Libya Country

Libyan Academy of Graduate Studies – Misurata Branch

Department of Computer Science

Foreign Currency Exchange Rates Forecasting Based on Hidden Markov Model

By

Afrah Ramadan Shaaib

B.Sc.in Computer Science, 7th October University, 2006

Supervised by

Dr. Idris El-Feghi

Thesis Submitted in Partial Fulfillment of the Requirement for the Degree of Master in Computer Science

Autumn 2013

Dedication

This thesis is dedicated to my parents for not telling but showing me what is important in life, and being always there for me. You have made all of this possible.

Thank you.

Acknowledgments

First of all, Praise be to **Allah** and all thanking to **Allah** for his help in this work and for all his gifts and his graces in all my life.

I would like to express my sincere regard and thanks to my supervisor ***Dr. Idris El-Feghi***, for his continuous support constant guidance and valuable comments. Without which, this thesis would not have been successfully completed.

My gratitude also goes out to all my teachers for their support and assistance, and for all utilizations provided by the Libyan Academy - Misurata.

In addition, wish to thank the members of my family, especially my mom and dad for always being there for me and act as my source of encouragement and inspiration.

My thanks also go to all the staff members and friends here at the Libyan Academy as a whole, whose company and friendship I dearly cherish. Sincere thanks to you all for making my stay here at the Academy memorable one.

Table of Contents

List of Tables

List of Figures

List of Terms

AI	Artificial Intelligent
AR	AutoRegressive
ANN	Artificial Neural Networks
ARMA	AutoRegressive Moving Average
EM	Expectation Maximization
FL	Fuzzy Logic
Forex, FX	Foreign Exchange
GA	Genetic Algorithm
HMM	Hidden Markov Model
MA	Moving Average
MLP	Multi-Layer Perceptron
PC	Principal Component
PCA	Principal Component Analysis
PCV	Principal Component Variable
SVM	Support Vector Machine

Nomenclature

$S = \{s_1, s_2, \dots, s_N\}$	a set of N hidden states,
$Q = \{q_1, q_2, \dots, q_T\}$	a state sequence of length T taking values from S,
$O = \{o_1, o_2, \dots, o_T\}$	a sequence consisting of T observations,
$A = \{a_{11}, a_{12}, \dots, a_{NN}\}$	the transition probability matrix A,
$B = b_i(o_t)$	a sequence of observation likelihoods,
$\Pi = \{\pi_1, \pi_2, \dots, \pi_N\}$	the initial probability distribution,
$\lambda = \{A, B, \Pi\}$	the complete parameter set of the HMM,
$\alpha_t(i)$	the joint probability of $\{o_1, o_2, \dots, o_t\}$ and $q_t = s_i$ given λ,
$\beta_t(i)$	the joint probability of $\{o_{t+1}, o_{t+2}, \dots, o_T\}$ and $q_t = s_i$ given λ,
γ_t	the probability of $q_t = s_i$ given λ,
$\xi_t(i,j)$	the joint probability of $q_t = s_i$ given O and λ,
$\delta_t(i)$	the highest joint probability of a state sequence ending in $q_t = s_i$ and a partial observation sequence ending in o_t given λ,
ψ_{tj}	the state s_i at time t which gives us $\delta_t(i)$, used for backtracking.

الملخص

التنبؤ بأسعار صرف العملات الأجنبية باستخدام نموذج ماركوف المخفي (HHM)

في هذه الأطروحة سيتم تقديم نموذج ماركوف المخفي للتنبؤ بأسعار صرف العملات الأجنبية. حيث إن نماذج ماركوف المخفية مستخدمة بكثرة حديثا لحل مشاكل تمييز الأنماط والتصنيف نظرا لكفاءتها في نمذجة الأنظمة الديناميكية. لكن هذه الطريقة ليست مباشرة ت

Abstract

Hidden Markov Models (HMMs) have been extensively used for pattern recognition and classification problems, due to its proven suitability for modeling dynamic systems. In this thesis, a HMM approach for predicting foreign currency exchange (Forex or FX) rates is presented.

Different macro-economic factors of FX market are used. Macro-economic factors include technical indicators. These technical indicators help to decide the patterns of the market at a particular time. There are hundreds of technical indicators are available, but all technical indicators are not useful. Therefore, we have obtained most effective technical indicators by applying Principal Component Analysis (PCA).

However, using HMM for predicting future events is not straightforward scheme. HMM uses a prediction financial time series that consists of open, high, low and close selected technical indicators as input variable. In this research, the HMM is trained on the past dataset of the chosen currencies (such as USD/EURO). The trained HMM is used to search for the variable of interest behavioral data pattern from the past dataset. By interpolating the neighboring values of these datasets, predictions are obtained. The obtained results was compared with real values from Forex database [1], and other results obtained using other techniques.

The power and predictive ability of the model are evaluated on the basis of Mean Square Error (MSE) with approximately 0.2 %. The Experimental results obtained using are encouraging, and HMM offers a new paradigm for currency market forecasting, an area that has been of much research interest lately.

Keywords: *Hidden Markov Model, Forex market forecasting, currency market, Foreign exchange market, FX market, Financial Time Series.*

Chapter1:

Introduction

Chapter 1: Introduction

1.1 Background:

Currency exchange is very attractive for both corporate and individual traders who make money on the Forex - a special financial market assigned for the foreign exchange. The currency trading (Forex) market is cash interbank market established 1971. It is the biggest and fastest growing market on earth. According to the Bank for International Settlements (BIS), as of April 2010, average daily turnover in global foreign exchange markets is estimated at 3.98 trillion dollars [2]. It allows businesses to convert one currency to another foreign currency. The currency market as a whole is referred to as Forex.

Forex has always been one of greatest challenges in business, finance and Artificial Intelligent (AI). It helps investors to hedge against potential market risks. Nevertheless, Forex forecasting is a complex and difficult task because of many factors that influence the market. There are two types of analysis namely fundamental and technical. (1) Fundamental analysis: which factors involved in price analysis, such as: supply and demand, seasonal cycles, weather, and government policy …etc., (2) Technical analysis: which charts are based on market action involved in price, volume, and history. The currency markets in the recent past years have become an integral part of the universal economy. Any fluctuation in this market influences personal and corporate financial lives, and the economic state of a country. The currency market has always been one of the most popular investments due to its high returns [3]. So an intelligent forecasting model would be desirable to develop. However, for better results, it is important to find the appropriate technique for a given forecasting task.

There are numerous forecasting models available. These models developed based on various techniques. Such as Statistical analysis like Autoregressive Moving Average (ARMA) model [4], Artificial Neural Networks (ANN) [5], Fuzzy Logic (FL) [6], Neuro-Fuzzy systems [7], Support Vector Machine (SVM) [8], evolutionary algorithms [9], Hidden Markov Models (HMM) [10]. Be among all the methods mentioned above HMM has received more attention due to its accurate prediction.

In this thesis, a HMM approach to forecast currency price for Forex market is presented. HMMs are statistical models of sequential data that have been successfully used in many machine learning applications, especially for speech recognition. The theory of HMMs was developed in the late 1960s and early 1970s. In recent years, HMMs have been applied to a variety of applications, such as handwriting recognition [11, 12], pattern recognition in molecular biology [13, 14], and fault-detection [15].

Furthermore, HMM is used in a method to develop forecasts. First, locating patterns from the past data sets that match with today's currency price behavior, then interpolate these two data sets with appropriate neighboring price elements and forecast tomorrow's currency price of the variable of interest. The technical indicators are used as input variable that gives more information as compared to FX prices.

1.2 Thesis Statement:

Evaluate the use of Hidden Markov Models as a powerful tool for foreign exchange data forecasting based on observed history.

1.3 Thesis Aim and Objectives:

The main aim of the research is to build a complete Forex currency pairs prices changes model based on the use of historical data with efficient artificial intelligence algorithms. Accordingly, the following objectives are achieved:

- Developing an efficient forecasting algorithm based on the use of Hidden Markov Model for forecasting Foreign Currency Exchange Market prices.
- Improving the accuracy of forecasting using HMM.
- Comparing the results obtained using HMM with real values from Forex Software.
- Comparing the results obtained using HMM with results obtained from other techniques.

1.4 Thesis Organization:

This thesis is split into seven chapters:

Chapter 1 is the introduction to this thesis and provides a summary of the background information to it. The problem description and the thesis organization are also provided here.

Chapter 2 examines current and past literature in the field of financial forecasting with the references to some of the popular models that have consistently provided a certain level of accuracy in this field.

Chapter 3 introduces the financial time series, focusing on Foreign Exchange Market and the basis of its structure, participants and recent trends.

Chapter 4 presents background and information on Markov chains and HMMs.

In **Chapter 5**, the developed model is described in detail to clear out how the HMM will use as a predictor.

In **Chapter 6**, an evaluation of the architecture and a selection of method for foreign exchange forecasting are presented.

Finally, **Chapter 7** states the conclusions drawn from this thesis and suggests possible directions for further research.

Chapter 2:

Literature Survey

Chapter 2: Literature Survey

2.1 Background:

Literature survey is the documentation of a comprehensive review of the published and unpublished work from secondary source data in the areas of specific interest to the researcher. It aims to review the critical points of current knowledge, including substantive findings as well as theoretical and methodological contributions to a particular topic. This chapter presents recent researches, which are related to financial forecasting.

2.2 Related Works:

There are many forecasting models available. These models developed based on different techniques. The following sections show some of these forecasting technique studies.

2.2.1 ARMA:

ARMA is probably the most known before, and often applied method for time series data' understanding and prediction. It consists of two parts, an Autoregressive (AR) part and a Moving Average (MA) part. First, the AR and MA parameters of an ARMA process given a time series must be estimated. After parameter estimation, it can be used predict the next sample value of the time series. The univariate output, describing the predicted next sample, depends on the present and the last inputs and outputs of the system. Before analyzing a time series with an ARMA model, it needs to be stripped from its trend and seasonal components, as ARMA only deals with stationary data.

Fung and Chung described the methodology for developing ARMA models to represent the workpiece roundness error in the machine taper turning process. The method employs a two stages approach in the determination of the AR and MA parameters of the ARMA model. It first calculates the parameters of the equivalent autoregressive model of the process, and then derives the AR and MA parameters of the ARMA model. Akaike's Information Criterion (AIC) is used to find the appropriate orders m and n of the AR and MA polynomials respectively (m, n) . Recursive algorithms are developed for the on-line implementation on a laboratory-turning machine. Evaluation of the effectiveness of using ARMA models in error forecasting is made using three time series obtained from the experimental machine. Analysis shows that ARMA(3,2) with forgetting factor of 0.95 gives acceptable results (nearly 17.61%) for this lathe turning machine [4].

2.2.2 ANN:

ANN is a mathematical model or computational model based on biological neural networks. It consists of an interconnected group of artificial neurons, and processes information using a connectionist approach to computation. An artificial neuron is a simple unit that computes a linear, weighted sum with an additional output function. Perhaps, the greatest advantage of ANNs is their ability to be used as an arbitrary function approximation mechanism that 'learns' from observed data. The most important neural network type is the Multi-Layer Perceptron (MLP) with strict feed-forward architecture of three layers. The connections between inputs and outputs are typically made via one or more hidden layers of neurons or nodes. The input layer is defined to assume the values of the input vector and does not perform any additional computation, the hidden layer of neurons or nodes is fully connected to the input and output layers and usually uses the sigmoid function as output function. Figure 2-1 shows a typical ANN with three inputs, and one hidden layer of two neurons and one output.

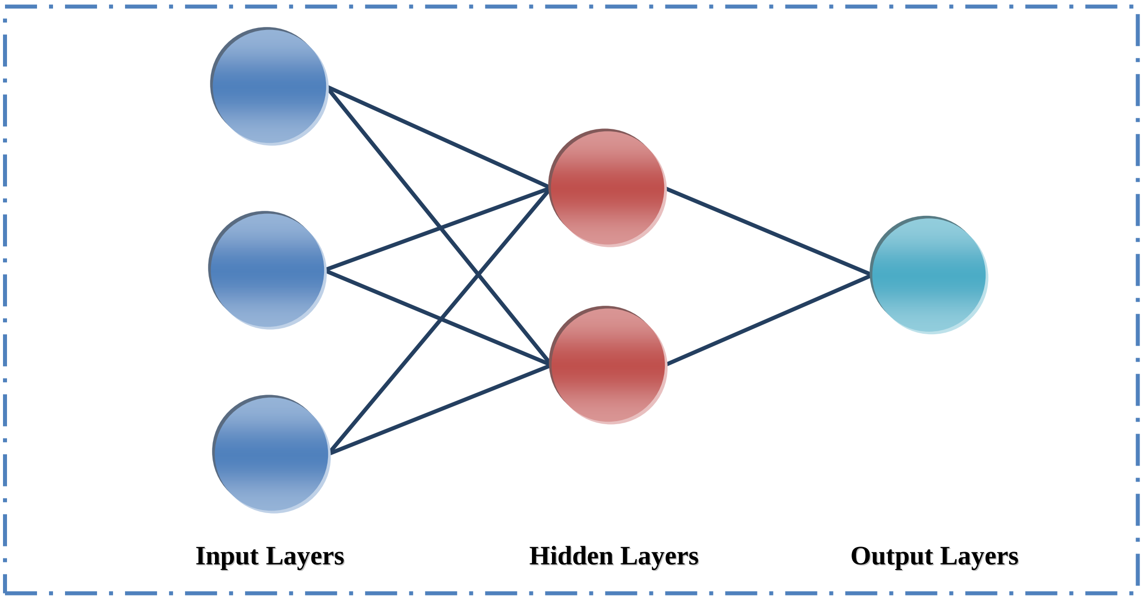


Figure 2-1: Architecture for a typical ANN with three inputs, one hidden layer of two neurons, and one output

Eng, Li, Wang and Lee report empirical evidence that an ANN is applicable to the prediction of foreign exchange rates. The effects of the choice of inputs into a neural network model are examined. Except for the normally used time series data and technical indicators, fundamental indicators such as interest rates and gross domestic products are fed into the neural networks to see if any relationship may be captured and improve the predictive capabilities of the model. Economic fundamentals are added to the ANN as inputs, results showed that the economic fundamentals are important in exchange rates' movements but their underlying relationships were not captured by the ANN. This was shown when they did not improve the networks predictive performance despite their use as inputs. This could be a result of the frequency of the economic fundamentals which are updated only quarterly. Perhaps, with economic indicators which are more frequently updated, there may be a greater contribution to the predictive performance of the ANN [5].

2.2.3 FL:

Fuzzy logic is a set of mathematical principles for knowledge representation based on degrees of membership. It is not logic that is fuzzy, but logic that is used to describe fuzziness. Fuzzy logic is the theory of fuzzy sets, sets that calibrate vagueness. It is based on the idea that all things admit of degrees. Unlike two-valued Boolean logic, fuzzy logic is multi-valued. It deals with degrees of membership and degrees of truth, and uses the continuum of logical values between 0 (completely false) and 1 (completely true). Temperature, height, speed, distance, beauty – all come on a sliding scale. For example, Figure 2-2 shows temperature measurements that have several separate membership functions defining particular temperature ranges. The meanings of the expressions *cold*, *warm*, and *hot* are represented by functions mapping a temperature scale. A point on that scale has three "truth values" - one for each of the three functions. The vertical line in the image represents a particular temperature that the three arrows (truth values) gauge. Since the red arrow points to zero, this temperature may be interpreted as "not hot". The yellow arrow (pointing at 0.2) may describe it as "slightly warm" and the blue arrow (pointing at 0.8) "fairly cold".

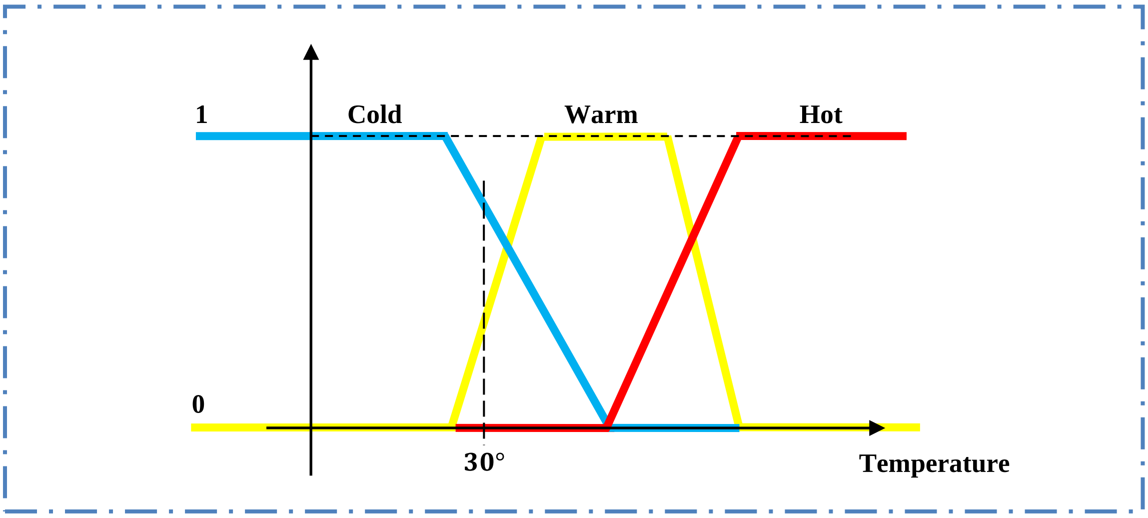


Figure 2-2: Fuzzy Logic Example

Lohani, Goel, and Bhatia presented a case study involving the application of fuzzy logic technique for real time flood forecasting. It is observed that the model performs well in calibration and validation as evaluated by model performance criteria this may be due to its capability to model nonlinearity present in the hydrological system. The modeling approach based on fuzzy logic exploits the available large historical data and avoids the very expensive work related to the development and the use of the complete hydrological model. Fuzzy logic has an advantage over many statistical and artificial neural network based methods in that the performance of a fuzzy model is not solely dependent on the volume of historical data available. The modeling performance can be further improved by real time updating and considering a suitable structure for modeling errors [6].

2.2.4 Neuro-Fuzzy Systems:

In the last few years, scientists and engineers have tried different methods to combine neural network and fuzzy systems together to take advantage of each one's strength. Neuro-Fuzzy system combines the neural network with fuzzy logic, it is a fuzzy logic system trained by a neural network, in which fuzzy rules replace the traditional crisp logic in the reasoning.

Abbasi and Abouec have managed to design a model to investigate the current trend of stock price of the "IRAN KHODRO Corporation" at Tehran Stock Exchange. For the Long term Period, a Neuro-Fuzzy with two Triangular membership functions and four independent Variables including trade volume, Dividend Per Share (DPS), Price to Earning Ratio (P/E), and also closing Price and Stock Price fluctuation as an dependent variable are selected as an optimal model. For the short-term Period, a Neuro-fuzzy model with two triangular membership functions for the first quarter of a year, two trapezoidal membership functions for the Second quarter of a year, two Gaussian combination membership functions for the third quarter of a year and two trapezoidal membership functions for the fourth quarter of a year were selected as an optimal model for the stock price forecasting. In addition, three independent variables including trade volume, price to earning ratio, closing Stock Price and a dependent variable of stock price fluctuation were selected as an optimal model. The findings of the research demonstrate that the trend of stock price could be forecasted with the lower level of error [7].

2.2.5 SVM:

Support Vector Machines are a newer technique for machine learning and are suitable for both pattern recognition and regression estimation, which can be applied to time series prediction. The basic concept of SVMs is that data that is non-separable in its original space can be mapped to another space where it is separable by a linear hyperplane. This hyperplane is chosen so that the distance between the classes that should be separated is maximized. In the final solution of the separation problem, the hyperplane is defined only by those data points that lie closest to it, i.e. those points which are nearest to the other class and are consequently called *support vectors* because changing them would also change the hyperplane and thus the solution of the training. If all other points from the data set were removed or moved within the same class, training would still yield the same hyperplane.

Kim examines the feasibility of applying SVM in financial forecasting by comparing it with back-propagation neural networks and case-based reasoning. The experimental results show that SVM provides a promising alternative to stock market prediction. This study used SVM to predict a future direction of stock price index. The experimental result showed that the prediction performances of SVMs are sensitive to the value of these parameters. Thus, it is important to find the optimal value of the parameters. In addition, this study compared SVM with Back Propagation Network (BPN) and Case-Based Reasoning (CBR). The experimental results showed that SVM outperformed BPN and CBR. The results may be attributable to the fact that SVM implements the structural risk minimization principle, and this leads to better generalization than conventional techniques. Finally, this study concluded that SVM provides a promising alternative for financial time-series forecasting [8].

2.2.6 Evolutionary algorithms:

A genetic Algorithm (GA) is a search technique used in computing to find exact or approximate solutions to optimization and search problems. GA is a particular class of evolutionary algorithms that use techniques inspired by evolutionary biology such as inheritance, mutation, selection, and crossover (also called recombination). Genetic algorithms are implemented as a computer simulation in which a population of abstract representations (called chromosomes, the genotype, or the genome) of candidate solutions (called individuals, creatures, or phenotypes) to an optimization problem evolves towards better solutions. Traditionally, solutions are represented in binary as strings of 0s and 1s, but other encodings are also possible. The evolution usually starts from a population of randomly generated individuals, and happens in generations. In each generation, the fitness of every individual in the population is evaluated, multiple individuals are stochastically selected from the current population (based on their fitness), and modified (recombined, and possibly randomly mutated) to form a new population. The new population is then used in the next iteration of the algorithm. Commonly, the algorithm terminates when either a maximum number of generations has been produced, or a satisfactory fitness level has been reached for the population. If the algorithm has terminated due to a maximum number of generations, a satisfactory solution may or may not have been reached.

Slany described a self-adapting architecture for Forex market prediction, which is being developed. The proposed system utilizes genetic programming (GP) for predictor representation. The goal of the system is the design and adaptation of simple predictors, which can either be used by the system itself or be ’manually’ used by a human trader. The results show that the system design is capable, under certain circumstances, of a relatively good prediction rate. The major drawback is the high ratio of mispredicted turning points.

Chapter 3: Financial Time Series Forecasting

Chapter 3: Financial Time Series Forecasting

3.1 Time Series:

A time series is a set of observations measured sequentially through time. These measurements may be made continuously through time or be taken at a discrete set of time points. By convention, these two types of series are called continuous and discrete time series, respectively, even though the measured variable may be discrete or continuous in either case. In other words, for discrete time series, for example, it is the time axis that is discrete. For a continuous time series, the observed variable is typically a continuous variable recorded continuously on a trace, such as a measure of brain activity recorded from an EEG (electroencephalography) machine.

The usual method of analyzing a continuous series is to sample (or digitize) the series at equal intervals of time to give a discrete time series. This process loses little or no information if the sampling interval is small enough.

On the other side, most of econometric models are constructed based on the assumption, time series is a stationary process. A stationary process is a stochastic process whose joint probability distribution does not change when shifted in time or space. It means the parameters such as the mean and variance, if they exist, do not change over time or position. In fact, many time series are non-stationary. If we process the non-stationary process as a stationary process, it will cause the temporary fluctuation has a long-term influence.

3.2 Forecasting:

Forecasting is an important activity in economics, commerce, marketing and various branches of science. Forecasting is to make statements about the likely course of future events. In technical terms, conditional on what one knows, what can one say about the future?

Good forecasts are vital in many areas of scientific, industrial, commercial and economic activity. This thesis is concerned with financial time-series forecasting, where forecasts are made based on data comprising one or more time series. A time-series is a collection of observations made sequentially through time.

A forecasting method is a procedure for computing forecasts from present and past values. As such, it may simply be an algorithmic rule and need not depend on an underlying probability model. Alternatively, it may arise from identifying a particular model for the given data and finding optimal forecasts conditional on that model. Thus the two terms 'method' and 'model' should be kept clearly distinct. It is unfortunate that the term 'forecasting model' is used rather loosely in the literature and is sometimes wrongly used to describe a forecasting method.

Forecasting methods may be broadly classified into three types:

- **Judgemental forecasts:** based on subjective judgement, intuition, 'inside' commercial knowledge, and any other relevant information.
- **Univariate methods:** where forecasts depend only on present and past values of the single series being forecasted, possibly augmented by a function of time such as a linear trend.

- **Multivariate methods:** where forecasts of a given variable depend, at least partly, on values of one or more additional time series variables, called predictor or explanatory variables. Multivariate forecasts may depend on a multivariate model involving more than one equation if the variables are jointly dependent.

3.2.1 Time Series Forecasting:

To estimate the future values of the series, most authors use the terms 'forecasting' and 'prediction' interchangeably and this convention followed here. There is a clear distinction between steady-state forecasting, where the future expected to be much like the past, and What-if forecasting where a multivariate model is used to explore the effect of changing policy variables.

3.2.2 The Time Plot:

The first step in any time-series analysis or forecasting exercise is to plot the observations against time, to give what is called a time plot of the data. The graph should show up important features of the data such as trend, seasonality, outliers, smooth changes in structure, turning points and/or sudden discontinuities, and is vital, both in describing the data, in helping to formulate a sensible model and in choosing an appropriate forecasting method.

3.3 Forex Market:

The currency trading (foreign exchange, Forex or FX) market is cash interbank market established 1971. It is the biggest and fastest growing market on earth. According to the Bank for International Settlements (BIS), as of April 2010, average daily turnover in global foreign exchange markets is estimated at 3.98 trillion dollars [16]. It is where banks and other official institutions facilitate the buying and selling of foreign currencies. In the other word, the Forex objective is to facilitate the world trade and investment. The investor's goal in Forex trading is to profit from foreign currency movements.

3.3.1 Exchange Rate:

Exchange rate is the price of one country's currency expressed in another country's currency. In other words, the rate at which one currency can be exchanged for another. e.g. Rs. 51.42 per one USD. The major currencies of the World are:

- U.S. Dollar - (USD, $),
- European Currency Unit - (EUR, €),
- Japanese Yen - (JPY, ¥),
- British Pound Sterling - (GBP, £),
- Swiss Franc - (CHF),
- Canadian Dollar - (CAD, $),
- Australian Dollar - (AUD, $).

Currencies are traded against one another. Each pair of currencies thus constitutes an individual product and is traditionally noted XXX/YYY, where YYY is the ISO 4217 international three-letter code of the currency into which the price of one unit of XXX currency is expressed. First listed currency is the 'BASE' currency - basis for buying or selling transaction. Second listed currency is called the 'Counter' or 'Quote' currency.

For instance, EUR/USD is the price of the euro expressed in US dollars, as in 1 euro = 1.2045 dollar. If a trader places a buy EUR/USD order, the action that takes place is the trader sells the USD and buys EUR. The other example of listed currency pairs are:

- EUR/USD,
- USD/JPY,
- USD/CHF,
- GBP/USD.

Forecasting currency markets has always been one of greatest challenges in business, finance and AI. It helps investors to hedge against potential market risks. Nevertheless, Forex forecasting is a complex and difficult task because of many factors that influence the market trade, such as: political factors, economic factors, and market psychology.

3.3.2 Forex Structure:

The markets core is built up by a number of different banks. That is why it sometimes is called an inter-bank market. The market is opened 24 hours a day and moves according to activity in large exporting and importing countries as well as in countries with highly developed financial sectors [16].

The Forex is so large and is composed of so many participants, that is no one player, even the government central banks, can control the market. The participants in Forex market are:

- **Central banks:** Central banks play an important role in the Forex market. They try to maintain the money supply, interest rates, inflation, and other market factors. Participants in the market all tend to respect the opinions of the central banks because of the power and control they have over the value of their national currency.

- **Commercial banks:** Both small and large banks, working for themselves and their clients (institutions, individual investors), participate in the Forex markets. Commercial banks act as Market Makers, Ready to Buy as well as Sell any time.
- **Corporations:** Small and large companies also play an important role in the Forex market. These companies often use foreign exchange to pay for goods or services, and keep the market strong through international trade and foreign currency exchange between multinational companies.
- Institutional investors,
- Private individuals [17].

There are two aspects of trading namely fundamental and technical:

- **Fundamental analysis:** It focuses on what ought to happen in a market, which factors involved in price analysis, such as: supply and demand, seasonal cycles, weather, and government policy …etc.
- **Technical analysis:** It focuses on what actually happens in a market, which charts are based on market action involved in price, volume, and history.

3.3.3 Forex Candlesticks:

Over the last few decades, traders have begun to use candlestick charts far more frequently than any other technical analysis tool. Candlestick charts have a simple, easy to analyze appearance. Unlike bar or line charts, candlestick charts provide more detailed information about the market at a glance. Most traders prefer to use the candlestick chart because it can help them to:

- Determine the current state of the market at a glance.
- See the direction of the market more easily.
- Identify market patterns quickly.

Figure 3-1 shows a candlestick chart for GBP/USD currency pair. Each candlestick on the candlestick chart shows the range of a currency in a vertical line and is defined by four price points (high, low, open and close) as shown on figure 3-2.

Figure 3-1: Candlestick chart for GBP/USD currency pair.

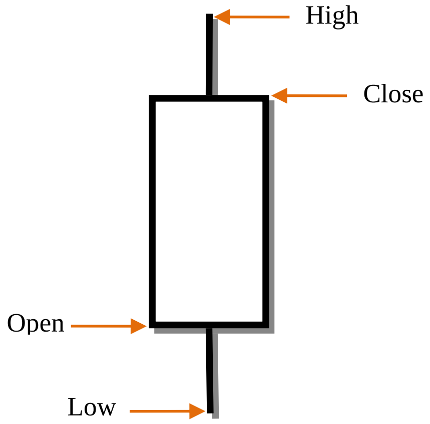


Figure 3-2: Candlestick price points

Each candlestick is made up of two parts, a body and shadows. Candlestick body is a rectangle that represents the level of trading activity for a specified period. Candlestick shadows (also called tails or wicks) are the thin lines above and below the body. An upper shadow displays how high trading went while a lower shadow shows how low it went. Figure 3-3 shows candlestick parts.

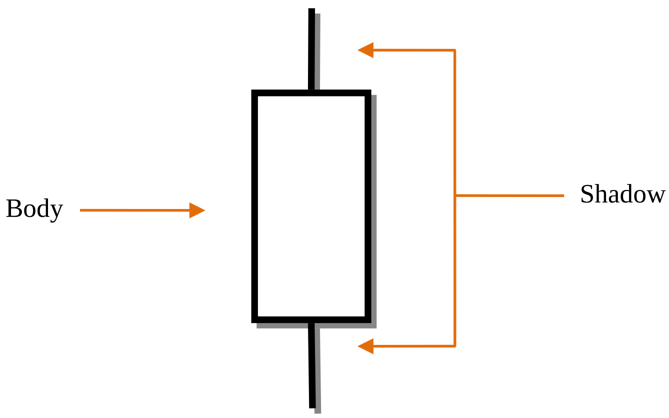


Figure 3-3: Candlestick Parts

The appearance of the candlestick body and its shadows provide information about the state of the market and where it is moving. The length of the candlestick body shows where the majority of the trading took place (figure 3-4). A long body suggests that the market is trading heavily in one direction, while a small body indicates lighter trading.

The appearance of shadows can also tell which way the market is heading. Long shadows show that trading went far past the open and close values while short shadows indicate that most of the trading happened near the open and close. Typically, short shadows mean that there is little change in the market direction, but a long shadow can signify a big change.

Figure 3-4: Examples of long and short bodies and shadows

As shown in figure 3-5 the green candlesticks appear in an “up” candle; in other words, the currency closed higher than the previous candle’s close. Red candlesticks show a “down” candle or that the currency closed below the previous candle’s open. Traditionally, up trends were represented by white candlesticks, while down trends were depicted by black candlesticks.

Figure 3-5: Candlesticks of up trends are colored green and those of down trends are red.

Chapter 4: Hidden Markov Model

Chapter 4: Hidden Markov Model

4.1 Introduction:

Although initially introduced in the 1960's, HMM first gained popularity in the late 1980's. There are mainly two reasons for this; first, the models are very rich in mathematical structure and hence can form the theoretical basis for many applications. Second, the models, when applied properly, work very well in practice [18].

HMMs are one of the most fundamental and widely used statistical tools for modeling discrete time series, with widely diverse applications including automatic speech recognition, Natural Language Processing (NLP), and genomic sequence modeling.

Many practical problems have been found not to be solvable by simple Markov chain models, where the observations are mapped to states. HMMs thus add an additional hidden layer, hence the name, which contains a Markov chain model that indirectly influences the HMM output via a separate probability distribution. To detail, a complete HMM consists of a set of states, a state transition probability distribution, an observation symbol probability distribution and an initial state distribution. When training a HMM model from a data set, the number of states and the connections between the states need to be determined a priori and all three probability distributions are adjusted by training on the data.

A HMM is a statistical model in which the system being modeled is assumed to be a Markov process with unknown parameters, and the challenge is to determine the hidden parameters from the observable parameters. The extracted model parameters can then be used to perform further analysis, for example for pattern recognition applications.

4.2 Bayes Theorem:

A HMM can be considered as the simplest dynamic Bayesian network, which is a probabilistic graphical model that represents a set of variables and their probabilistic independencies, and where the variables appear in a sequence.

Bayesian probability is an interpretation of the probability calculus which holds that the concept of probability can be defined as the degree to which a person (or community) believes that a proposition is true. Bayesian theory also suggests that Bayes theorem can be used as a rule to infer or update the degree of belief in light of new information.

The probability of an event A conditional on event B is generally different from the probability of B conditional on A. However, there is a definite relationship between the two, and Bayes' theorem is the statement of that relationship. To derive the theorem, the definition of conditional probability used. The probability of event A given B is

$$P(A|B) = \frac{P(A \cap B)}{P(B)}.$$

Likewise, the probability of event *B*, given event *A* is

$$P(B|A) = \frac{P(B \cap A)}{P(A)}.$$

Rearranging and combining these two equations, one find

$$P(A|B)P(B) = P(A \cap B) = P(B \cap A) = P(B|A)P(A).$$

This sometimes called the product rule for probabilities. Dividing both sides by $P(B)$, given $P(B) \neq 0$, Bayes' theorem is obtained:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}.$$

4.3 Markov Chains:

A Markov chain, sometimes also referred to as an observed Markov Model, can be seen as a weighted finite-state automaton, which is defined by a set of states and a set of transitions between the states based on the observed input. In the case of the Markov chain the weights on the arcs going between the different states can be seen as probabilities of how likely it is that a particular path is chosen. The probabilities on the arcs leaving a node (state) must all sum up to 1. In figure 4-1 there is a simple example of a Markov chain explaining the weather. a_{ij} found on the arcs going between the nodes, represents the probability of going from state s_i to state s_j.

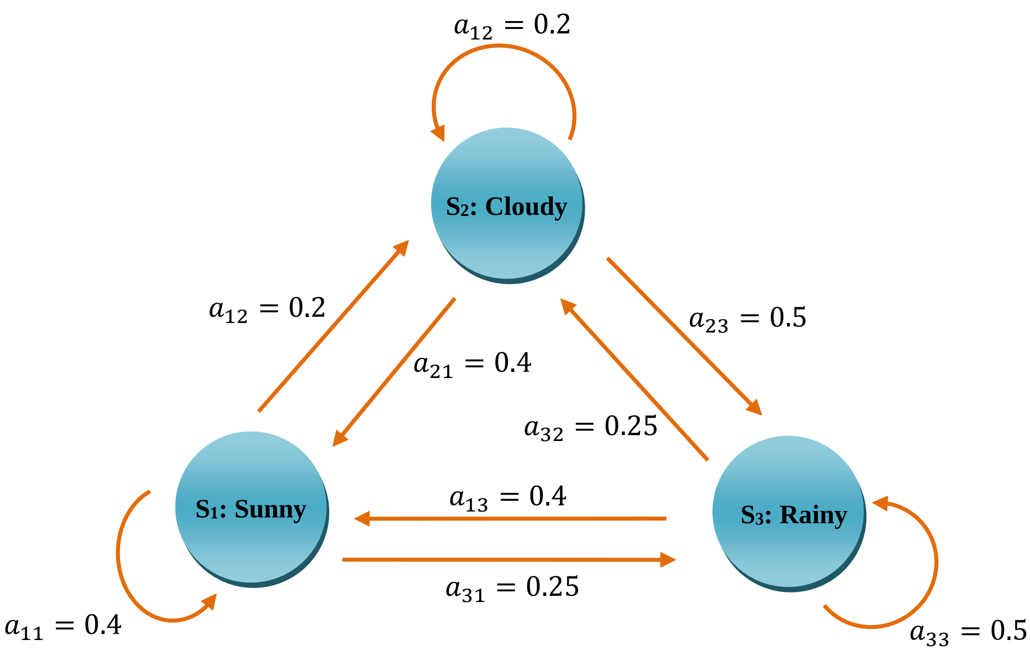


Figure 4-1: A simple example of a Markov chain explaining the weather

In the previous figure, a simple model of the weather is set up, specified by the three states; sunny, cloudy and rainy. Given that the weather is rainy (state 3) on day 1 $(t = 1)$, what is the probability that the three following days will be sunny? Stated more formal, one have an observation sequence $O = \{s_3, s_1, s_1, s_1\}$ for $t = 1, 2, 3, 4$, and wish to determine the probability of O given the model in figure 4-1. The probability is given by:

$$\begin{aligned} P(O|Model) &= P(s_3, s_1, s_1, s_1|Model) \\ &= P(s_3) \cdot P(s_1|s_3) \cdot P(s_1|s_1) \cdot P(s_1|s_1) \\ &= 1 \cdot 0.25 \cdot 0.4 \cdot 0.4 = 0.04 \end{aligned}$$

For a more formal description, the Markov chain is specified by:

$S = \{s_1, s_2, \dots, s_N\}$	a set of N states,
$A = \{a_{11}, a_{12}, \dots, a_{NN}\}$	a transition probability matrix A, where each a_{ij} represents the probability of moving from state i to state j, with $\sum_{j=1}^{N} a_{ij} = 1, \forall i$,
$\Pi = \{\pi_1, \pi_2, \dots, \pi_N\}$	an initial probability distribution, where π_i indicates the probability of starting in state i. Also, $\sum_{i=1}^{N} \pi_i = 1$.

Instead of specifying Π, one could use a special start node not associated with the observations, and with outgoing probabilities $a_{start,i} = \pi_i$ as the probabilities of going from the $start$ state to state i, and $a_{start,start} = a_{i,start} = 0$, $\forall i$. The time instants associated with state changes are defined as as $t = 1, 2, \dots$ and the actual state at time t as q_t.

An important feature of the Markov chain is its assumptions about the probabilities. In a first-order Markov chain, the probability of a state only depends on the previous state, that is:

$$P(q_t|q_{t-1}, \dots, q_1) = P(q_t|q_{t-1})$$

Markov chains, where the probability of moving between any two states are nonzero, are called fully-connected or ergodic. But this is not always the case; in for example a left-right (also called Bakis) Markov model there are no transitions going from a higher-numbered to a lower-numbered state. The way the trellis is set up depends on the given situation.

4.4 HMM:

A Markov chain is useful when one want to compute the probability of a particular sequence of events, all observable in the world. However, the events that are of interest might not be directly observable, and this is where HMM comes in handy. HMM is a generalization of a Markov chain, in which each state is not directly observable but variables influenced by the state are visible, also called "emission", according to given stationary probability law. In this case, time evolution of the internal states can be induced only through the sequence of observed output states.

HMM is characterized by the following elements:

$S = \{s_1, s_2, \dots, s_N\}$ a set of N hidden states. In this model, the states are defined by price buckets,

$Q = \{q_1, q_2, \dots, q_T\}$ a state sequence of length T taking values from S,

$O = \{o_1, o_2, \dots, o_T\}$ an observation sequence consisting of T observations,

	taking values from the discrete alphabet $V = \{\mathrm{v}_1, \mathrm{v}_2, \dots, q_{\mathrm{M}}\}$. This can be thought of as the number of observations that fall in each price bucket.
$A = \{\mathrm{a}_{11}, \mathrm{a}_{12}, \dots, \mathrm{a}_{\mathrm{NN}}\}$	a transition probability matrix A, where each a_{ij} represents the probability of moving from state s_i to state s_j, with $\sum_{j=1}^{N} a_{ij}, \forall i$,
$B = b_i(o_t)$	a sequence of observation likelihoods, also called emission probabilities, expressing the probability of an observation o_t being generated from a state s_i at time t,
$\Pi = \{\pi_1, \pi_2, \dots, \pi_{\mathrm{T}}\}$	an initial probability distribution, where π_i indicates the probability of starting in state s_i. Also, $\sum_{j=1}^{N} \pi_i = 1$.

As before, at time instants associated with state changes are defined as $t = 1, 2, \dots$ and the actual state at time t as q_t. The notation $\lambda = \{\mathrm{A}, \mathrm{B}, \Pi\}$ is also introduced used as compact notation to HMM, which indicates the complete parameter set of the model.

A first-order HMM makes two assumptions; first, the probability of a state is only dependent on the previous state:

$$\mathrm{P}(\mathrm{q}_t | \mathrm{q}_{t-1}, \dots, \mathrm{q}_1) = \mathrm{P}(\mathrm{q}_t | \mathrm{q}_{t-1})$$

Second, the probability of an output observation o_t is only dependent on the state that produced the observation, q_t, and not on any other observations or states:

$$\mathrm{P}(\mathrm{o}_t | \mathrm{q}_t, \mathrm{q}_{t-1}, \dots, \mathrm{q}_1, \mathrm{o}_{t-1}, \dots, \mathrm{o}_1) = \mathrm{P}(\mathrm{o}_t | \mathrm{q}_t)$$

For example, imagine that a climatologist in the year 2799 want to study how the weather was in 2007 in a certain region. Unfortunately he do not have any records for this particular region and time, but what he do have is the diary of a young man for 2007, that tells him how many ice creams he "young man" had every day. Stated more formal, he have an observation sequence, $O = \{o_1, \ldots, o_{365}\}$, where each observation assumes a value from the discrete alphabet $V = \{1, 2, 3\}$, i.e. the number of ice creams he had every day. Your task is to find the "correct" hidden state sequence, $Q = \{q_1, \ldots, q_{365}\}$, with the possible states $sunny$ (s_1) and $rainy$ (s_2), that corresponds to the given observations. Say for example that you know that he had one ice cream at time $t-1$, three ice creams at time t and two ice creams at time $t+1$. The most probable hidden state sequence for these three days might for example be $q_{t-1} = s_2$, $q_t = s_1$ and $q_{t+1} = s_2$, given the number of eaten ice creams. In other words, the most likely weather during these days would be $rainy$, $sunny$ and $rainy$. An example of how this could look is presented in figure 4-2, where the most probable state path (s_2, s_1, s_2) is outlined. a_{ij} states the probability of going from state s_i to state s_j, and $b_j(o_t)$ the probability of a specific observation at time t, given state s_j [19].

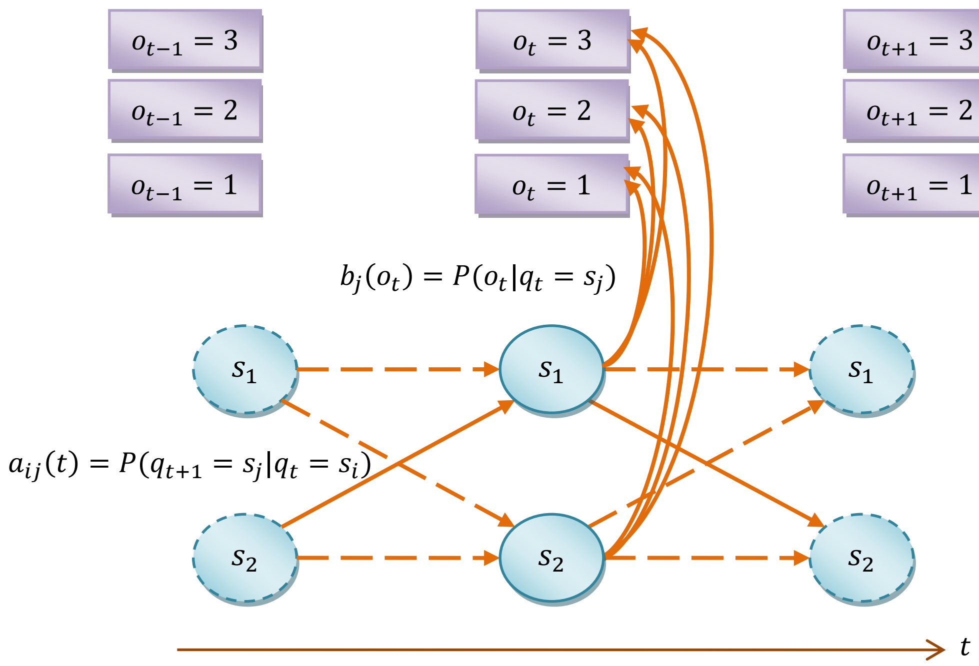


Figure 4-2: A HMM example [19]

4.5 HMM Fundamental Problems:

HMM should be characterized by three fundamental problems [18]:

1. **Computing likelihood:** Given the complete parameter set λ and an observation sequence O, determine the likelihood $P(O/\lambda)$. Forward-backward algorithms can solve this problem.

2. **Decoding:** Given the complete parameter set λ and an observation sequence O, determine the best hidden sequence Q. Viterbi algorithms can solve this problem.

3. **Learning:** Given an observation sequence O and the set of states in the HMM, learn the HMM λ. Baum-Welch algorithms can solve this problem.

In the world of finance, considered Q as an observed financial time series, X as some underlying economic forces, then the second problem is to find the hidden path of the economic forces, the third problem is to estimate the model parameters by using the observed information, the first problem is to predict the future by employing the model with the historical observation.

4.5.1 Computing Likelihood:

This is an evaluation problem, which means that given a model and a sequence of observations, what is the probability that the observations was generated by the model. This information can be very valuable when choosing between different models wanting to know which one that best matches the observations.

To find a solution to problem 1, one wish to calculate the probability of a given observation sequence, $O = \{o_1, o_2, \dots, o_T\}$, given the model $\lambda = \{A, B, \Pi\}$. In other words one want to find $P(O/\lambda)$. The most intuitive way of doing this is to enumerate every possible state sequence of length T. One such state sequence is

$$Q = \{q_1, q_2, \dots, q_T\} \tag{4-1}$$

where q_1 is the initial state. The probability of observation sequence O given a state sequence such as 4-1 can be calculated as

$$P(O|Q, \lambda) = \prod_{t=1}^{T} P(o_t|q_t, \lambda) \tag{4-2}$$

where the different observations are assumed to be independent. The property of independence makes it possible to calculate equation 4-2 as

$$P(O|Q, \lambda) = b_{q_1}(o_1) \cdot b_{q_2}(o_2) \cdot \ \dots \cdot b_{q_T}(o_T) \tag{4-3}$$

The probability of such a sequence can be written as

$$P(Q|\lambda) = \pi_{q_1} \cdot a_{q_1 q_2} a_{q_2 q_3} \cdot \ \dots \cdot a_{q_{T-1} q_T} \tag{4-4}$$

The joint probability of O and Q, the probability that O an Q occurs simultaneously, is simply the product of 4-3 and 4-4 as

$$P(O,Q|\lambda) = P(O|Q,\lambda)P(Q|\lambda). \tag{4-5}$$

Finally, the probability of O given the model λ is calculated by summing the right hand side of equation 4-5 over all possible state sequences Q

$$P(O|\lambda) = \sum_{q_1,q_2,\dots,q_T} P(O|Q,\lambda)P(Q|\lambda)$$

$$= \sum_{q_1,q_2,\dots,q_T} \pi_{q_1} b_{q_1}(o_1) a_{q_1 q_2} b_{q_2}(o_2) \dots\ a_{q_{T-1} q_T} b_{q_T}(o_T)$$

The last equation says that one at the initial time $t = 1$ are in state q_1 with probability π_{q_1}, and generate the observation o_1 with probability $b_{q_1}(o_1)$. As time ticks from t to $t+1$ ($t = 2$) one transform from state q_1 to q_2 with probability $a_{q_1 q_2}$, and generate observation o_2 with probability $b_{q_2}(o_2)$ and so on until $t = T$.

This procedure involves a total of $2TN^T$ calculations, which makes it unfeasible, even for small values of N and T. As an example it takes $2 \cdot 100 \cdot 5^{100} \approx 10^{72}$ calculations for a model with 5 states and 100 observations. Therefore it is needed to find a more efficient way of calculating $P(O|\lambda)$. Such a procedure exists and is called the Forward-Backward Procedure (The backward part of the calculation is not needed to solve Problem 1. It will be introduced when solving Problem 3). For initiation one need to define the forward variable as

$$\alpha_t(i) = P(o_1, o_2, \dots, o_t, q_t = s_i|\lambda).$$

In other words, the probability of the partial observation sequence, $o_1, o_2, \dots, o_t$ until time t and given state s_i at time t. One can solve for $\alpha_t(i)$ inductively as follows:

1. **Initialization:** $\alpha_1(i) = \pi_i b_i(o_1), \quad 1 \le i \le N.$ (4-6)
2. **Induction:** $\alpha_{t+1} = \left[\sum_{j=1}^{N} \alpha_t(i) a_{ij}\right] b_j(o_{t+1}), \quad 1 \le i \le T+1$

$1 \le j \le N$ (4-7)

3. Termination: $P(O|\lambda) = \sum_{i=1}^{N} \alpha_T(i)$. (4-8)

Step 1 sets the forward probability to the joint probability of state s_j and initial observation o_1. The second step, which is the heart of the forward calculation is illustrated in figure 4-3.

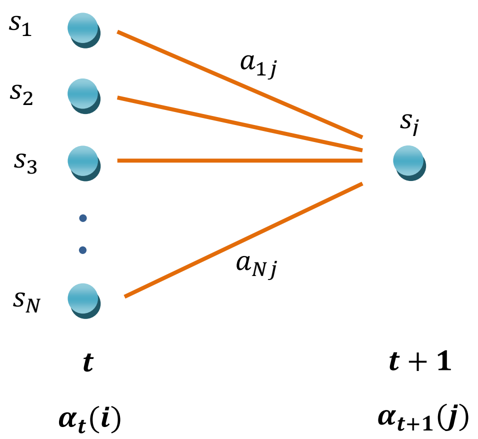


Figure 4-3: Illustration of the sequence of operations required for the computation of the forward variable $\alpha_t(i)$

One can see that state s_j at time $t+1$ can be reached from N different states at time t. By summing the product over all possible states s_i, $1 \le i \le N$ at time t results in the probability of s_j at time $t+1$ with all previous observations in consideration. Once it is calculated for s_j, it is easy to see that $\alpha_{t+1}(j)$ is obtained by accounting for observation o_{t+1} in state s_j, in other words by multiplying the summed value by the probability $b_j(o_{t+1})$. The computation of 2-7 is performed for all states s_j, $1 \le j \le N$, for a given time t and iterated for all $t = 1, 2, \dots, T-1$. Step 3 then gives $P(O|\lambda)$ by summing the terminal forward variables $\alpha_T(i)$. This is the case because, by definition

$$\alpha_T(i) = P(o_1, o_2, \dots, o_T, q_T = s_i|\lambda)$$

and therefore $P(O|\lambda)$ is just the sum of the $\alpha_T(i)$'s. This method just needs N^2T which is much more efficient than the more traditional method. Instead of 10^{72} calculations, a total of 2500 is enough, a saving of about 69 orders of magnitude. In the next two sections - one can see that this decrease of magnitude is essential because $P(O|\lambda)$ serves as the denominator when estimating the central variables when solving the last two problems.

In a similar manner, one can consider a backward variable $\beta_t(i)$ defined as follows:

$$\beta_t(i) = P(o_{t+1}, o_{t+2}, \dots, o_T | q_t = s_i, \lambda)$$

$\beta_t(i)$ is the probability of the partial observation sequence from $t + 1$ to the last time, T, given the state s_i at time t and the HMM λ. By using induction, $\beta_t(i)$ is found as follows:

1. **Initialization:** $\beta_T(i) = 1, \quad 1 \le i \le N.$
2. **Induction:** $\beta_t = \sum_{j=1}^{N} a_{ij} b_j(o_t)\, \beta_{t+1}(j), \quad t = T - 1, T - 2, \dots, 1$

   $1 \le i \le N$

Step 1 defines $\beta_T(i)$ to be one for all s_i. Step 2, which is illustrated in figure 4-4, shows that in order to have been in state s_i at time t, and to account for the observation sequence from time $t + 1$ and on, one have to consider all possible states s_j at time $t + 1$, accounting for the transition from s_i to s_j as well as the observation o_{t+1} in state s_j, and then account for the remaining partial observation sequence from state s_j.

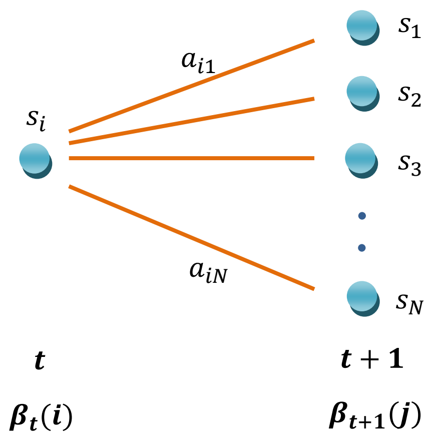


Figure 4-4: Illustration of the sequence of operations required for the computation of the backward variable $\beta_t(i)$

As mentioned before the backward variable is not used to find the probability $P(O|\lambda)$. Later, it will be shown how the backward as well as the forward calculation are used extensively to help one solve the second as well as the third fundamental problem of HMMs.

4.5.2 Decoding:

In this, the second problem one try to find the “correct” hidden path, i.e. trying to uncover the hidden path. This is often used when one wants to learn about the structure of the model or to get optimal state sequences. There are several ways of finding the “optimal” state sequence according to a given observation sequence. The difficulty lies in the definition of an optimal state sequence. One possible way is to find the states q_t which are individually most likely. This criteria maximizes the total number of correct states. To be able to implement this as a solution to the second problem one start by defining the variable

$$\gamma_t(i) = P(q_t = s_i|O,\lambda) \tag{4-9}$$

which gives the probability of being in state s_i at time t given the observation sequence, O, and the model, λ. Equation 4-9 can be expressed simply using the forward and backward variables, $\alpha_t(i)$ and $\beta_t(i)$ as follows:

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{P(O|\lambda)} = \frac{\alpha_t(i)\beta_t(i)}{\sum_{i=1}^{N} \alpha_t(i)\beta_t(i)}$$

It is simple to see that $\gamma_t(i)$ is a true probability measure. This since $\alpha_t(i)$ accounts for the partial observation sequence $o_1, o_2, \dots, o_t$ and the state s_i at t, while $\beta_t(i)$ accounts for the remainder of the observation sequence $o_{t+1}, o_{t+2}, \dots, o_T$ given state s_i at time t. The normalization factor $P(O|\lambda) = \sum_{i=1}^{N} \alpha_t(i)\beta_t(i)$ makes $\gamma_t(i)$ a probability measure, which means that

$$\sum_{i=1}^{N} \gamma_t(i) = 1$$

One can now find the individually most likely state q_t at time t by using $\gamma_t(i)$ as follows:

$$q_t = \text{argmax}_{1 \le i \le N}[\gamma_t(i)], \qquad 1 \le t \le T \tag{4-10}$$

Although equation 4-10 maximizes the expected number of correct states there could be some problems with the resulting state sequence. For example, when the HMM has state transitions which has zero probability the optimal state sequence may, in fact, not even be a valid state sequence. This is due to the fact that the solution of 4-10 simply determines the most likely state at every instant, without regard to the probability of occurrence of sequences of states.

To solve this problem one could modify the optimality criterion. For example by solving for the state sequence that maximizes the number of correct pairs of states (q_t, q_{t+1}) or triples of states (q_t, q_{t+1}, q_{t+2}).

The most widely used criterion however, is to find the single best state sequence, in other words to maximize $P(Q|O,\lambda)$ which is equivalent to maximizing $P(Q,O|\lambda)$. To find the optimal state sequence one often uses a method, based on dynamic programming, called the Viterbi Algorithm.

To find the best state sequence $Q = \{q_1, q_2, \dots, q_T\}$ for a given observation sequence $O = \{o_1, o_2, \dots, o_T\}$, one need to define the quantity

$$\delta_t(i) = \max_{q_1, q_2, \dots, q_{t-1}} P[q_1, q_2, \dots, q_t = s_i, o_1, o_2, \dots, o_t|\lambda]$$

which means the highest probability along a single path, at time t, which accounts for the first t observations and ends in state s_i. By induction one have:

$$\delta_{t+1}(j) = \left[\max_i \delta_t(i) a_{ij}\right] b_j(o_{t+1}) \tag{4-11}$$

To be able to retrieve the state sequence, one need to keep track of the argument which maximized 4-11, for each t and j. This is done via the array $\psi_t(j)$. The complete procedure for finding the best state sequence can now be stated as follows:

4. **Initialization:** $\delta_1(i) = \pi_i b_i(o_1), \; 1 \le i \le N$

   $\psi_1 = 0$

5. **Recursion:** $\delta_t(j) = \max_{1 \le i \le N} [\delta_{t-1}(i) a_{ij}] b_j(o_t), \quad 2 \le t \le T$

   $1 \le j \le N$

   $\psi_t(j) = \mathrm{argmax}_{1 \le i \le N} [\delta_{t-1}(i) a_{ij}], \quad 2 \le t \le T$

   $1 \le j \le N$

6. **Termination:** $P^* = \max_{1 \le i \le N} [\delta_T(i)]$

   $P_T^* = \mathrm{argmax}_{1 \le i \le N} [\delta_T(i)]$

7. **Path (state sequence) backtracking:**

$$P_T^* = \psi_{t+1}(P_{t+1}^*), \qquad t = T-1, T-2, \dots, 1$$

4.5.3 Learning:

The third, last, and at the same time most challenging problem with HMMs is to determine a method to adjust the model parameters $\lambda = \{A, B, \Pi\}$ to maximize the probability of the observation sequence given the model. There is no known analytical solution to this problem, but one can however choose λ such that $P(O|\lambda)$ is locally maximized using an iterative process such as the Baum-Welch method, a type of Expectation Maximization (EM) algorithm. The Baum-Welch is however the most commonly used procedure, and will therefore be the one utilized in this the first investigation of HMM on Forex data. To be able to re-estimate the model parameters, using the Baum-Welch method, one should start with defining $\xi_t(i,j)$, the probability of being in state s_i at time t, and state s_j at time $t+1$, given the model and the observations sequence. In other words the variable can be defined as:

$$\xi_t(i,j) = P(q_t = s_i, q_{t+1} = s_j | O, \lambda)$$

The needed information for the variable $\xi_t(i,j)$ is shown in figure 4-5.

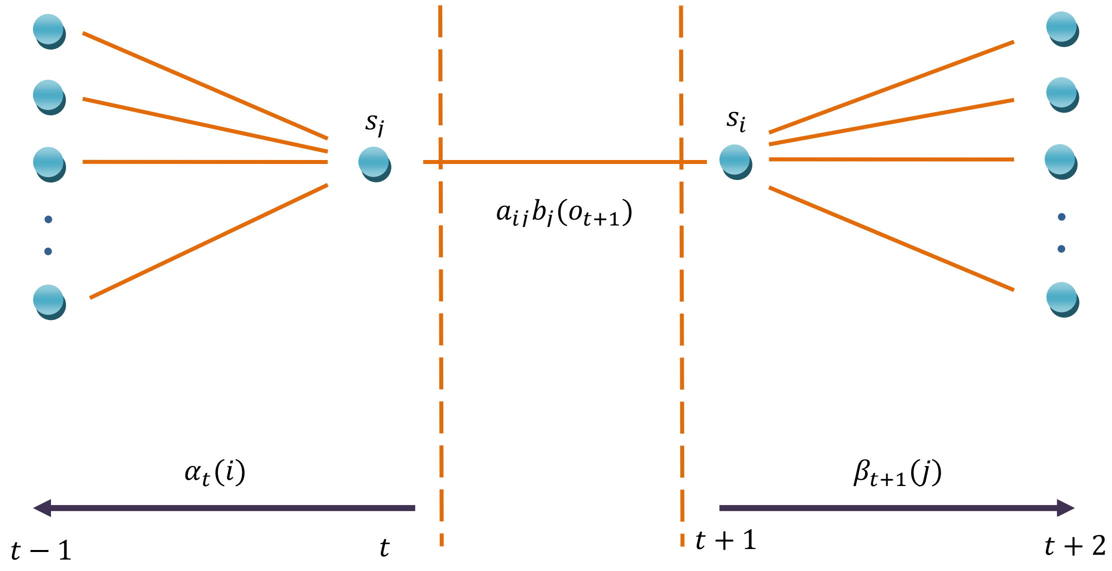


Figure 4-5: Illustration of the sequence of operations required for the computation of the joint event that the system is in state s_i at time t and state s_j at time $t+1$.

From this figure, one should be able to understand that $\xi_t(i,j)$ can be written using the forward and backward variables as follows:

$$\xi_t(i,j) = \frac{\alpha_t(i)a_{ij}b_j(o_{t+1})\beta_{t+1}(j)}{P(O|\lambda)} = \frac{\alpha_t(i)a_{ij}b_j(o_{t+1})\beta_{t+1}(j)}{\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_t(i)a_{ij}b_j(o_{t+1})\beta_{t+1}(j)}$$

which is a probability measure since the numerator is simply $P(q_t = s_i, q_{t+1} = s_j, O|\lambda)$ and denominator is $P(O|\lambda)$.

As described before, $\gamma_t(i)$ is the probability of being in state s_i at time t, given the observation sequence and the model. Therefore there is a close relationship between $\gamma_t(i)$ and $\xi_t(i,j)$. One can express $\gamma_t(i)$ as the sum of all $\xi_t(i,j)$ over all existing states as follows:

$$\gamma_t(i) = \sum_{j=1}^{N}\xi_t(i,j)$$

By summing $\gamma_t(i)$ over time one get a number which can be interpreted as the number of times that state s_i is visited. This can also be interpreted as the number of transitions from state s_i. Similarly, summation of $\xi_t(i,j)$ over time can be interpreted as the number of transitions from state s_i to state s_j. By using these interpretations, a method for the re-estimation of the model parameters Π, A, B for the HMM is as follows:

$$\bar{\pi}_i = \gamma_1(i),$$

One should see that last equation can be interpreted as the frequency in state s_i at time $t = 1$. The next equation should be interpreted as the expected number of transitions from state s_i to s_j divided by the number of transitions from state s_i.

$$\bar{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)}$$

And finally, next can be seen as the expected number of times in state s_j and observing the symbol v_k, divided by the expected number of times in state s_j.

$$\bar{b}_j(v_k) = \frac{\sum_{\substack{t=1 \\ o_t = v_k}}^{T-1} \gamma_t(j)}{\sum_{t=1}^{T-1} \gamma_t(j)}$$

If the current HMM is defined as $\lambda = \{A, B, \Pi\}$ and used to compute the right hand side of last three equations, and at the same time define the re-estimation HMM as $\bar{\lambda} = \{\bar{A}, \bar{B}, \bar{\Pi}\}$ as determined from the left hand side of last three equations it has been proven that either:

1. The initial model λ defines a critical point of the likelihood function, in which case $\lambda = \bar{\lambda}$, or

2. Model $\bar{\lambda}$ is more likely than model λ in the sense that $P(O|\bar{\lambda}) > P(O|\lambda)$, which means that one have found a new model $\bar{\lambda}$ from which the observation sequence is more likely to have been produced.

An iterative re-estimation process, replacing λ with $\bar{\lambda}$ can be done to a certain extent, until some limiting point is reached. The final result of this re-estimation procedure is called a maximum likelihood estimation of the HMM.

An important aspect of the re-estimation procedure is that the stochastic constraints of the HMM parameters, namely

$$\sum_{i=1}^{N} \bar{\pi}_i = 1,$$

$$\sum_{j=1}^{N} \bar{a}_{ij} = 1, \qquad 1 \leq i \leq N$$

$$\sum_{k=1}^{M} \bar{b}_j(v_k) = 1, \qquad 1 \leq j \leq N$$

are automatically satisfied at each iteration.

4.6 Forecasting using HMM:

An HMM based tool for financial time series' forecasting for the currency market implemented. HMM is constructed by estimating parameter set $\lambda(A, B, \pi)$.While implementing the HMM, the choice of the model number of states and the observation symbol become a tedious task.

For simplicity, four input features for a currency exchange will be considered – that is the opening price, closing price, highest price, and the lowest price. The next days' highest and lowest price rates are taken as the target price associated with the four input features, and the observations are being discrete.

In this work, discrete HMM with three states will be used as a model. The states of HMM are: increasing state, decreasing state and no change state. Observation sequence is built using principal technical indicators. Initially parameter values are chosen randomly. The HMM was trained using training dataset, so that the values of A, B, π are re-estimated to suit the training dataset.

To predict the next day's closing price, we applied Baum-Welch algorithm. Suppose the likelihood value for the day x is lcp_i. Now from the historical data set observation sequences are located that would produce the same or nearly same value of lcp_i (locate past days where the Forex behavior is matched to the current day). HMM found many observations that produced the same likelihood value lcp_i. Then for each of the matched day, difference between match day and it next day is calculated using

$$wd_k = \frac{\sum_m W_m diff_m}{\sum_m W_m}$$

where W_m is weight assigned to day m, wd_k is weighted average of price difference for current day k, $diff_m$ is price difference between day m and $m+1$. Then forecast the value for day $k+1$, by

$$fp_{k+1} = p_k + wd_k$$

where, fp_{k+1} is the forecasted price (Highest or Lowest price) and p_k is current day price (Highest or Lowest price).

4.6.1 Principal Component Analysis (PCA):

Principal Component Analysis is a statistical technique that linearly transforms an original set of variables into a substantially smaller set of uncorrelated variables that represents most of the information in the original set of variables. Its goal is to reduce the dimensionality of the original data set. A small set of uncorrelated variables is much easier to understand and can be useful in further analyses than a larger set of correlated variables. [20]

In Principal Component Analysis, the variance of a matrix (Z) is explained in terms of new latent variables which are called Principal Components (PC). The first Principal Component variable is the linear combination of matrix element that has the greatest variance. The second Principal Component Variable (PCV) is the linear combination with the next greatest variance among coefficient vectors of unit length that are orthogonal to the first coefficient vector. In this manner, one can obtain k possible Principal Component Variables. The calculated Principal Component is given by

$$t_1 = p_1' z \text{ subject to } |p_1| = 1$$

$$t_1 = p_1' z \text{ subject to } |p_1| = 1$$

$$\text{and } p_2' p_1 = 0$$

The Principal Component loading vectors p are the eigenvectors of the covariance matrix Σ of Z and the corresponding eigenvalues λ_i are the variances of the Principal Components. Using loading vectors the observation can be written as

$$Z = \sum_{i=1}^{k} t_i p_i' + E$$

where k, is the number of Principal Components obtained and E is the residual matrix.[21]

Chapter 5:

System Implementation and Result Analysis

Chapter 5: System Implementation and Result Analysis

5.1 Introduction:

This thesis outlines a description of the current exchange price rate of a currency as states in an HMM model. These states are used as the basis to forecast next days currency prices. The HMM algorithm used to builds the system and to calculate the similarity between forecasted prices and the real prices. The system built using a MATLAB software application; the structure of the final software application is illustrated. Furthermore, the results of its performance are illustrated by a detailed example.

5.2 Design:

The experiment is conducted on three Forex pairs that are namely the USD/JPY, JPY/USD, EURO/JPY, JPY/EURO, EURO/USD and USD/EURO by taking six years daily data from 1-Jan-2006 to 31-Dec-2012. The total number of observations N is 2190. The data divided into training and testing data. Several technical indicators as inputs for the model is considered. They are EMA, RSI, PPMA, AMMA, DMI-DIP, DMI-DIM, Regression and ROC.

Principal Components are extracted using PCA function. It is a component matrix helped to determine principal components as shown in Table (5-1). The correlated components are EMA, PPMA, AMMA and Regression in order of decreasing correlation. Therefore we selected EMA, PPMA, AMMA and Regression for further analyses. These four indicators are used to make observation sequences in HMM.

Component Matrix				
	Component			
	1	2	3	4

Table (5-1): Determining Principal Component Matrix

Three hidden states are assumed, Increasing, Decreasing and No change.

- When $C_n - C_{n-1} > 0$, it is taken as increasing state.
- When $C_n - C_{n-1} > 0$, it is taken as decreasing state.
- When $C_n - C_{n-1} = 0$, it is in no change state.

where C_n is current closing price and C_{n-1} is the previous closing price. The states transition probability is expressed as

$$\begin{array}{c c} & \begin{array}{ccc} Increase & Decrease & No\ change \end{array} \\ \begin{array}{c} Increase \\ Decrease \\ No\ change \end{array} & \begin{bmatrix} Increase/Increase & Increase/Decrease & Increase/No\ change \\ Decrease/Increase & Decrease/Decrease & Decrease/No\ change \\ No\ change/Increase & No\ change/Decrease & No\ change/No\ change \end{bmatrix} \end{array}$$

The transition states from 1st January 2006 to 31st December 2012 is shown in Figure (5-1).

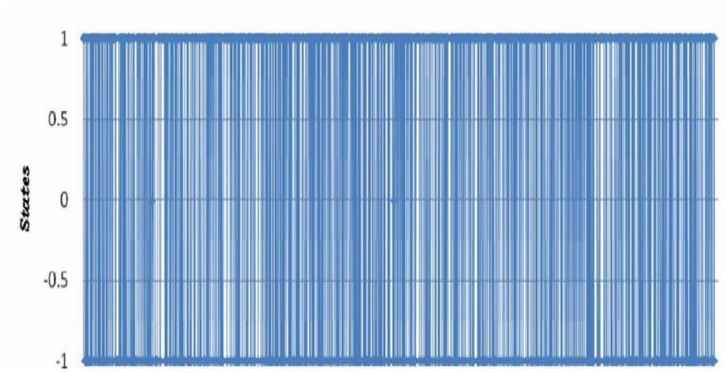


Figure (5-1): State Transition from 1st January 2002 to 31th December 2012

In the side of previous Figure 1 indicates increasing state, -1 indicates decreasing state and 0 indicates no change state.

For finding the trend of, we need to find state transition probability. The result is obtained in the following way

$$increase \Rightarrow increase : 369\ days$$

$$increase \Rightarrow decrease : 396\ days$$

$$increase \Rightarrow no\ change : 1\ day$$

$$decrease \Longrightarrow increase : 404\ days$$

$$decrease \square decrease : \square 577\ days$$

$$decrease\ \square no\ change : 0\ day$$

$$no\ change\ \square increase : 1\ day$$

$$no\ change\ \square decrease : 1\ day$$

$$no\ change\ \square no\ change : 0\ day$$

We get transition matrix as

Figure (5-2) illustrates the proposed algorithm in a brief:

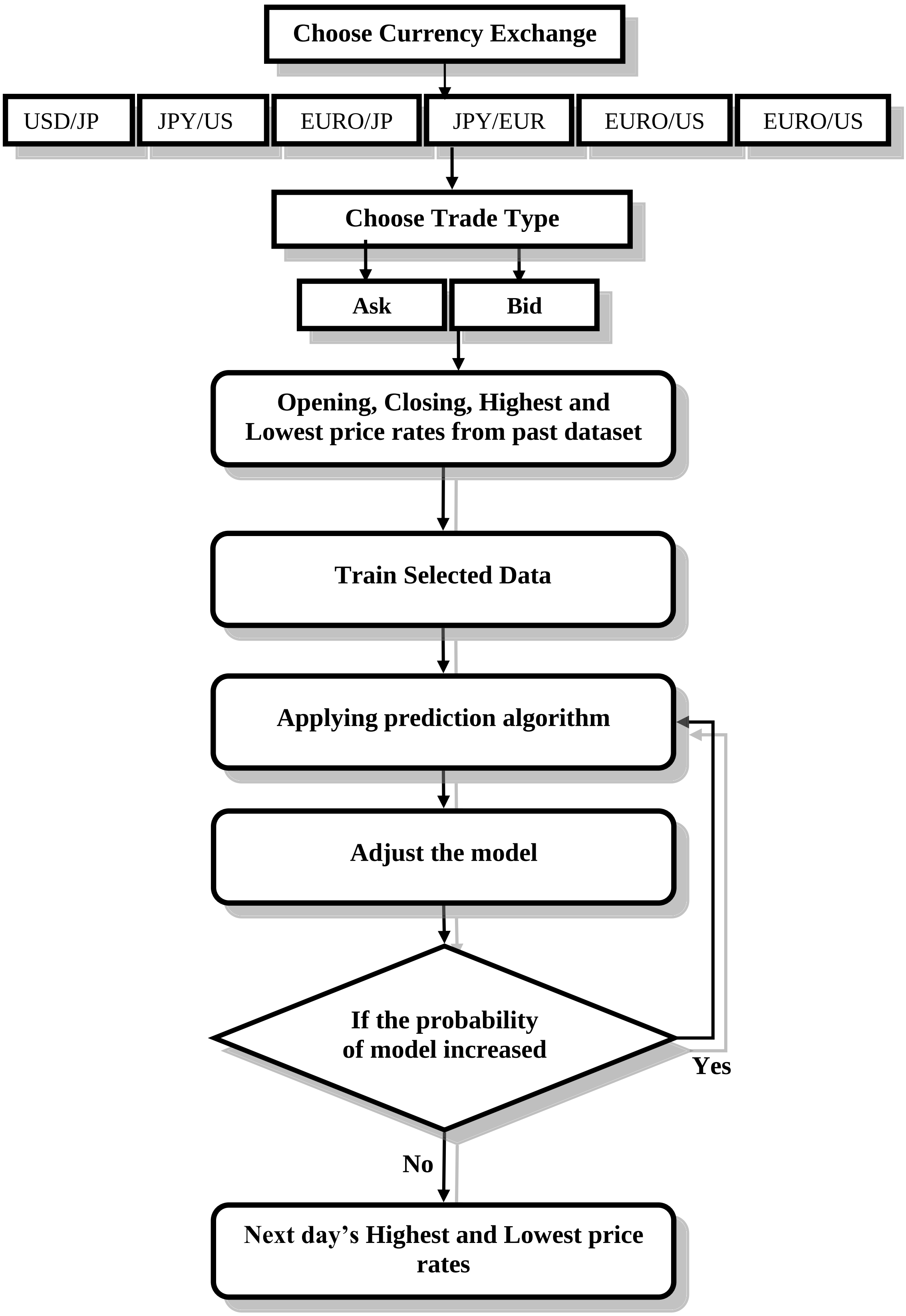


Figure 5-3: Proposed algorithm chart

5.2.1 Adjustment process of the HMM:

1. Initialize $\lambda = \{A, B, \Pi\}$;
2. Compute $\alpha_t(i)$, $\beta_t(i)$, $\xi_t(i,j)$, $\gamma_t(i)$, $t = 1, \dots T$, $i = 0, \dots, N-1$, $j = 0, \dots, N-1$;
3. Adjust the model $\lambda = \{A, B, \Pi\}$;
4. If $P(O|\lambda)$ increases, go to 2.

5.2.2 Initializing the model:

- The transition probabilities between the hidden states $A(N \times N) = \{a_{ij}\}$, are randomly initialized to approximately $1/N$, each row summing to 1.
- The probabilities of the observable states $B(N \times M) = \{b_j(k)\}$, are randomly initialized to approximately $1/M$, each row summing to 1.
- The initial hidden state probabilities $\Pi(1 \times N) = \{\pi_i\}$ are randomly set to approximately $1/N$, their sum being 1.

5.2.3 Prediction Algorithm using HMM:

1. $T = 1$ (T is the length of the observation sequence);
2. $T = T + 1$;
3. If $T < H$ go to step (2).
4. $c = 0$ (c is the number of current iteration, its maximum value is given by I);
5. The model $\lambda = \{A, B, \Pi\}$ is repeatedly adjusted based on the last H obs ervations $O_{T-H+1}, O_{T-H+2}, \dots, O_T$ (the entire observation sequence if $H = T$), in order to increase the probability of the observation sequence $P(O_{T-H+1}\ O_{T-H+2}\ \dots\ O_T|\lambda)$. In the following steps the denominators are used in order to obtain a probability measure, and to avoid underflow.
    - Compute the forward variable α in a recursive manner.

- Compute the backward variable β in a recursive manner.
- Compute ξ and γ.
- Adjust Π, A and B.

6. $c = c + 1$.

7. If $\log\left[P\left(O_{T-H+1} \dots O_T \middle| \bar{\lambda}\right)\right] > \log[P(O_{T-H+1} \dots O_T|\lambda)]$ and $c < I$ then go to step (5).

8. At current time T, it is predicted the next observation symbol O_{T+1}, using the adjusted model $\bar{\lambda} = (\bar{A}, \bar{B}, \bar{\Pi})$:

- Choose a hidden state S_i at time T, $i = 0, \dots, N - 1$, maximizing α_T;
- Choose next hidden state S_j (at time $T + 1$), $j = 0, \dots, N - 1$, maximizing $\bar{a}_{ij}$;
- Predict next symbol V_k (at time T+1), $k = 0, \dots, M - 1$, maximizing $\bar{b}_j(k)$.

9. If the process continues, then T=T+1 and go to step (4).

5.3 Graphical User Interface (GUI):

The Graphical User Interface was constructed using MATLAB GUIDE (Graphical User Interface Design Environment). Using the layout tools provided by GUIDE, designed the following graphical user interface Figure (5-2) for this Forex Forecasting application:

Figure 5-2: The system interface of Forex Forecasting

The handlers for clicking on the buttons are coded using MATLAB code to perform the necessary operations. To demonstrate the project application, implemented the following example:

The application started by typing guide and pressing return in the MATLAB GUI Command Window. After that, it started by typing FirstGUI and pressing return in MATLAB Command Window then the application window started.

At the application window, select the "Enter" button. If wrong password is typed, a message will display about wrong password. If correct password is typed, new window that have menu options will displayed. Figure (5-4) shows system-entering process.

Figure 5-3: The system interface of Entering process

In addition to the above outlined design, also designed a simple menu structure, using the new window, as shown below in Figure (5-4):

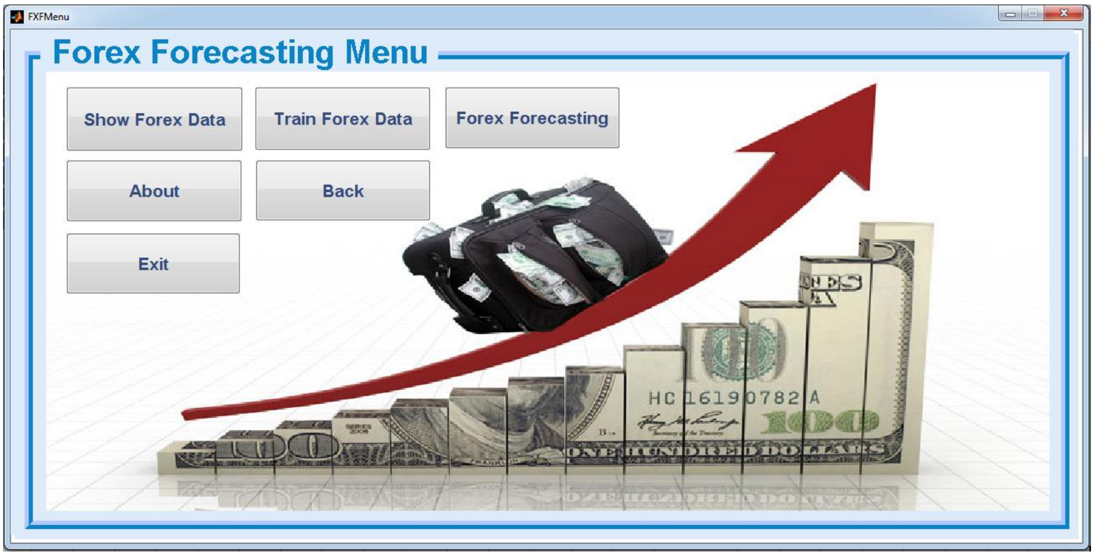


Figure 5-4: The system interface of Forex Forecasting Menu

If Show Forex Data button is selected, a new window will display as Figure (5-5). To view data, select currency exchange type and trade type (ask-bid), and select View button. Then, the database will be displaying on screen. If Save button is pressed, data will saved. Press Back button to return to the previous interface, and Exit button to exit from program system.

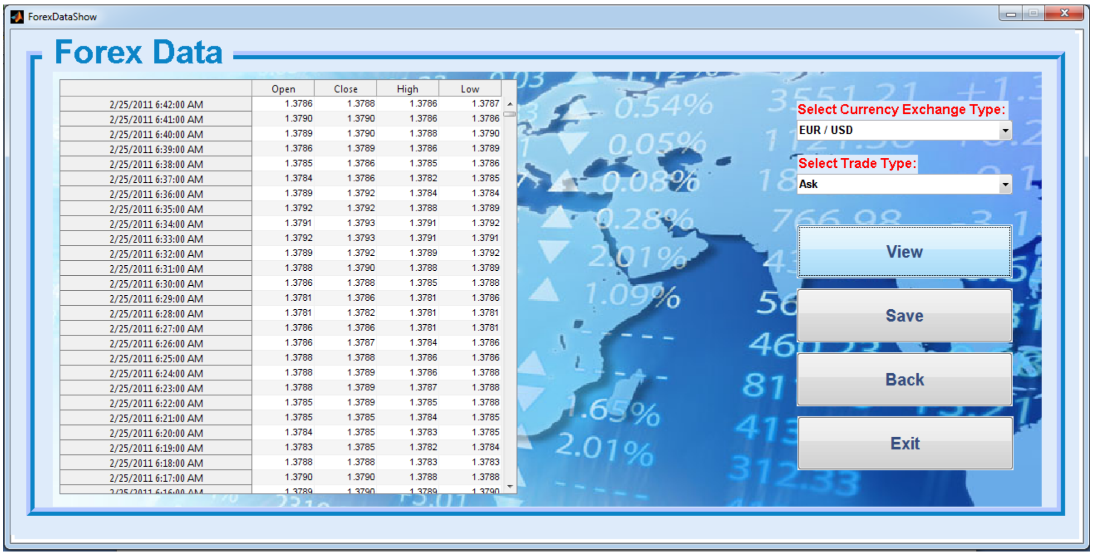


Figure 5-5: The system interface of Forex Forecasting Data View

From the main menu (Figure (5-4)), select currency exchange type and trade type (ask-bid), and press Train button. Then, the data will be training. Press Back button to return to the previous interface, and Exit button to exit from program system (see Figure (5-6)).

Figure 5-6: The system interface of Forex Data Training

From the main menu (Figure (5-4)), select currency exchange type and trade type (ask-bid), and press Forecast button. Then, the data will forecast and display the predicted prices on screen. Press Back button to return to the previous interface, and Exit button to exit from program system (see Figure (5-6)).

Figure 5-7: The system interface of Forex Forecasting

From the main menu (Figure (5-4)), if About button is selected, browsing window will display, that have information about this thesis as in Figure (5-8). Press Back button to return to the previous interface, and Exit button to exit from program system.

Figure 5-8: The system interface of information about Forex Forecasting

From the main menu (Figure (5-4)), press on Back button to return to the previous window, as in Figure (5-9). Moreover, press Exit button to exit from the Forex Forecasting system as in Figure (5-10).

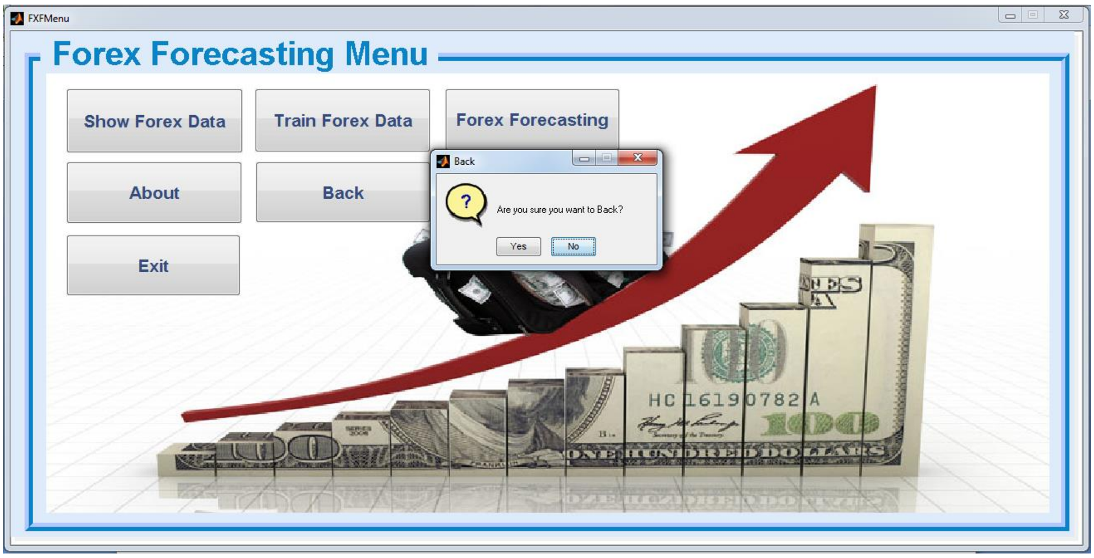


Figure 5-9: The system interface of pressing Back button

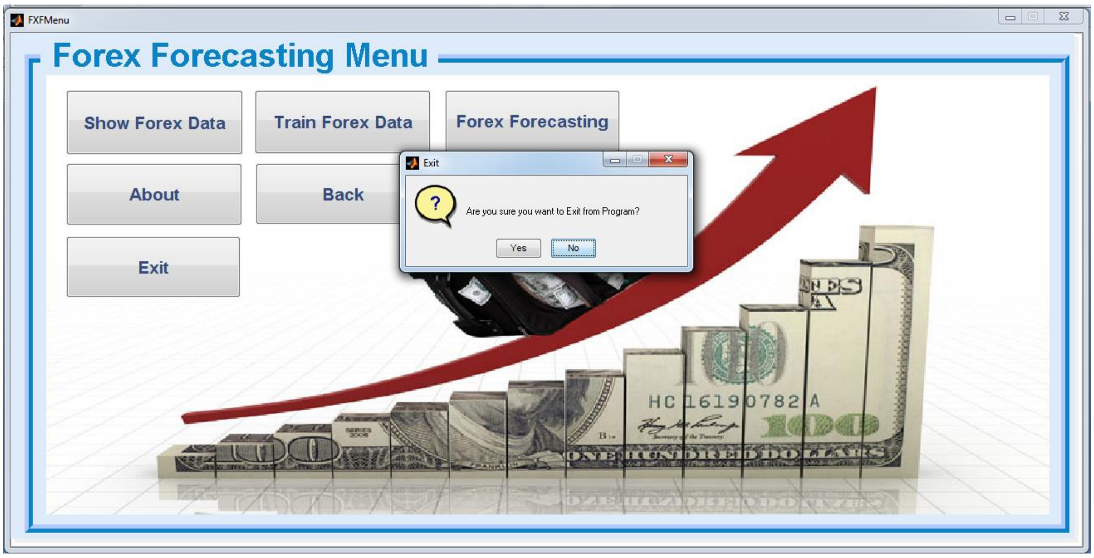


Figure 5-9: The system interface of pressing Exit button

5.4 Evaluation of Query Results:

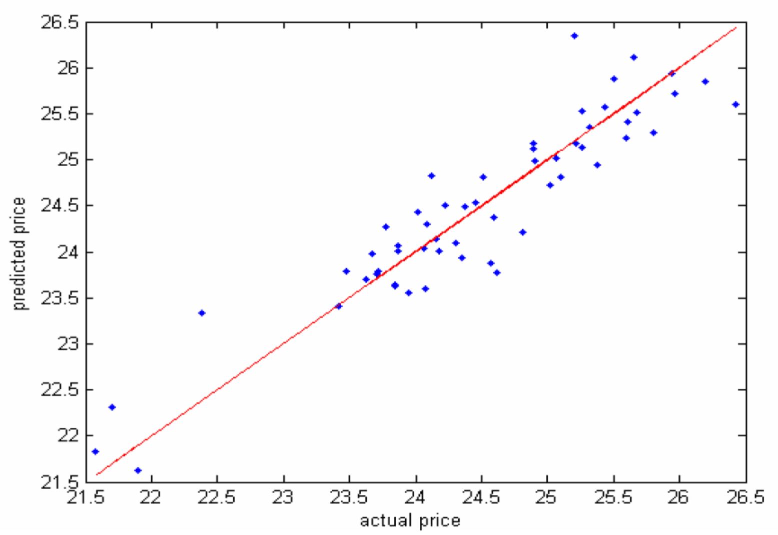
5.5 Results Analysis and Comparison:

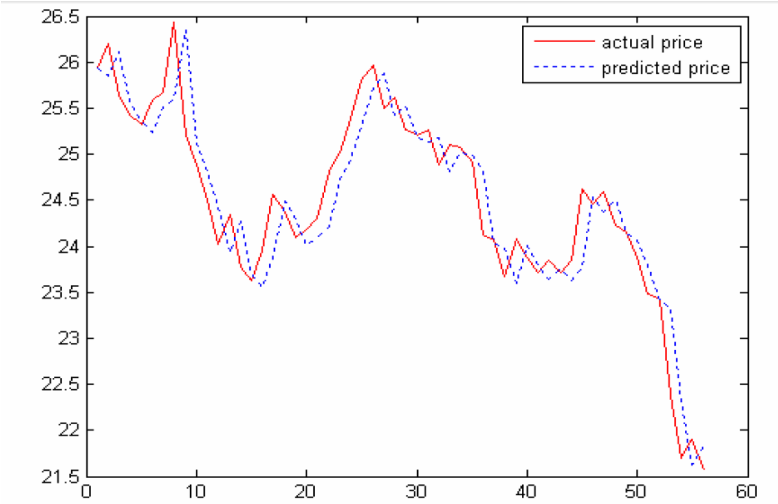
5.6 Model-Prediction Results:

The selection of Data Sets: Each of Data set contains historical prices from --/--/20 to --/--/20, Data Source is from Forex market, From any historical price providers we can get 5 kinds of daily prices: open, low, high and close prices.

The most significant advantage of this hidden Markov model while implemented with daily data is its predictability. The three figures below show the prediction performance on Forex prices.

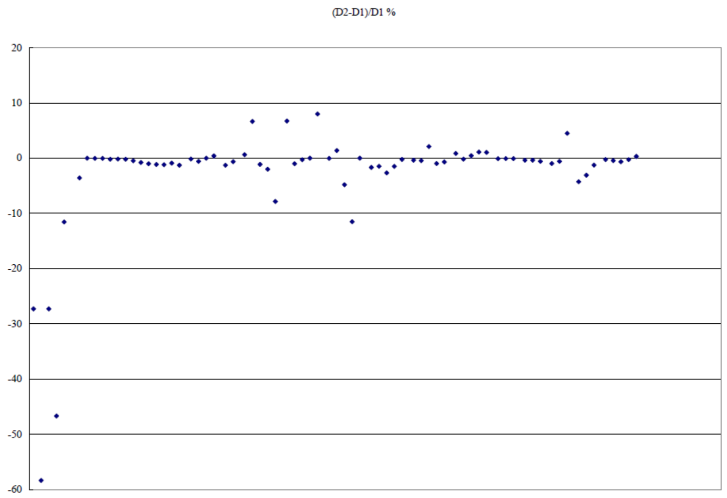
Take the prediction on USD/JPY for example. First, we plot the predicted prices on the y-axis and the actual prices on the x-axis. The first plot displays the prediction of prices on a range of 56 days. For each point, the x coordinate represents the actual price and the y coordinate represents the predicated price given information of the actual prices up to the last trading day. We can see that all points are closely gathered around the diagonal $y = x$. Generally it indicates a good prediction. To see how good it is, the second plot shows the whole paths of actual price (solid line) and predicted price (dashed line). Not only we find the predicted path closely resembles the actual path, but also we discover an interesting phenomenon. When the prices are increasing, we see under prediction, conversely when the prices are decreasing, we see over estimate. This phenomenon can be found with quite a lot of data sets. One might take this into account when using our program to predict the prices. And of course more work has to done to study the cause of this phenomenon.





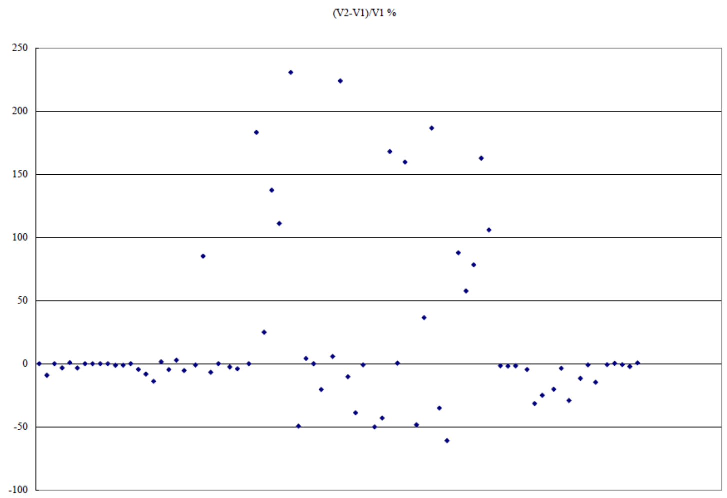
111

Historically most of the models assume constant drift. If that is indeed the case, we should not see much differences in the estimates of two states. The figure below plots $\frac{D_1 - D_2}{D_1}\%$ where $\{D_1, D_2\}$ is the estimate of the state space of the drift. The 73 points representing the results from 73 data sets. We find there are indeed two different states for the drift (except for a few points lie on the line zero). Moreover, the difference in the two states is .

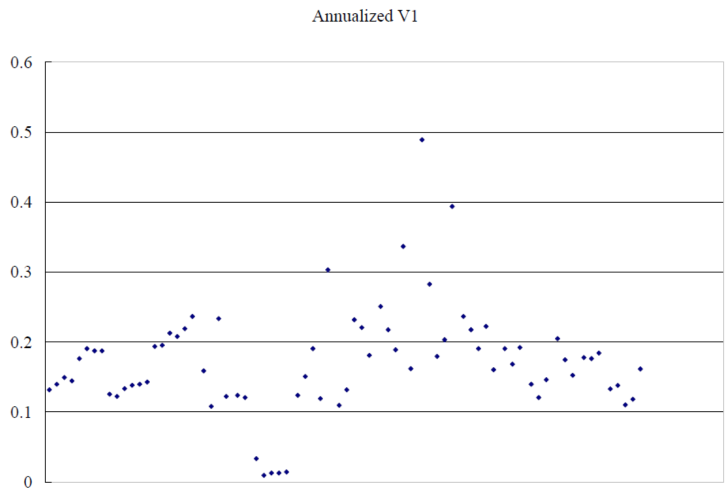


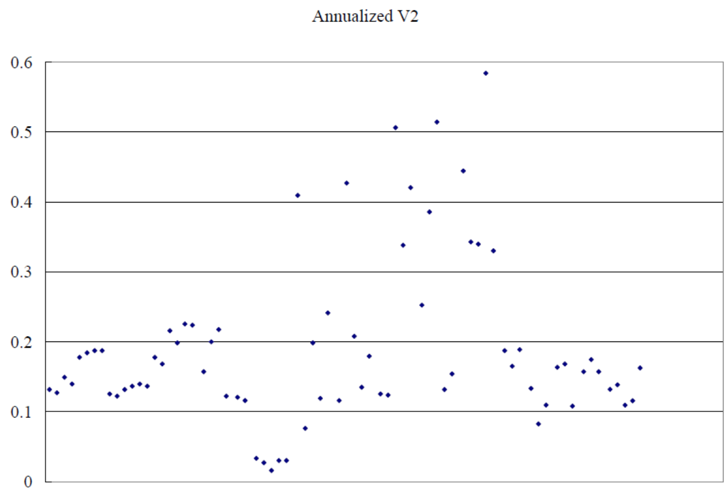
There is another interesting finding, during the period from Mar 2001 to Oct 2006, when most of the U.S. indices suggest negative drifts, the estimate of the drifts for the foreign indices (Hang Seng, Nikkei, FTSE) are positive for both states.

Next we plot $\frac{V_1-V_2}{V_1}\%$ where $\{V_1, V_2\}$ is the estimate of the state space of the volatility. Comparatively, the difference in the two states of volatilities (mostly from -50% to $+100\%$) is a lot more significant than the difference in the two states of drifts (mostly from -10% to $+10\%$). Therefore, for analytic reason, some practices can assume constant mean.



According to our estimates, the typical values of annualized volatility are within 10% to 30%, as indicated from the next two plots.





Moreover, when we look at the transition probabilities, the overall financial market (indicated by the indices) has pretty close probabilities of staying in each state which is indicative to a steady movement of the price process. However, some of the individual stocks such as XOM, PFE have big differences in the transition probabilities, which is an evidence that one state dominant the other over that period.

Chapter 6:

Conclusions and Further Works

Chapter 6: Conclusions and Further Works

In this chapter, the work concluded and provided some further avenues for the future work.

6.1 Conclusions:

This work starts with summarizing and comparing all the popular models for financial market return process on both a theoretic level and an empirical level. While judging the prediction performance, we employ the concepts of "relative α" and "relative standard error" as modification to the three criteria for a good model. This modification has been proved to be more appropriate.

6.2 Further Work:

In addition to the improvements and developments we have made in this research, there are further avenues for future work that could be done.

Bibliography

Bibliography

[1] "FXCM Trading Station Program," 01.08.040910 ed: FXCM, 2010.

[2] X. Wang, M. D. Sykora, R. Archer, D. Parish, and H. E. Bez, "Case Based Reasoning Approach for Transaction Outcomes Prediction on Currency Markets," 2009, pp. 93-98.

[3] R. J. Kuo, L. C. Lee, and C. F. Lee, "Integration of Artificial Neural Networks and Fuzzy Delphi for Stock Market Forecasting," in *IEEE International Conference on Systems, Man, and Cybernetics*, 1996, pp. 1073-1078.

[4] E. H. K. Fung and A. P. L. Chung, "Using ARMA Models to Forecast Workpiece Roundness Error in a Turning Operation," *Applied Mathematical Modelling*, vol. 23, pp. 567-585, 1999.

[5] M. H. Eng, Y. Li, Q. G. Wang, and T. H. Lee, "Forecast Forex with Artificial Neural Network Using Fundamental Data," 2009, pp. 279-282.

[6] A. K. Lohani, N. K. Goel, and K. K. S. Bhatia, "Development of Fuzzy Logic Based Real Time Flood Forecasting System for River Narmada in Central India," 2005.

[7] E. Abbasi and A. Abouec, "Stock Price Forecast by Using Neuro-Fuzzy Inference System," 2008, pp. 320-323.

[8] K. Kim, "Financial Time Series Forecasting Using Support Vector Machines," *Neurocomputing*, vol. 55, pp. 307-319, 2003.

[9] K. Slany, "Towards the Automatic Evolutionary Prediction of the FOREX Market Behaviour," 2009, pp. 141-145.

[10] M. R. Hassan and B. Nath, "Stock Market Forecasting using Hidden Markov Model: A New Approach," in *Proceedings 5th International Conference on Intelligent Systems Design and Applications (ISDA'05)*, 2005, pp. 192-196.

[11] R. Nag, K. Wong, and F. Fallside, "Script Recognition using Hidden Markov Models," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'86)*, 1986, pp. 2071-2074.

[12] Y. Bengio, Y. LeCun, C. Nohl, and C. Burges, "A NN/HMM Hybrid for On-Line Handwriting Recognition," *Neural Computation*, vol. 7, pp. 1289-1303, 1995.

[13] A. Krogh, M. Brown, I. S. Mian, K. Sj olander, and D. Haussler, "Hidden Markov models in Computational Biology: Applications to Protein Modeling," *J. Mol. Biol*, vol. 235, pp. 1501-1531, 1994.

[14] P. Baldi and S. Brunak, *Bioinformatics: The Machine Learning Approach*: The MIT Press, 2001.

[15] P. Smyth, "Hidden Markov Models for Fault Detection in Dynamic Systems," *Pattern Recognition*, vol. 27, pp. 149-164, 1994.

[16] "Bank of International Settlement," Triennial Central Bank Survey - Foreign exchange and derivatives market activity in 2007.

[17] *Forex-Tell me about it!*, July 2006 ed.: Easy Forex Trading Platform, 2006.

[18] L. R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," *Proceedings of the IEEE*, vol. 77, pp. 257-286, 1989.

[19] D. Jurafsky and J. H. Martin, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, 2nd ed.: Pearson Prentice Hall, London, 2006.

[20] G. H. Dunteman, Principal Component Analysis, Saga, 1989.

[21] B. Abraham, N. U. Nair, Quality Improvement through Statistical Methods, Springer, Canada, 1998.

Appendices

Appendices

Appendix A:

A. Estimates of Drift, Volatility, Transition

B. Prediction Performance