the model, maybe the model can predict the unemployment rate more accurately.

APPENDIX A

R CODE AND FIGURES

A.1 R Code

```
library(dplyr)
library(ggplot2)
library(reshape2)
library(HMM)
library(caret)
library(pROC)
#Import dataset
nas = read.csv("D:\\Data\\NASDAQ.csv", header = TRUE)
rate = read.csv("D:\\Data\\UNRATE.csv", header = TRUE)
str(nas)
str(rate)
#data preprocess
#select the variable we need
data = nas %>% select(Date, Adj.Close)
temp_ts = ts(data = data$Adj.Close, frequency = 12, start = c(2016,1))
plot.ts(temp_ts, ylab = 'NASDAQ Price(Monthly)',  main = "NASDAQ Price from 2016 to 2020",
type = "o")
temp_tsrate = ts(data = rate$UNRATE , frequency = 12, start = c(2016,1))
plot.ts(temp_tsrate21, ylab = 'Unemployment Rate(Monthly)', main = "Unemployment Rate from
2016 to 2020", type = "o")
#calculate the difference
NASdata = diff(data$Adj.Close)
date = data$Date[-1]
#combine date and data together.
data = data.frame(date = date, Diff = NASdata)
#if difference is nagetive or 0 then Fall, else Rise.
data$Diff = ifelse(NASdata <= 0, "F", "R")
data #observation
#remove the last row since NAS data only have until 2020-12-01
rate = rate[-61,]
#calculate the difference
ratediff = diff(rate$UNRATE)
ratedate = rate$DATE[-1]
#combine date and data together.
ratec = data.frame(date = ratedate, ratediff = ratediff)
#if difference is nagetive or 0 then rate go Down, else rate go up.
ratec$ratediffe = ifelse(ratec$ratediff <= 0, "D", "U")
rate = subset(ratec, select = -c(ratediff))
```

```
rate #state
#creat the data table
df = data.frame(NAS = data$Diff, Rate = rate$ratediffe)
#Split data, use 2016 to 2019 data to be the training set, 2020's data be the testing set.
train <- df[c(1:47),]
test <- df[c(48:59),]
#set the inital HMM
hmm = initHMM(c("D","U"),c("F","R"),
              transProbs=matrix(c(.5,.5,.5,.5),2),
              emissionProbs=matrix(c(.4,.7,.6,.3),2))
observations = train$NAS
#Max transprob and emissionprob use the training set
bw = baumWelch(hmm,observations,30)
#use viterbi algorthm and new hmm to predict the training and testing set and compare with the real
answer.
predtrain = viterbi(bw$hmm,train$NAS)
predtrain
train$Rate
data.frame(train$Rate)
##Accuracy rate for training set
xtabtrain <- table(predtrain, train$Rate)
confusionMatrix(xtabtrain)
##Prediction for Testset
pred = viterbi(bw$hmm,test$NAS)
pred
test$Rate
##Accuracy rate for testing set
xtabtest <- table(pred, test$Rate)
confusionMatrix(xtabtest)
#Use more data for training and testing
nas21 = read.csv("D:\\Data\\NDAQ09-21.csv", header = TRUE)
rate21 = read.csv("D:\\Data\\UNRATE09-21.csv", header = TRUE)
str(nas21)
str(rate21)
#data preprocess
#select the variable we need
data21 = nas21 %>% select(Date, Adj.Close)
temp_ts21 = ts(data = data21$Adj.Close, frequency = 12, start = c(2009,1))
plot.ts(temp_ts21, ylab = 'NASDAQ Price(Monthly)', main = "NASDAQ Price from 2009 to 2021",
type = "o")
temp_tsrate21 = ts(data = rate21$UNRATE , frequency = 12, start = c(2009,1))
plot.ts(temp_tsrate21, ylab = 'Unemployment Rate(Monthly)', main = "Unemployment Rate from
2009 to 2021", type = "o")
#calculate the difference
NASdata21 = diff(data21$Adj.Close)
date21 = data21$Date[-1]
#combine date and data together.
data21 = data.frame(date = date21, Diff = NASdata21)
#if difference is nagetive or 0 then Fall, else Rise.
data21$Diff = ifelse(NASdata21 <= 0, "F", "R")
data21 #observation
#calculate the difference
ratediff21 = diff(rate21$UNRATE)
```

```
ratedate21 = rate21$DATE[-1]
#combine date and data together.
ratec21 = data.frame(date = ratedate21, ratediff = ratediff21)
#if difference is nagetive or 0 then rate go Down, else rate go up.
ratec21$ratediffe21 = ifelse(ratec21$ratediff <= 0, "D", "U")
rate21 = subset(ratec21, select = -c(ratediff))
rate21 #state
#creat the data table
df21 = data.frame(NAS = data21$Diff, Rate = rate21$ratediffe21)
train21 <- df21[c(1:119),]
test21 <- df21[c(120:147),]
hmm = initHMM(c("D","U"),c("F","R"),
            transProbs=matrix(c(.5,.5,.5,.5),2),
            emissionProbs=matrix(c(.4,.7,.6,.3),2))
observations = train21$NAS
bw = baumWelch(hmm,observations,30)
#use viterbi algorthm and new hmm to predict the training and testing set and compare with the real
answer.
predtrain21 = viterbi(bw$hmm,train21$NAS)
predtrain21
train21$Rate
data.frame(train21$Rate)
#Accuracy rate for training set
xtabtrain <- table(predtrain21, train21$Rate)
confusionMatrix(xtabtrain)
##Prediction for Testset
pred21 = viterbi(bw$hmm,test21$NAS)
pred21
test21$Rate
#Accuracy rate for testing set
xtabtest <- table(pred21, test21$Rate)
confusionMatrix(xtabtest)
```

A.2 Figures

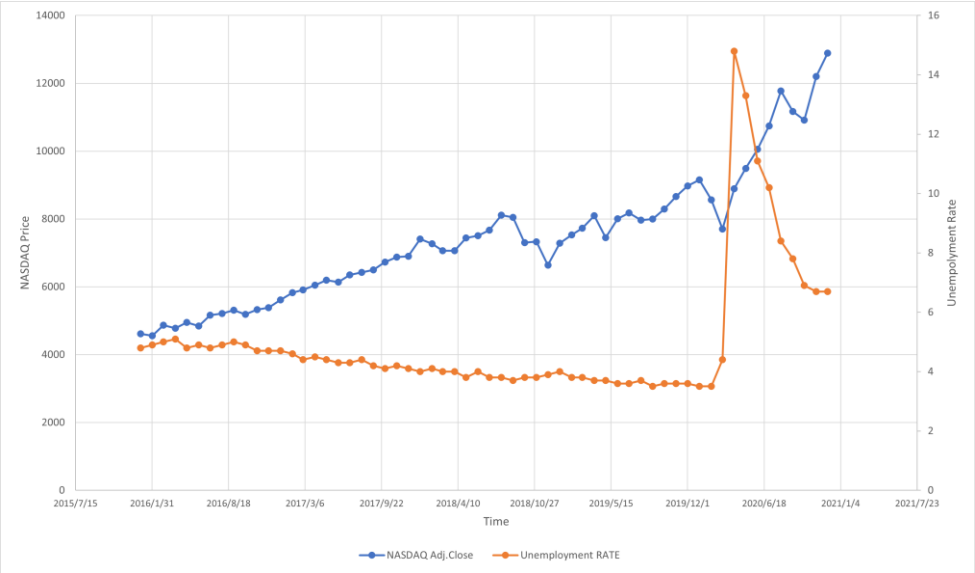

Figure A-1 NASDAQ and Unemployment Rate Trend

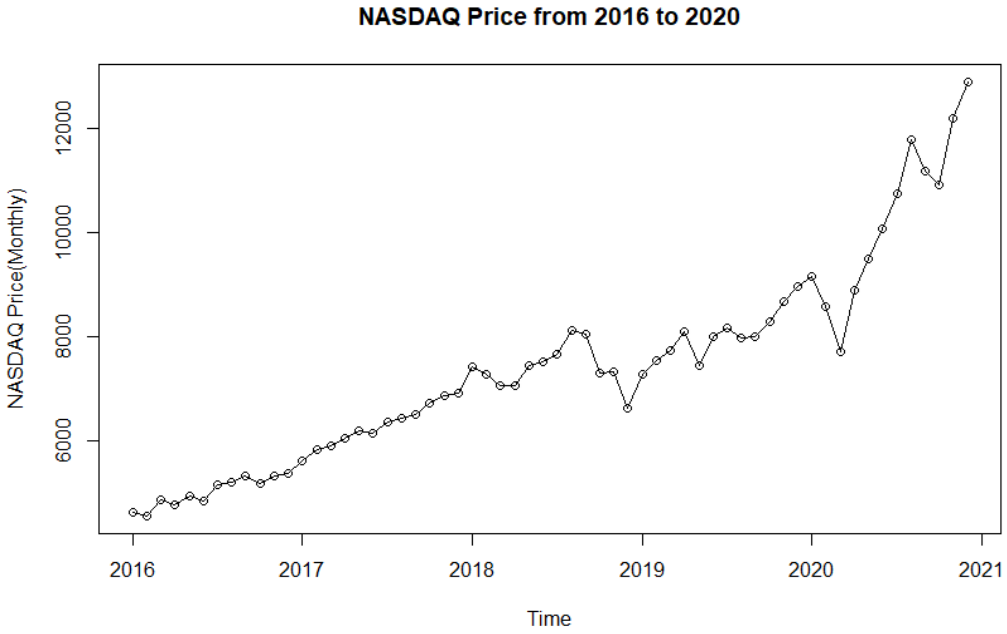

Figure A-2 NASDAQ Price from 2016 to 2020

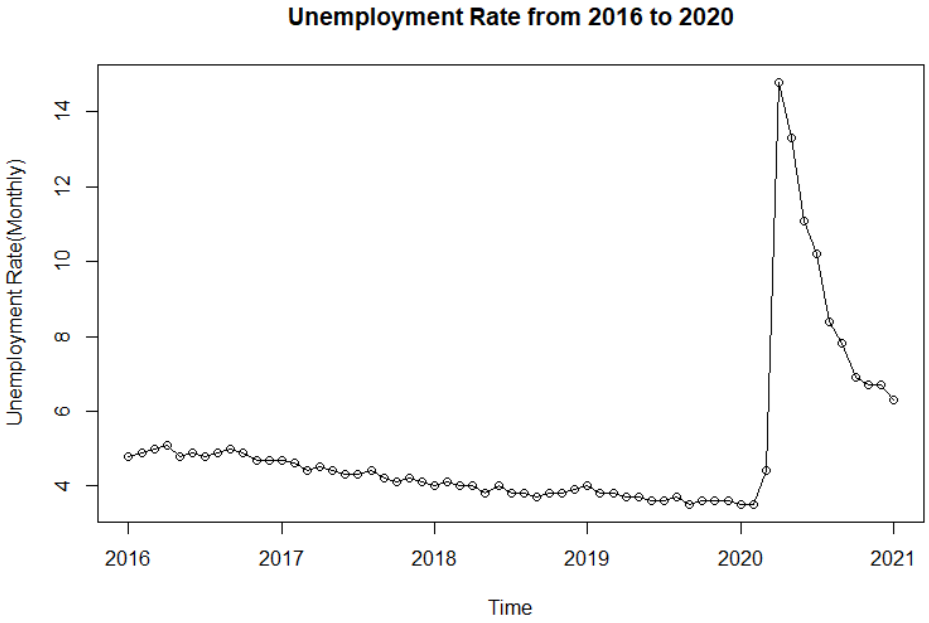

Figure A-3 Unemployment Rate from 2016 to 2020

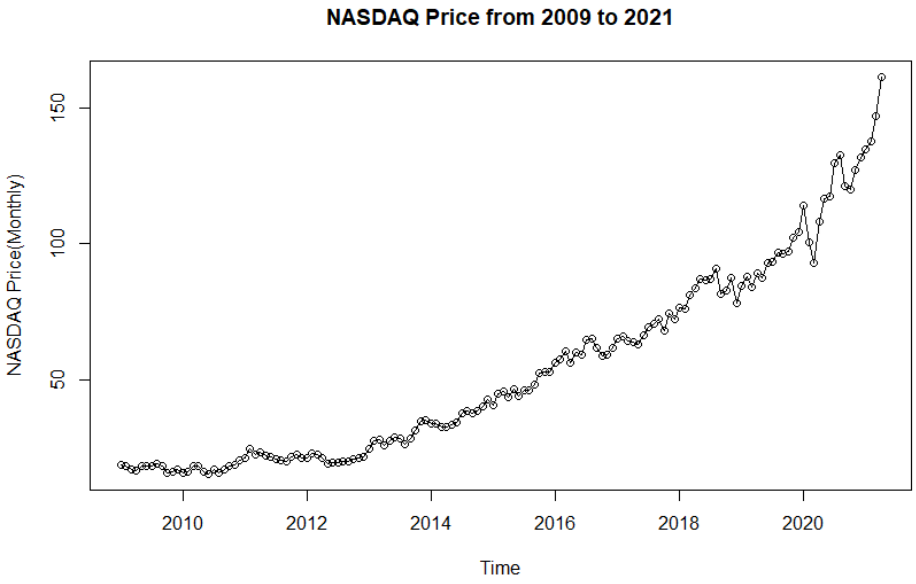

Figure A-4 NASDAQ Price from 2009 to 2020

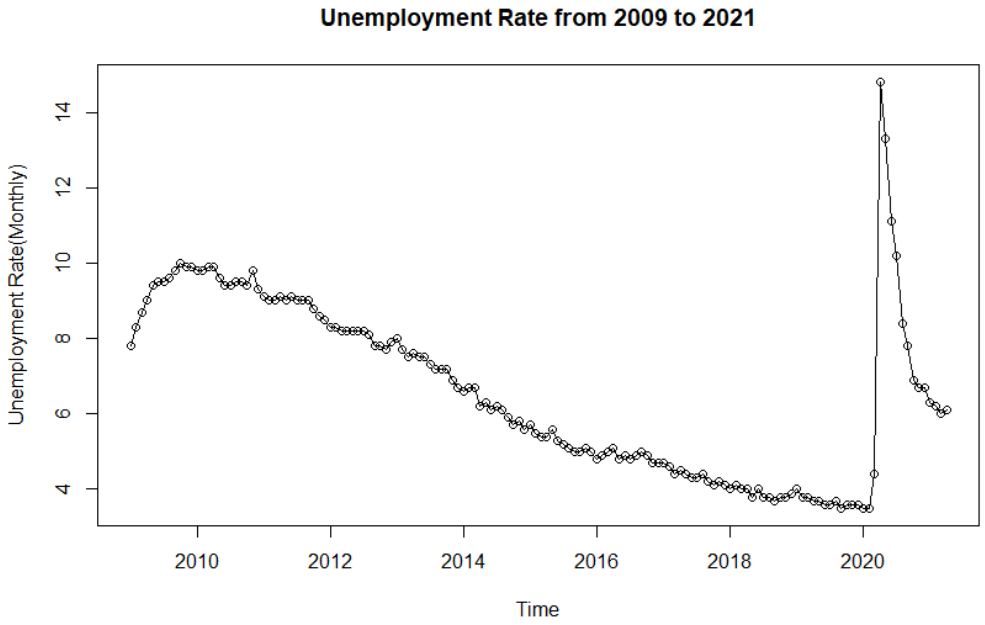

Figure A-5 Unemployment Rate from 2009 to 2020

BIBLIOGRAPHY

Basharin, Gely P and Langville, Amy N and Naumov, Valeriy A. "The life and work of AA Markov." *Linear algebra and its applications*, 2004: 3-26.

Brown, Lawrence D and Hagerman, Robert L and Griffin, Paul A and Zmijewski, Mark E. "Security analyst superiority relative to univariate time-series models in forecasting quarterly earnings." *Journal of Accounting and Economics*, 1987: 61-87.

Churbanov, Alexander and Winters-Hilt, Stephen. "Implementing EM and Viterbi algorithms for Hidden Markov Model in linear memory." *BMC bioinformatics*, 2008: 1-15.

Diaconis, Persi. "The markov chain monte carlo revolution." *Bulletin of the American Mathematical Society*, 2009: 179-205.

Hassan, Md Rafiul and Nath, Baikunth. "Stock market forecasting using hidden Markov model: a new approach." In *5th International Conference on Intelligent Systems Design and Applications (ISDA'05)*, by Md Rafiul and Nath, Baikunth Hassan, 192-196. NW Washington: IEEE, 2005.

Khiatani, Diksha and Ghose, Udayan. *Weather forecasting using hidden Markov model.* Gurgaon: IEEE, 2018.

Mor, Bhavya and Garhwal, Sunita and Kumar, Ajay. "A systematic review of hidden markov models and their applications." *Archives of computational methods in engineering*, 2021: 1429-1448.

Nguyen, Nguyet. "Hidden Markov model for stock trading." *International Journal of Financial Studies*, 2018: 36.

Somani, Poonam and Talele, Shreyas and Sawant, Suraj. "Stock market prediction using hidden Markov model." In *2014 IEEE 7th Joint International Information Technology and Artificial Intelligence Conference*, by Poonam and Talele, Shreyas and Sawant, Suraj Somani, 89-92. Chongqing: IEEE, 2014.

Yu, Shun-Zheng and Kobayashi, Hisashi. "An efficient forward-backward algorithm for an explicit-duration hidden Markov model." *IEEE signal processing letters*, 2003: 11-14.