Master's Thesis

Algorithmic Trading
Hidden Markov Models on Foreign Exchange Data

Patrik Idvall, Conny Jonsson

LiTH - MAT - EX - - 08 / 01 - - SE

Algorithmic Trading
Hidden Markov Models on Foreign Exchange Data

Department of Mathematics, Linköpings Universitet

Patrik Idvall, Conny Jonsson

LiTH - MAT - EX - - 08 / 01 - - SE

Master’s Thesis: **30 hp**

Level: **A**

Supervisor: **J. Blomvall**,
Department of Mathematics, Linköpings Universitet

Examiner: **J. Blomvall**,
Department of Mathematics, Linköpings Universitet

Linköping: **January 2008**

LINKÖPINGS UNIVERSITET

Avdelning, Institution
Division, Department

Matematiska Institutionen
581 83 LINKÖPING
SWEDEN

Datum
Date

January 2008

Språk
Language

☐ Svenska/Swedish
☒ Engelska/English

☐ ______________

Rapporttyp
Report category

☐ Licentiatavhandling
☒ Examensarbete
☐ C-uppsats
☐ D-uppsats
☐ Övrig rapport
☐ ______________

ISBN

ISRN
LiTH - MAT - EX - - 08 / 01 - - SE

Serietitel och serienummer
Title of series, numbering

ISSN

URL för elektronisk version

http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-10719

Titel
Title

Algorithmic Trading – Hidden Markov Models on Foreign Exchange Data

Författare
Author

Patrik Idvall, Conny Jonsson

Sammanfattning
Abstract

In this master's thesis, hidden Markov models (HMM) are evaluated as a tool for forecasting movements in a currency cross. With an ever increasing electronic market, making way for more automated trading, or so called algorithmic trading, there is constantly a need for new trading strategies trying to find alpha, the excess return, in the market.
HMMs are based on the well-known theories of Markov chains, but where the states are assumed hidden, governing some observable output. HMMs have mainly been used for speech recognition and communication systems, but have lately also been utilized on financial time series with encouraging results. Both discrete and continuous versions of the model will be tested, as well as single- and multivariate input data.
In addition to the basic framework, two extensions are implemented in the belief that they will further improve the prediction capabilities of the HMM. The first is a Gaussian mixture model (GMM), where one for each state assign a set of single Gaussians that are weighted together to replicate the density function of the stochastic process. This opens up for modeling non-normal distributions, which is often assumed for foreign exchange data. The second is an exponentially weighted expectation maximization (EWEM) algorithm, which takes time attenuation in consideration when re-estimating the parameters of the model. This allows for keeping old trends in mind while more recent patterns at the same time are given more attention.
Empirical results shows that the HMM using continuous emission probabilities can, for some model settings, generate acceptable returns with Sharpe ratios well over one, whilst the discrete in general performs poorly. The GMM therefore seems to be an highly needed complement to the HMM for functionality. The EWEM however does not improve results as one might have expected. Our general impression is that the predictor using HMMs that we have developed and tested is too unstable to be taken in as a trading tool on foreign exchange data, with too many factors influencing the results. More research and development is called for.

Nyckelord
Keyword

Algorithmic Trading, Exponentially Weighted Expectation Maximization Algorithm, Foreign Exchange, Gaussian Mixture Models, Hidden Markov Models

Abstract

In this master's thesis, hidden Markov models (HMM) are evaluated as a tool for forecasting movements in a currency cross. With an ever increasing electronic market, making way for more automated trading, or so called algorithmic trading, there is constantly a need for new trading strategies trying to find alpha, the excess return, in the market.

HMMs are based on the well-known theories of Markov chains, but where the states are assumed hidden, governing some observable output. HMMs have mainly been used for speech recognition and communication systems, but have lately also been utilized on financial time series with encouraging results. Both discrete and continuous versions of the model will be tested, as well as single- and multivariate input data.

In addition to the basic framework, two extensions are implemented in the belief that they will further improve the prediction capabilities of the HMM. The first is a Gaussian mixture model (GMM), where one for each state assign a set of single Gaussians that are weighted together to replicate the density function of the stochastic process. This opens up for modeling non-normal distributions, which is often assumed for foreign exchange data. The second is an exponentially weighted expectation maximization (EWEM) algorithm, which takes time attenuation in consideration when re-estimating the parameters of the model. This allows for keeping old trends in mind while more recent patterns at the same time are given more attention.

Empirical results shows that the HMM using continuous emission probabilities can, for some model settings, generate acceptable returns with Sharpe ratios well over one, whilst the discrete in general performs poorly. The GMM therefore seems to be an highly needed complement to the HMM for functionality. The EWEM however does not improve results as one might have expected. Our general impression is that the predictor using HMMs that we have developed and tested is too unstable to be taken in as a trading tool on foreign exchange data, with too many factors influencing the results. More research and development is called for.

Keywords: Algorithmic Trading, Exponentially Weighted Expectation Maximization Algorithm, Foreign Exchange, Gaussian Mixture Models, Hidden Markov Models

Acknowledgements

Writing this master's thesis in cooperation with Nordea Markets has been a truly rewarding and exciting experience. During the thesis we have experienced great support and interest from many different individuals.

First of all we would like to thank Per Brugge, Head of Marketing & Global Business Development at Nordea, who initiated the possibility of this master's thesis and gave us the privilege to carry it out at Nordea Markets in Copenhagen. We thank him for his time and effort! Erik Alpkvist at e-Markets, Algorithmic Trading, Nordea Markets in Copenhagen who has been our supervisor at Nordea throughout this project; it has been a true pleasure to work with him, welcoming us so warmly and for believing in what we could achieve, for inspiration and ideas!

Jörgen Blomvall at Linköpings Universitet, Department of Mathematics has also been very supportive during the writing of this thesis. It has been a great opportunity to work with him! We would also like to thank our colleagues and opponents Viktor Bremer and Anders Samuelsson, for helpful comments during the process of putting this master's thesis together.

Patrik Idvall & Conny Jonsson
Copenhagen, January 2008

Nomenclature

Most of the reoccurring symbols and abbreviations are described here.

Symbols

$S = \{s_1, s_2, ..., s_N\}$	a set of N hidden states,
$Q = \{q_1, q_2, ..., q_T\}$	a state sequence of length T taking values from S,
$O = \{o_1, o_2, ..., o_T\}$	a sequence consisting of T observations,
$A = \{a_{11}, a_{12}, ..., a_{NN}\}$	the transition probability matrix A,
$B = b_i(o_t)$	a sequence of observation likelihoods,
$\Pi = \{\pi_1, \pi_2, ..., \pi_N\}$	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 O and λ,
μ_{im}	the mean for state s_i and mixture component m,
Σ_{im}	the covariance matrix for state s_i and mixture component m,
$\rho = \{\rho_1, \rho_2, \dots, \rho_t\}$	a vector of real valued weights for the Exponentially Weighted Expectation Maximization,
w_{im}	the weights for the mth mixture component in state s_i,
$\xi_t(i, j)$	the joint probability of $q_t = s_i$ and $q_{t+1} = s_j$ 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,
$\psi_t j$	the state s_i at time t which gives us $\delta_t(j)$, used for backtracking.

Abbreviations

AT	Algorithmic Trading
CAPM	Capital Asset Pricing Model
EM	Expectation Maximization
ERM	Exchange Rate Mechanism
EWEM	Exponentially Weighted Expectation Maximization
EWMA	Exponentially Weighted Moving Average
FX	Foreign Exchange
GMM	Gaussian Mixture Model
HMM	Hidden Markov Model
LIBOR	London Interbank Offered Rate
MC	Monte Carlo
MDD	Maximum Drawdown
OTC	Over the Counter
PPP	Purchasing Power Parity
VaR	Value at Risk

Contents

List of Figures

Chapter 1

Introduction

In this first chapter a description of the background to this master's thesis will be given. It will focus on the foreign exchange market on the basis of its structure, participants and recent trends. The objective will be pointed out, considering details such as the purpose of the thesis and the delimitations that have been considered throughout the investigation. In the end of the objectives the disposal will be gone through, just to give the reader a clearer view of the thesis' different parts and its mutual order.

1.1 The Foreign Exchange Market

The Foreign Exchange (FX) market is considered the largest and most liquid[1] of all financial markets with a $3.2 trillion daily turnover. Between 2004 and 2007 the daily turnover has increased with as much as 71 percent. [19]

The increasing growth in turnover over the last couple of years, seen in figure 1.1 seems to be led by two related factors. First of all, the presence of trends and higher volatility in FX markets between 2001 and 2004, led to an increase of momentum trading, where investors took large positions in currencies that followed appreciating trends and short positions in decreasing currencies. These trends also induced an increase in hedging activity, which further supported trading volumes. [20]

Second, interest differentials encouraged so called carry trading, i.e. investments in high interest rate currencies financed by short positions in low interest rate currencies, if the target currencies, like the Australian dollar, tended to appreciate against the funding currencies, like the US dollar. Such strategies fed back into prices and supported the persistence of trends in exchange rates. In addition, in the context of a global search for yield, so called real money managers[2] and leveraged[3] investors became increasingly interested in foreign exchange as an asset class alternative to equity and fixed income.

As one can see in figure 1.1, the trend is also consistent from 2004 and forward, with more and more money put into the global FX market. The number of participants

[1] The degree to which an asset or security can be bought or sold in the market without affecting the asset's price.

[2] Real money managers are market players, such as pension funds, insurance companies and corporate treasurers, who invest their own funds. This distinguishes them from leveraged investors, such as for example hedge funds, that borrow substantial amounts of money.

[3] Leverage is the use of various financial instruments or borrowed capital, such as margin, to increase the potential return of an investment.

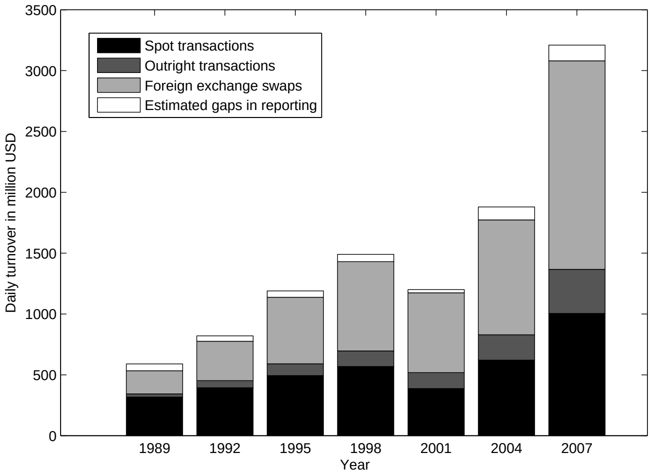


Figure 1.1: Daily return on the global FX market 1989-2007

and the share of the participants portfolio towards FX are continuously increasing, comparing to other asset classes.

1.1.1 Market Structure

The FX market is unlike the stock market an over the counter (OTC) market. There is no single physical located place were trades between different players are settled, meaning that all participants do not have access to the same price. Instead 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. [19]

The participants of the FX market can roughly be divided into the following five groups, characterized by different levels of access:

- Central Banks
- Commercial Banks
- Non-bank Financial Entities
- Commercial Companies
- Retail Traders

Central banks have a significant influence in FX markets by virtue of their role in controlling their countries' money supply, inflation, and/or interest rates. They may also have to satisfy official/unofficial target rates for their currencies. While they may have substantial foreign exchange reserves that they can sell in order to support their own currency, highly overt intervention, or the stated threat of it, has become less common in recent years. While such intervention can indeed have the desired effect, there have been several high profile instances where it has failed spectacularly, such as Sterling's exit from the Exchange Rate Mechanism (ERM) in 1992.

Through their responsibility for money supply, central banks obviously have a considerable influence on both commercial and investment banks. An anti-inflationary regime that restricts credit or makes it expensive has an effect upon the nature and level of economic activity, e.g. export expansion, that can feed through into changes in FX market activity and behaviour.

At the next level we have commercial banks. This level constitute the inter-bank section of the FX market and consist of participants such as Deutsche Bank, UBS AG, City Group and many others including Swedish banks such as Nordea, SEB, Swedbank and Handelsbanken.

Within the inter-bank market, spreads, which are the difference between the bid and ask prices, are sharp and usually close to non-existent. These counterparts act as market makers toward customers demanding the ability to trade currencies, meaning that they determine the market price.

As you descend the levels of access, from commercial banks to retail traders, the difference between the bid and ask prices widens, which is a consequence of volume. If a trader can guarantee large numbers of transactions for large amounts, they can demand a smaller difference between the bid and ask price, which is referred to as a better spread. This also implies that the spread is wider for currencies with less frequent transactions. The spread has an important role in the FX market, more important than in the stock market, because it is equivalent to the transaction cost. [22]

When speculating in the currency market you speculate in currency pairs, which describes the relative price of one currency relative to another. If you believe that currency A will strengthen against currency B you will go long in currency pair A/B. The largest currencies is the global FX market is US Dollar (USD), Euro (EUR) and Japanese Yen (JPY). USD stands for as much as 86 percent of all transactions, followed by EUR (37 percent) and JPY (17 percent).[4] [19]

1.2 Shifting to Electronic Markets

Since the era of floating exchange rates began in the early 1970s, technical trading has become widespread in the stock market as well as the FX markets. The trend in the financial markets industry in general is the increased automation of the trading, so called algorithmic trading (AT), at electronic exchanges. [2]

Trading financial instruments has historically required face-to-face contact between the market participants. The Nasdaq OTC market was one of the first to switch from physical interaction to technological solutions, and today many other market places has also replaced the old systems. Some markets, like the New York Stock Exchange (NYSE), still uses physical trading but has at the same time opened up some functions to electronic trading. It is of course innovations in computing and communications that has made this shift possible, with global electronic order routing, broad dissemination of quote and trade information, and new types of trading systems. New technology also reduces the costs of building new trading systems, hence lowering the entry barriers for new entrants. Finally, electronic markets also enables for a faster order routing and data transmission to a much larger group of investors than before. The growth in electronic

[4] Because two currencies are involved in each transaction, the sum of the percentage shares of individual currencies totals 200 percent instead of 100 percent.

trade execution has been explosive on an international basis, and new stock markets are almost exclusively electronic. [11]

1.2.1 Changes in the Foreign Exchange Market

The FX market differs somewhat from stock markets. These have traditionally been dealer markets that operate over the telephone, and physical locations where trading takes place has been non-existing. In such a market, trade transparency is low. But for the last years, more and more of the FX markets has moved over to electronic trading. Reuters and Electronic Broking Service (EBS) developed two major electronic systems for providing quotes, which later on turned into full trading platforms, allowing also for trade execution. In 1998, electronic trading accounted for 50 percent of all FX trading, and it has continued rising ever since. Most of the inter-dealer trading nowadays takes place on electronic markets, while trading between large corporations and dealers still remain mostly telephone based. [11] Electronic trading platforms for companies have however been developed, like Nordea e-Markets, allowing companies to place and execute trades without interaction from a dealer.

1.3 Algorithmic Trading

During the last years, the trend towards electronic markets and automated trading has, as mentioned in section 1.2, been significant. As a part of this many financial firms has inherited trading via so called algorithms[5], to standardize and automate their trading strategies in some sense. As a central theme in this thesis AT deserves further clarification and explanation. In this section different views of AT is given, to present a definition of the term.

In general AT can be described as trading, with some elements being executed by an algorithm. The participation of algorithms enables automated employment of predefined trading strategies. Trading strategies are automated by defining a sequence of instructions executed by a computer, with little or no human intervention. AT is the common denominator for different trends in the area of electronic trading that result in increased automation in:

1. Identifying investment opportunities (what to trade).
2. Executing orders for a variety of asset classes (when, how and where to trade).

This includes a broad variety of solutions employed by traders in different markets, trading different assets. The common denominator for the most of these solutions is the process from using data series for pre-trade analysis to final trade execution. The execution can be made by the computer itself or via a human trader. In figure 1.2 one can see a schematic figure over this process. This figure is not limited to traders using AT; rather it is a general description of some of the main elements of trading. The four steps presented in figure 1.2 defines the main elements of trading. How many of these steps that are performed by a computer is different between different traders and is a measure of how automated the process is. The first step includes analysis of market data as well as adequate external news. The analysis is often supported by computer tools such as spreadsheets or charts. The analysis is a very important step, which will end up

[5] A step-by-step problem-solving procedure, especially an established, recursive computational procedure for solving a problem in a finite number of steps. [21]

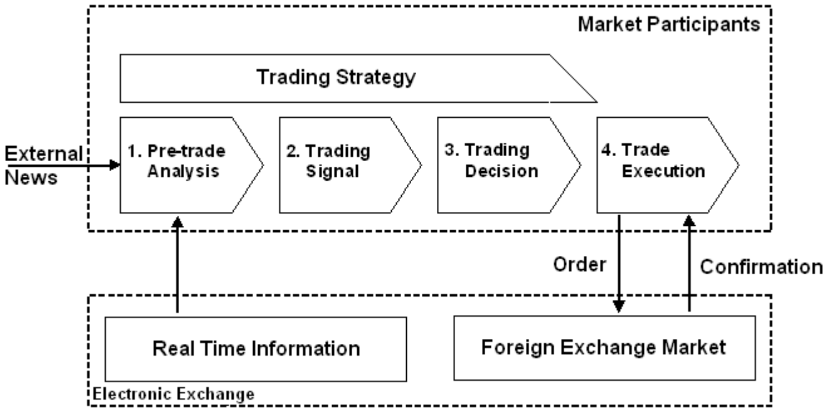


Figure 1.2: Schematic picture over the process for trading

in a trading signal and a trading decision in line with the overlaying trading strategy. The last step of the process is the real execution of the trading decision. This can be made automatically by a computer or by a human trader. The trade execution contains an order, sent to the foreign exchange market and the response as a confirmation from the same exchange. [3]

This simplified description might not hold for any specific trader taking into account the many different kinds of players that exists on a financial market. However, it shows that the trading process can be divided into different steps that follow sequentially. If one are now to suggest how trading is automated, there is little difficulty in discuss the different steps being separately programmed into algorithms and executed by a computer. Of course this is not a trivial task, especially as the complex considerations previously made by humans have to be translated into algorithms executed by machines. One other obstacle that must be dealt with in order to achieve totally automated trading is how to connect the different steps into a functional process and how to interface this with the external environment, in other words the exchange and the market data feed. In the next section different levels of AT will be presented. The levels are characterized by the number of stages, presented in figure 1.2, that are replaced by algorithms, performed by a machine.

1.3.1 Different Levels of Automation

A broad variety of definitions of AT is used, dependent on who is asked. The differences is often linked to the level of technological skill that the questioned individual possess. Based on which steps that are automated and the implied level of human intervention in the trading process, different categories can be distinguished. It is important to notice that the differences do not only consist of the number of steps automated. There can also be differences between sophistication and performance within the steps. This will not be taken in consideration here though, focusing on the level of automation.

Four different ways of defining AT are to be described, influenced by [3]. The first category, here named AT_1 presuppose that the first two steps are fully automated. They somewhat goes hand in hand because the pre-trade analysis often leads to a trading signal in one way or another. This means that the human intervention is limited to the two last tasks, namely the trading decision and the execution of the trade.

The second category, AT_2 is characterized by an automation of the last step in the trading process, namely the execution. The aim of execution algorithms is often to divide large trading volumes into smaller orders and thereby minimizing the adverse price impact a large order otherwise might suffer. It should be mentioned that different types of execution algorithms are often supplied by third party software, and also as a service to the buy-side investor of a brokerage. Using execution algorithms leaves the first three steps, analysis, trading signal and trading decision to the human trader.

If one combines the first two categories a third variant of AT is developed, namely AT_3. AT_3 is just leaving the trading decision to the human trader, i.e. letting algorithms taking care of step 1, 2 and 4.

Finally, fully automated AT, AT_4, often referred to as black-box-trading, is obtained if all four steps is replaced by machines performing according to algorithmically set decisions. This means that the human intervention is only control and supervision, programming and parameterizations. To be able to use systems, such as AT_3 and AT_4, great skills are required, both when it comes to IT solutions and algorithmic development.

Independent on what level of automation that is intended, one important issue is to be considered, especially if regarding high-frequency trading, namely the markets microstructure. The microstructure contains the markets characteristics when dealing with price, information, transaction costs, market design and other necessary features. In the next section a brief overview of this topic will be gone through, just to give the reader a feeling of its importance when implementing any level of algorithmic trading.

1.3.2 Market Microstructure

Market microstructure is the branch of financial economics that investigates trading and the organization of markets. [8] It is of great importance when trading is carried out on a daily basis or on an even higher frequency, where micro-based models, in contrary to macro-based, can account for a large part of variations in daily prices on financial assets. [4] And this is why the theory on market microstructure is also essential when understanding AT, taking advantage of for example swift changes in the market. This will not be gone through in-depth here, as it lies out of the this master's thesis' scope; the aim is rather to give a brief overview of the topic. Market microstructure mainly deals with four issues, which will be gone through in the following sections [5].

Price formation and price discovery

This factor focuses on the process by which the price for an asset is determined, and it is based on the demand and supply conditions for a given asset. Investors all have different views on the future prices of the asset, which makes them trade it at different prices, and price discovery is simply when these prices match and a trade takes place. Different ways of carrying out this match is through auctioning or negotiation. Quote-driven markets, as opposed to order-driven markets where price discovery takes place as just described, is where investors trade on the quoted prices set by the markets makers, making the price discovery happen quicker and thus the market more price efficient.

Transaction cost and timing cost

When an investor trades in the market, he or she faces two different kinds of costs: implicit and explicit. The latter are those easily identified, e.g. brokerage fees and/or taxes. Implicit however are described as hard to identify and measure. Market impact costs relates to the price change due to large trades in a short time and timing costs to the price change that can occur between decision and execution. Since the competition between brokers has led to significant reduction of the explicit costs, enhancing the returns is mainly a question of reducing the implicit. Trading in liquid markets and ensuring fast executions are ways of coping with this.

Market structure and design

This factor focuses on the relationship between price determination and trading rules. These two factors have a large impact on price discovery, liquidity and trading costs, and refers to attributes of a market defined in terms of trading rules, which amongst others include degree of continuity, transparency, price discovery, automation, protocols and off-markets trading.

Information and disclosure

This factor focuses on the market information and the impact of the information on the behavior of the market participants. A well-informed trader is more likely to avoid the risks related to the trading, than one less informed. Although the theory of market efficiency states that the market is anonymous and that all participants are equally informed, this is seldom the case which give some traders an advantage. When talking about market information, we here refer to information that has a direct impact on the market value of an asset.

1.3.3 Development of Algorithmic Trading

As the share of AT increases in a specific market, it provides positive feedback for further participants to automate their trading. Since algorithmic solutions benefits from faster and more detailed data, the operators in the market have started to offer these services on request from algorithmic traders. This new area of business makes the existing traders better of if they can deal with the increasing load of information. The result is that the non-automated competition will loose out on the algorithms even more than before. For this reason it is not bold to predict that as the profitability shifts in favor of AT, more trading will also shift in that direction. [3]

With a higher number of algorithmic traders in the market, there will be an increasing competition amongst them. This will most certain lead to decreasing margins, technical optimization and business development. Business development contains, amongst other things, innovation as firms using AT search for new strategies, conceptually different from existing ones. Active firms on the financial market put a great deal of effort into product development with the aim to find algorithms, capturing excess return by unveiling the present trends in the FX market.

1.4 Objectives

As the title of this master's thesis might give away, an investigation of the use of hidden Markov models (HMM) as an AT tool for investments on the FX market will be carried out. HMMs is one among many other frameworks such as artificial neural networks and support vector machines that could constitute a base of a successful algorithm.

The framework chosen to be investigated was given in advance from Algorithmic Trading at Nordea Markets in Copenhagen. Because of this initiation, other frameworks, as the two listed above, will not be investigated or compared to the HMMs throughout this study. Nordea is interested in how HMMs can be used as a base for developing an algorithm capturing excess return within the FX market, and from that our purpose do emerge.

1.4.1 Purpose

To evaluate the use of hidden Markov models as a tool for algorithmic trading on foreign exchange data.

1.4.2 Purpose Decomposition

The problem addressed in this study is, as we mentioned earlier, based on the insight that Nordea Markets is looking to increase its knowledge about tools for AT. This is necessary in order to be able to create new trading strategies for FX in the future. So, to be able to give a recommendation, the question to be answered is: *should Nordea use hidden Markov models as a strategy for algorithmic trading on foreign exchange data?*

To find an answer to this one have to come up with a way to see if the framework of HMMs can be used to generate a rate of return that under risk adjusted manners exceeds the overall market return. To do so one have to investigate the framework of HMMs in great detail to see how it could be applied to FX data. One also have to find a measure of market return to see if the algorithm is able to create higher return using HMMs. Therefore the main tasks is the following:

1. Put together an index which reflects the market return.

2. Try to find an algorithm using HMMs that exceeds the return given by the created index.

3. Examine if the algorithm is stable enough to be used as a tool for algorithmic trading on foreign exchange data.

If these three steps are gone through in a stringent manner one can see our purpose as fulfilled. The return, addressed in step one and two, is not only the rate of return itself. It will be calculated together with the risk of each strategy as the return-to-risk ratio, also called Sharpe ratio described in detail in 3.8.2. The created index will be compared with well known market indices to validate its role as a comparable index for market return. To evaluate the models stability back-testing will be used. The models performance will be simulated using historical data for a fixed time period. The chosen data and time period is described in section 3.1.

1.4.3 Delimitations

The focus has during the investigation been set to a single currency cross, namely the EURUSD. For the chosen cross we have used one set of features given to us from Nordea Quantitative Research as supportive time series. The back testing period for the created models, the comparative index as well as the HMM, has been limited to the period for which adequate time series has been given, for both the chosen currency cross and the given features. Finally the trading strategy will be implemented to an extent containing step one and two in figure 1.2. It will not deal with the final trading decision or the execution of the trades.

1.4.4 Academic Contribution

This thesis is written from a financial engineering point of view, combining areas such as computer science, mathematical science and finance. The main academic contributions of the thesis are the following:

- The master's thesis is one of few applications using hidden Markov models on time dependent sequences of data, such as financial time series.
- It is the first publicly available application of hidden Markov models on foreign exchange data.

1.4.5 Disposal

The rest of this master's thesis will be organized as follows. In chapter 2, the theoretical framework will be reviewed in detail to give the reader a clear view of the theories used when evaluating HMM as a tool for AT. This chapter contains information about different trading strategies used on FX today as well as the theory of hidden Markov models, Gaussian mixture models (GMM), an exponentially weighted expectation maximization (EWEM) algorithm and Monte Carlo (MC) simulation.

In chapter 3 the developed model is described in detail to clear out how the theories has been used in practice to create algorithms based on the framework of HMMs. The created market index will also be reviewed, described shortly to give a comparable measure of market return.

Chapter 4 contains the results of the tests made for the developed models. Trajectories will be presented and commented using different features and settings to see how the parameters affect the overall performance of the model. The models performance will be tested and commented using statistical tests and well known measurements.

The analysis of the results are carried out in chapter 5, on the base of different findings made throughout the tests presented in chapter 4. The purpose of the analysis is to clear out the underlying background to the results one can see, both positive and negative.

Finally, chapter 6 concludes our evaluation of the different models and the framework of HMMs as a tool for AT on FX. Its purpose is to go through the three steps addressed in section 1.4.2 to finally answer the purpose of the master's thesis. This chapter will also point out the most adequate development needed to improve the models further.

Chapter 2

Theoretical Framework

This chapter will give the reader the theoretical basis needed to understand HMMs, i.e. the model to be evaluated in this master's thesis. Different variants, like the one making use of GMMs, and improvements to the HMM, that for example takes time attenuation in consideration, will also be presented. The chapter however starts with some background theories regarding market indices and FX benchmarks, which will constitute the theoretical ground for the comparative index.

2.1 Foreign Exchange Indices

Throughout the years many different trading strategies have been developed to capture return from the FX market. As one finds in any asset class, the foreign exchange world contains a broad variety of distinct styles and trading strategies. For other asset classes it is easy to find consistent benchmarks, such as indices like Standard and Poor's 500 (S&P 500) for equity, Lehmans Global Aggregate Index (LGAI) for bonds and Goldman Sachs Commodity Index (GSCI) for commodities. It is harder to find a comparable index for currencies.

When viewed as a set of trading rules, the accepted benchmarks of other asset classes indicate a level of subjectivity that would not otherwise be apparent. In fact, they really reflect a set of transparent trading rules of a given market. By being widely followed, they become benchmarks. By looking at benchmarks from this perspective there is no reason why there should not exist an applicable benchmark for currencies. [6]

The basic criteria for establishing a currency benchmark is to find approaches that are widely known and followed to capture currency return on the global FX market. In march 2007 Deutsche Bank unveiled their new currency benchmark, The Deutsche Bank Currency Return (DBCR) Index. Their index contains a mixture of three strategies, namely Carry, Momentum and Valuation. These are commonly accepted indices that also other large banks around the world make use of.

Carry is a strategy in which an investor sells a certain currency with a relatively low interest rate and uses the funds to purchase a different currency yielding a higher interest rate. A trader using this strategy attempts to capture the difference between the rates, which can often be substantial, depending on the amount of leverage the investor chooses to use.

To explain this strategy in more detail an example of a ”JPY carry trade” is here presented. Lets say a trader borrows 1 000 000 JPY from a Japanese bank, converts the funds into USD and buys a bond for the equivalent amount. Lets also assume that the bond pays 5.0 percent and the Japanese interest rate is set to 1.5 percent. The trader stands to make a profit of 3.5 percent (5.0 - 1.5 percent), as long as the exchange rate between the countries does not change. Many professional traders use this trade because the gains can become very large when leverage is taken into consideration. If the trader in the example uses a common leverage factor of 10:1, then he can stand to make a profit of 35 percent.

The big risk in a carry trade is the uncertainty of exchange rates. Using the example above, if the USD were to fall in value relative to the JPY, then the trader would run the risk of losing money. Also, these transactions are generally done with a lot of leverage, so a small movement in exchange rates can result in big losses unless hedged appropriately. Therefore it is important also to consider the expected movements in the currency as well as the interest rate for the selected currencies.

Another commonly used trading strategy is *Momentum*, which is based on the appearance of trends in the currency markets. Currencies appear to trend over time, which suggests that using past prices may be informative to investing in currencies. This is due to the existence of irrational traders, the possibility that prices provide information about non-fundamental currency determinants or that prices may adjust slowly to new information. To see if a currency has a positive or negative trend one have to calculate a moving average for a specific historical time frame. If the currency has a higher return during the most recent moving average it said to have a positive trend and vice versa. [6]

The last trading strategy, described by Deutsche Bank, is *Valuation*. This strategy is purely based on the fundamental price of the currency, calculated using Purchasing Power Parity (PPP). A purchasing power parity exchange rate equalizes the purchasing power of different currencies in their home countries for a given basket of goods. If for example a basket of goods costs 125 USD in US and a corresponding basket in Europe costs 100 EUR, the fair value of the exchange rate would be 1.25 EURUSD meaning that people in US and Europe have the same purchasing power. This is why it is believed that the currencies in the long run tend to revert towards their fair value based on PPP. But in short- to medium-run they might deviate somewhat from this equilibrium due to trade, information and other costs. These movements allows for profiting by buying undervalued currencies and selling overvalued. [6]

2.1.1 Alphas and Betas

The described strategies are often referred to as beta strategies, reflecting market return. Beside these there is a lot of other strategies, trying to find the alpha in the market. Alpha is a way of describing excess return, captured by a specific fund or individual trader. Alpha is defined using Capital Asset Pricing Model (CAPM). CAPM for a portfolio is

$$r_p = r_f + \beta_p(r_M - r_f)$$

where r_f is the risk free rate, $\beta_p = \frac{\rho_{pM}\sigma_p\sigma_M}{\sigma_M^2}$ the volatility of the portfolio relative some index explaining market return, and $(r_M - r_f)$ the market risk premia. Alpha can now be described as the excess return, comparing to CAPM, as follows

$$\alpha = r_p^* - r_p = r_p^* - (r_f + \beta_p(r_m - r_f))$$

where r_p^* is the actual return given by the asset and r_p the return given by CAPM.

The above description of alpha is valid for more or less all financial assets. But when it comes to FX, it might sometimes be difficult to assign a particular risk free rate to the portfolio. One suggestion is simply to use the rates of the country from where the portfolio is being managed, but the most reoccurring method is to leave out the risk free rate. This gives

$$\alpha = r_p^* - r_p = r_p^* - \beta_p r_m$$

where r_p^* is adjusted for various transaction costs, where the most common is the cost related to the spread.

2.2 Hidden Markov Models

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. [17] The term HMM is more familiar in the speech recognition community and communication systems, but has during the last years gained acceptance in finance as well as economics and management science. [16]

The theory of HMM deals with two things: estimation and control. The first include signal filtering, model parameter identification, state estimation, signal smoothing, and signal prediction. The latter refers to selecting actions which effect the signal-generating system in such a way as to achieve ceratin control objectives. Essentially, the goal is to develop optimal estimation algorithms for HMMs to filter out the random noise in the best possible way. The use of HMMs is also motivated by empirical studies that favors Markov-switching models when dealing with macroeconomic variables. This provides flexibility to financial models and incorporates stochastic volatility in a simple way. Early works, proposing to have an unobserved regime following a Markov process, where the shifts in regimes could be compared to business cycles, stock prices, foreign exchange, interest rates and option valuation. The motive for a regime-switching model is that the market may switch from time to time, between for example periods of high and low volatility.

The major part of this section is mainly gathered from [17] if nothing else is stated. This article is often used as a main reference by other authors due to its thorough description of HMMs in general.

2.2.1 Hidden Markov Models used in Finance

Previous applications in the field of finance where HMMs have been used range all the way from pricing of options and variance swaps and valuation of life insurances policies to interest rate theory and early warning systems for currency crises. [16] In [14] the author uses hidden Markov models when pricing bonds through considering a diffusion model for the short rate. The drift and the diffusion parameters are here modulated by an underlying hidden Markov process. In this way could the value of the short rate successfully be predicted for the next time period.

HMMs has also, with great success, been used on its own or in combination with e.g. GMMs or artificial neural networks for prediction of financial time series, as equity indices such as the S&P 500. [18, 9] In these cases the authors has predicted the rate of return for the indices during the next time step and thereby been able to create an accurate trading signal.

The wide range of applications together with the proven functionality, in both finance and other communities such as speak recognition, and the flexible underlying mathematical model is clearly appealing.

2.2.2 Bayes Theorem

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

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

There is also a version of Bayes' theorem for continuous distributions. It is somewhat harder to derive, since probability densities, strictly speaking, are not probabilities, so Bayes' theorem has to be established by a limit process. Bayes' theorem for probability density functions is formally similar to the theorem for probabilities:

$$f(x|y) = \frac{f(x,y)}{f(y)} = \frac{f(y|x)f(x)}{f(y)}$$

and there is an analogous statement of the law of total probability:

$$f(x|y) = \frac{f(y|x)f(x)}{\int_{-\infty}^{\infty} f(y|x)f(x)dx}.$$

The notation have here been somewhat abused, using f for each one of these terms, although each one is really a different function; the functions are distinguished by the names of their arguments.

2.2.3 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 2.1 there is a simple example of how this could work.

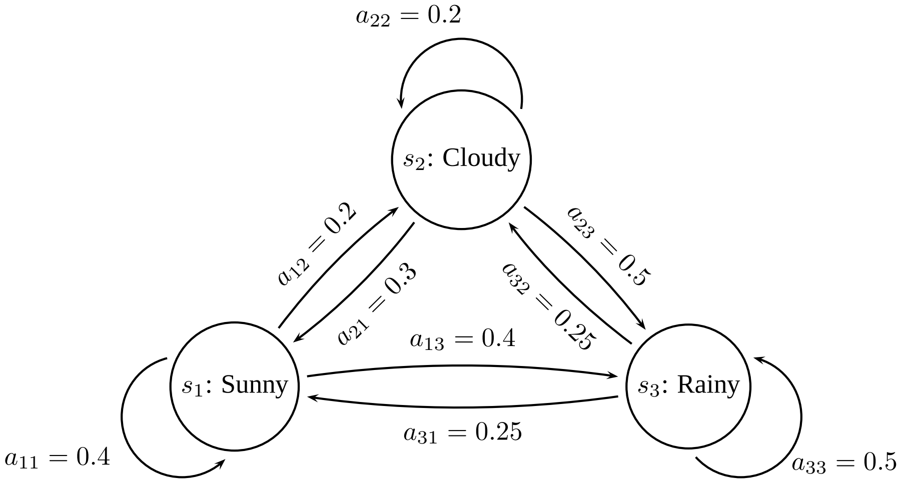


Figure 2.1: A simple example of a Markov chain explaining the weather. a_{ij}, found on the arcs going between the nodes, represents the probability of going from state s_i to state s_j.

In the 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 2.1. 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, as mentioned above, by:

$S = \{s_1, s_2, ..., s_N\}$ a set of N states,

$A = \{a_{11}, a_{12}, ..., a_{NN}\}$ a transition probability matrix A, where each a_{ij} represents the probability of moving from state i to state j, with $\sum_{j=1}^{N} a_{ij} = 1, \forall i$,

$\Pi = \{\pi_1, \pi_2, ..., \pi_N\}$ 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 $t = 1, 2, ...$ 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:

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

Markov chains, where the probability of moving between any two states are non-zero, 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.

The Markov Property Applied in Finance

Stock prices are often assumed to follow a Markov process. This means that the present value is all that is needed for predicting the future, and that the past history and the taken path to today's value is irrelevant. Considering that equity and currency are both financial assets, traded under more or less the same conditions, it would not seem farfetched assuming that currency prices also follows a Markov process.

The Markov property of stock prices is consistent with the weak form of market efficiency, which states that the present price contains all information contained in historical prices. This implies that using technical analysis on historical data would not generate an above-average return, and the strongest argument for weak-form market efficiency is the competition on the market. Given the many investors following the market development, any opportunities rising would immediately be exploited and eliminated. But the discussion about market efficiency is heavily debated and the view presented here is seen somewhat from an academic viewpoint. [12]

2.2.4 Extending the Markov Chain to a Hidden Markov Model

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.

Extending the definition of Markov chains presented in the previous section gives:

$S = \{s_1, s_2, ..., s_N\}$ a set of N *hidden states*,

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

$O = \{o_1, o_2, ..., o_T\}$ an *observation sequence* consisting of T observations, taking values from the discrete alphabet $V = \{v_1, v_2, ..., v_M\}$,

$A = \{a_{11}, a_{12}, ..., a_{NN}\}$ 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,

$\Pi = \{\pi_1, \pi_2, ..., \pi_N\}$ an *initial probability distribution*, where π_i indicates the probability of starting in state s_i. Also, $\sum_{i=1}^{N} \pi_i = 1$.

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

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

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

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:

$$P(o_t|q_t, q_{t-1}, ..., q_1, o_{t-1}, ..., o_1) = P(o_t|q_t)$$

To clarify what is meant by all this, a simple example based on one originally given by [13] is here presented. Imagine that you are a climatologist in the year 2799 and want to study how the weather was in 2007 in a certain region. Unfortunately you do not have any records for this particular region and time, but what you do have is the diary of a young man for 2007, that tells you how many ice creams he had every day.

Stated more formal, you 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 2.2.

2.2.5 Three Fundamental Problems

The structure of the HMM should now be clear, which leads to the question: in what way can HMM be helpful? [17] suggests in his paper that HMM should be characterized by three fundamental problems:

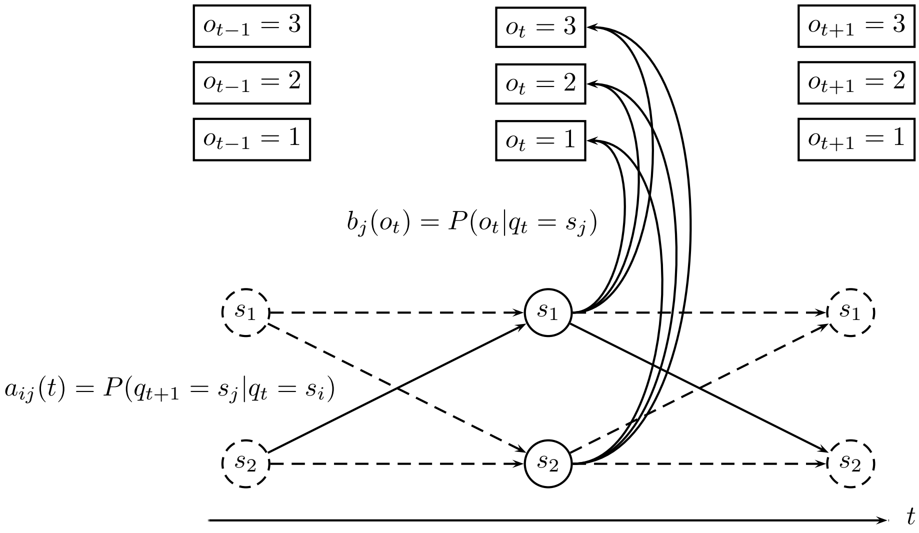


Figure 2.2: A HMM example, where the most probable state path (s_2, s_1, s_2) is outlined. a_{ij} states the probability of going from state s_i to state s_j, and $b_j(o_t)$ the probability of a specific observation at time t, given state s_j.

Problem 1 - Computing likelihood: Given the complete parameter set λ and an observation sequence O, determine the likelihood $P(O|\lambda)$.

Problem 2 - Decoding: Given the complete parameter set λ and an observation sequence O, determine the best hidden sequence Q.

Problem 3 - Learning: Given an observation sequence O and the set of states in the HMM, learn the HMM λ.

In the following three subsections these problems will be gone through thoroughly and how they can be solved. This to give the reader a clear view of the underlying calculation techniques that constitutes the base of the evaluated mathematical framework. It should also describe in what way the framework of HMMs can be used for parameter estimation in the standard case.

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, $O = \{o_1, o_2, ..., o_T\}$, 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

$$Q = \{q_1, q_2, \ldots, q_T\} \tag{2.1}$$

where q_1 is the initial state. The probability of observation sequence O given a state sequence such as 2.1 can be calculated as

$$P(O|Q,\lambda) = \prod_{t=1}^{T} P(o_t|q_t,\lambda) \tag{2.2}$$

where the different observations are assumed to be independent. The property of independence makes it possible to calculate equation 2.2 as

$$P(O|Q,\lambda) = b_{q_1}(o_1) \cdot b_{q_2}(o_2) \cdot \ldots \cdot b_{q_T}(o_T). \tag{2.3}$$

The probability of such a sequence can be written as

$$P(Q|\lambda) = \pi_{q_1} \cdot a_{q_1q_2}a_{q_2q_3} \cdot \ldots \cdot a_{q_{T-1}q_T}. \tag{2.4}$$

The joint probability of O and Q, the probability that O an Q occurs simultaneously, is simply the product of 2.3 and 2.4 as

$$P(O,Q|\lambda) = P(O|Q,\lambda)P(Q|\lambda). \tag{2.5}$$

Finally the probability of O given the model λ is calculated by summing the right hand side of equation 2.5 over all possible state sequences Q

$$\begin{aligned} P(O|\lambda) = \textstyle\sum_{q_1,q_2,\ldots,q_T} P(O|Q,\lambda)P(Q|\lambda) = \\ = \textstyle\sum_{q_1,q_2,\ldots,q_T} \pi_{q_1}b_{q_1}(o_1)a_{q_1q_2}b_{q_2}(o_2)\ldots a_{q_{T-1}q_T}b_{q_T}(o_T). \end{aligned} \tag{2.6}$$

Equation 2.6 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 $a_{q_1q_2}$, 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. Therefor 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*.[1] For initiation one need to define the forward variable as

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

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

1. Initialization:

$$\alpha_1(i) = \pi_i b_i(o_1), \quad 1 \le i \le N. \tag{2.7}$$

2. Induction:

$$\begin{aligned} \alpha_{t+1}(j) = \left[\textstyle\sum_{j=1}^{N} \alpha_t(i)a_{ij}\right] b_j(o_{t+1}), \quad 1 \le t \le T-1, \\ 1 \le j \le N. \end{aligned} \tag{2.8}$$

[1] The backward part of the calculation is not needed to solve Problem 1. It will be introduced when solving Problem 3 later on.

3. Termination:

$$P(O|\lambda) = \sum_{i=1}^{N} \alpha_T(i). \tag{2.9}$$

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 2.3.

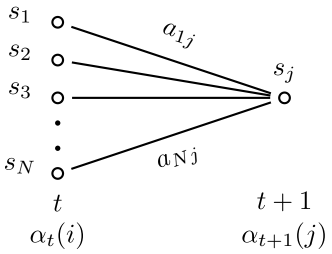


Figure 2.3: Illustration of the sequence of operations required for the computation of the forward variable $\alpha_t(i)$.

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 $s_i, 1 \leq i \leq N$ 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 2.8 is performed for all states $s_j, 1 \leq j \leq N$, for a given time t and iterated for all $t = 1, 2, \ldots, T-1$. Step 3 then gives $P(O|\lambda)$ by summing the terminal forward variables $\alpha_T(i)$. This is the case because, by definition

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

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:

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

$\beta_t(i)$ 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:

$$\beta_T(i) = 1, \quad 1 \leq i \leq N.$$

2. Induction:

$$\beta_t(i) = \textstyle\sum_{j=1}^{N} a_{ij} b_j(o_{t+1})\beta_{t+1}(j), \quad \begin{array}{l} t = T-1, T-2, \ldots, 1, \\ 1 \leq i \leq N. \end{array}$$

Step 1 defines $\beta_T(i)$ to be one for all s_i. Step 2, which is illustrated in figure 2.4, shows that in order to 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.

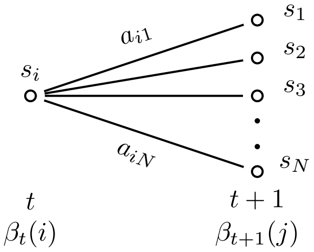


Figure 2.4: Illustration of the sequence of operations required for the computation of the backward variable $\beta_t(i)$.

As mentioned before the backward variable is not used to find the probability $P(O|\lambda)$. Later on 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.

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 a 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

$$\gamma_t(i) = P(q_t = s_i|O, \lambda) \tag{2.10}$$

which gives the probability of being in state s_i at time t given the observation sequence, O, and the model, λ. Equation 2.10 can be expressed simply using the forward and backward variables, $\alpha_t(i)$ and $\beta_t(i)$ as follows:

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{P(O|\lambda)} = \frac{\alpha_t(i)\beta_t(i)}{\sum_{i=1}^{N}\alpha_t(i)\beta_t(i)}. \tag{2.11}$$

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 $o_1, o_2, \ldots, o_t$ and the state s_i at t, while $\beta_t(i)$

accounts for the remainder of the observation sequence $o_{t+1}, o_{t+2}, \ldots, o_T$ 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:

$$q_t = \text{argmax}_{1 \leq i \leq N}[\gamma_t(i)], \quad 1 \leq t \leq T. \tag{2.12}$$

Although equation 2.12 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 2.12 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 $Q = \{q_1, q_2, \ldots, q_T\}$ for a given observation sequence $O = \{o_1, o_2, \ldots, o_T\}$, one need to define the quantity

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

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:

$$\delta_{t+1}(j) = [\max_i \delta_t(i)a_{ij}]b_j(o_{t+1}). \tag{2.13}$$

To be able to retrieve the state sequence, one need to keep track of the argument which maximized 2.13, 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:

1. Initialization:

$$\begin{aligned} &\delta_1(i) = \pi_i b_i(o_1),\ 1 \leq i \leq N \\ &\psi_1(i) = 0. \end{aligned}$$

2. Recursion:

$$\begin{aligned} \delta_t(j) &= \max\nolimits_{1 \leq i \leq N}[\delta_{t-1}(i)a_{ij}]b_j(o_t), & 2 \leq t \leq T \\ & & 1 \leq j \leq N \\ \psi_t(j) &= \text{argmax}_{1 \leq i \leq N}[\delta_{t-1}(i)a_{ij}], & 2 \leq t \leq T \\ & & 1 \leq j \leq N \end{aligned} \tag{2.14}$$

3. Termination:

$$\begin{aligned} P^* &= \max\nolimits_{1 \leq i \leq N}[\delta_T(i)] \\ q_T^* &= \text{argmax}_{1 \leq i \leq N}[\delta_T(i)]. \end{aligned} \tag{2.15}$$

4. Path (state sequence) backtracking:

$$q_t^* = \psi_{t+1}(q_{t+1}^*), \quad t = T-1, T-2, \ldots, 1.$$

It is important to note that the Viterbi algorithm is similar in implementation to the forward calculation in 2.7 - 2.9. The major difference is the maximization in 2.14 over the previous states which is used in place of the summing procedure in 2.8. It should also be clear that a lattice structure efficiently implements the computation of the Viterbi procedure.

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. Options to this is to use a form of Maximum A Posteriori (MAP) method, explained in [7], or setting up the problem as an optimization problem, which can be solved with for example gradient technique, which can yield solutions comparable to those of the aforementioned method. But 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 FX 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 2.5.

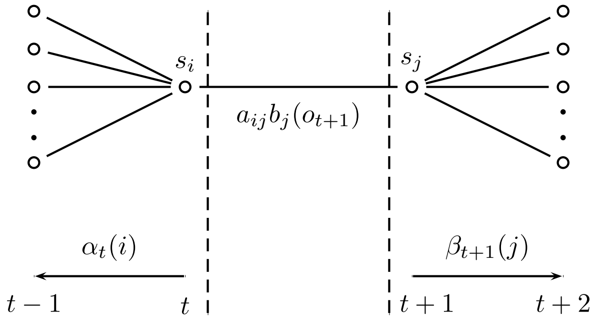


Figure 2.5: Illustration of the sequence of operations required for the computation of the joint event that the system is in state s_i at time t and state s_j at time $t+1$.

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:

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

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 in 2.11, $\gamma_t(i)$ 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:

$$\begin{aligned}\overline{\pi}_i &= \gamma_1(i), & (2.16)\\ \overline{a}_{ij} &= \frac{\sum_{t=1}^{T-1}\xi_t(i,j)}{\sum_{t=1}^{T-1}\gamma_t(i)}, & (2.17)\\ \overline{b}_j(v_k) &= \frac{\sum^{T}_{\substack{t=1\\ s.t.\ o_t=v_k}}\gamma_t(j)}{\sum_{t=1}^{T}\gamma_t(j)}. & (2.18)\end{aligned}$$

One should see that equation 2.16 can be interpreted as the frequency in state s_i at time $t = 1$. Equation 2.17 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. And finally, equation 2.18 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.

If the current HMM is defined as $\lambda = \{A, B, \Pi\}$ and used to compute the right hand side of 2.16 – 2.18, and at the same time define the re-estimation HMM as $\overline{\lambda} = (\overline{A}, \overline{B}, \overline{\Pi})$ as determined from the left hand side of 2.16 – 2.18 it has been proven that either

1. the initial model λ defines a critical point of the likelihood function, in which case $\overline{\lambda} = \lambda$ or
2. model $\overline{\lambda}$ is more likely than model λ in the sense that $P(O|\overline{\lambda}) > P(O|\lambda)$, which means that one have found a new model $\overline{\lambda}$ from which the observation sequence is more likely to have been produced.

An iterative re-estimation process, replacing λ with $\overline{\lambda}$ 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. One problem, earlier discussed is that the forward-backward algorithm only leads to a local maximum. For most applications the optimization surface is very complex and has many local maxima.

The re-estimation formulas of 2.16 - 2.18 can be derived directly from Baum´s auxiliary function

$$Q(\lambda, \overline{\lambda}) = \sum_{Q} P(Q|O, \lambda) \log \left[P(O, Q|\overline{\lambda})\right]$$

by maximizing over $\overline{\lambda}$. It has been proven that maximization of $Q(\lambda, \overline{\lambda})$ leads to an increasing likelihood as follows:

$$\max_{\overline{\lambda}} [Q(\lambda, \overline{\lambda})] \Rightarrow P(O|\overline{\lambda}) \geq P(O|\lambda).$$

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

$$\begin{aligned} \sum_{i=1}^{N} \overline{\pi}_i &= 1, \\ \sum_{j=1}^{N} \overline{a}_{ij} &= 1,\ 1 \leq i \leq N, \\ \sum_{k=1}^{M} \overline{b}_j(v_k) &= 1,\ 1 \leq j \leq N, \end{aligned}$$

are automatically satisfied at each iteration. By looking at the parameter estimation problem as a constrained optimization of $P(O|\lambda)$ the techniques of Langrange multipliers can be used to find the best value of the adequate parameters. If setting up a Lagrange optimization using multipliers, it can be shown that P is maximized when the following set of conditions are met:

$$\pi_i = \frac{\pi_i \frac{\partial P}{\partial \pi_i}}{\sum_{k=1}^{N} \pi_k \frac{\partial P}{\partial \pi_k}}, \tag{2.19}$$

$$a_{ij} = \frac{a_{ij} \frac{\partial P}{\partial \pi_i}}{\sum_{k=1}^{N} \pi_k \frac{\partial P}{\partial \pi_k}}, \tag{2.20}$$

$$b_j(v_k) = \frac{b_j(v_k) \frac{\partial P}{\partial b_j(v_k)}}{\sum_{l=1}^{M} b_j(v_l) \frac{\partial P}{\partial b_j(v_l)}}. \tag{2.21}$$

By rewriting 2.19 – 2.21, the right hand side of every equation can be converted to be identical to the right hand sides of each part of 2.16 – 2.18. This is showing that the re-estimation formulas are indeed exactly correct at critical points of $P(O|\lambda)$.

This concludes the basics about what HMMs can be used for. In the next chapter one will be able to see how this can be applied for forecasting movements in a currency cross.

2.2.6 Multivariate Data and Continuous Emission Probabilities

When in previous sections addressing the observations, o_t, it has been more or less assumed that these take on scalar values. But sometimes they contain more than one element, which means that data is generated from several distinct underlying causes with an unknown relationship between them, where each cause contributes independently to the process. But it is however not possible to see how much and in what way they affect the final result. The observation sequence thus looks like:

$$O = \{\overline{o}_1, \ldots, \overline{o}_T\}$$

where each observation $\overline{o_t}$ is a vector. The overline is henceforward left out, since there should be no confusion in the continuing as to what o_t is (scalar or multivariate).

It has up until now also been assumed that the emission probabilities are discrete:

$$b_j(o_t) = P(o_t|q_t = s_j)$$

. This means that there is a finite number of possible observations at each time instant t. But this is not always the case, the rates of return in financial time series can for example assume any value on a continuous scale. One possible solution is to use vector quantization, creating a codebook, thus allowing the observation sequence O to assume values from the discrete alphabet V. One could for example use different thresholds for the returns and then sort them according to *large drop*, *small drop*, *no change*, etc.

The risk of loosing information using a discrete alphabet when dealing with continuous data due to poorly performed discretizing can be a problem. It is however fully possible to use continuous emission probabilities. Each emission probability $b_j(o_t)$ is then a D-dimensional log-concave or elliptically symmetric density. One of the more commonly used densities is the Gaussian density, or Normal distribution, which for the single-variate case gives

$$b_j(o_t) \sim \mathcal{N}(o_t, \mu_j, \sigma_j) = \frac{1}{\sqrt{2\pi}\sigma_j} e^{-\frac{(o_t-\mu_j)^2}{2\sigma_j^2}}$$

and for the multivariate

$$b_j(o_t) \sim \mathcal{N}(o_t, \mu_j, \Sigma_j) = \frac{1}{(2\pi)^{D/2}|\Sigma_j|^{1/2}} e^{-\frac{1}{2}(o_t-\mu_j)^T\Sigma_j^{-1}(o_t-\mu_j)}$$

where j denominate the state. The variables μ and Σ (similar for σ) are calculated as

$$\begin{aligned}\mu_j &= \frac{\sum_{t=1}^{T}\gamma_t(j)o_t}{\sum_{t=1}^{T}\gamma_t(j)} \\ \Sigma_j &= \frac{\sum_{t=1}^{T}\gamma_t(j)(o_t-\mu_j)(o_t-\mu_j)^T}{\sum_{t=1}^{T}\gamma_t(j)}\end{aligned}$$

where $\gamma_t(j)$ is the probability of being in state s_j at time t, as stated in equation 2.10.

2.3 Gaussian Mixture Models

As just described in the last section, HMMs can take advantage of continuous emission probabilities, e.g. Gaussian. But what if the underlying mechanism is non-normal?

In [15], statistic tests on a EURUSD returns series shows that it is non-normal at a 99 percent confidence level, containing both skewness and kurtosis. This would be an argument for using not one, but several, Gaussians to describe the process. And this is where the Gaussian mixture model comes in handy. GMMs are widely used as statistical models, being enduring and well-weathered models of applied statistics. [1] Sample data are thought of as originating from various sources where the data from each source is modeled as a Gaussian. The goal is to find the generating Gaussians, that is their mean and covariances, and the ratio (the mixing weights) in which each source is present.

Another argument for applying the GMM is that financial returns often behave differently in normal situations and during crisis times, and in such a situation one can assume an underlying mechanism so that each observation belongs to one of the number of different sources or categories. In this form of mixture, each of the sources is described by a component probability density function, and its mixture weight is the probability that an observation comes from this component. [15]

For a more mathematical definition: suppose that the discrete random variable X is a mixture of M discrete random variables Y_i. Then the probability mass function of X, $f_X(x)$, is a weighted sum of its component distributions:

$$\begin{aligned} & f_X(x) = \textstyle\sum_{m=1}^{M} w_m f_{Y_m}(x) \\ & \textstyle\sum_{m=1}^{M} w_m = 1,\, w_m \geq 0\, m = 1, \ldots, M \end{aligned} \tag{2.22}$$

In figure 2.6 three Gaussians probability density functions are plotted together with the resulting mixture density function. The Gaussians constituting the mixture are referred to as the mixture components.

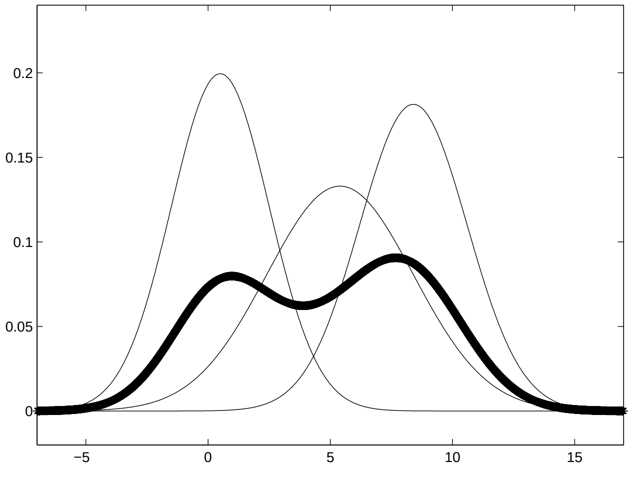


Figure 2.6: A mixture of three Gaussians equally weighted ($w_i = 1/3, i = 1, 2, 3$).

2.3.1 Possibilities to More Advanced Trading Strategies

Using GMM will supply additional information about the returns to be forecasted. In the following two subsections some opportunities that emerge and from which one can draw advantages will be gone trough.

Filtering Trades

When trading with foreign exchange one have to take the spread between the bid and ask rate in consideration. On the EURUSD rate, the spread is about 3 pips[2] and with an average exchange rate of 0.8971 this gives an average cost of 0.033 percent per position. [15] It could therefore be interesting sorting out those trading decisions that returns below 0.033 percent. Since the GMM returns a probability density function, it is possible to derive more information than if the exchange rate is to go up or down. It is for example possible to say: ”a trade will only be executed if the expected return exceeds a threshold d and the probability of this is greater than x percent”.

[15] finds in their report that applying this filter results in a lower number of trades, which might not be surprising. The accumulated return over the test period is however below that of the non-filter trading strategy.

Using Leverage

One way of gaining a higher profit despite fewer trades, following the filtering strategy described above, would be to use leverage. By testing with different thresholds d, while applying a leverage factor that ensures a volatility equal to that of the non-leverage strategy, it is possible to find a combination that yields higher returns than in the case without leverage. In [15] the use of leverage shows an increasing profit, with a leverage factor around 45%. But the risk of large losses has also increased — with to few trades, the losses far exceeds those of the other models in the peer group. It is evidently very important to use leverage with care.

2.3.2 The Expectation Maximization Algorithm on Gaussian Mixtures

When applying the Gaussian mixtures to HMM every state is associated with one mixture, see figure 2.7. Note that the number of components in a mixture is allowed to vary between different states. Applying equation 2.22 to the case with HMM, where $m = 1, \ldots, M$ is the mixture component of state s_j, $j = 1, \ldots, N$, one get:

$$b_j(o_t) = \textstyle\sum_{m=1}^{M} w_{jm} b_{jm}(o_t)$$

$$\textstyle\sum_{m=1}^{M} w_{jm} = 1\,\forall j,\; w_{jm} \geq 0\,\forall j, m$$

The task now lies in finding the set of mixture parameters that optimizes the log-likelihood of the model. This is done by using the EM algorithm described in previous sections, and for this it should be recalled that for a state s_j at a time t one have $b_j(o_t) \sim \mathcal{N}(o_t, \mu_j, \sigma_j)$ for the single observation sequence and $b_j(o_t) \sim \mathcal{N}(o_t, \mu_j, \Sigma_j)$ for the multivariate observation sequence. This gives:

[2] 1 pip = 0.0001

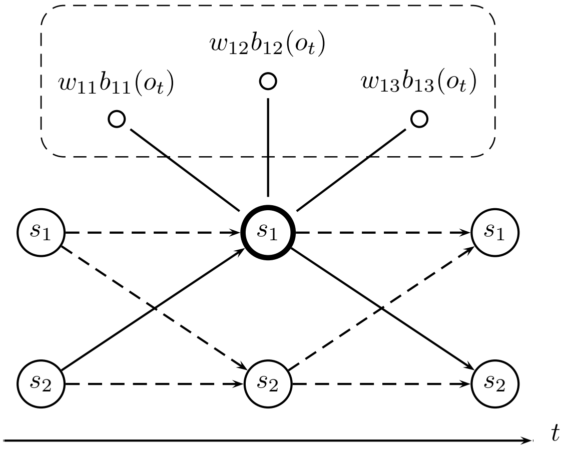


Figure 2.7: A state sequence with a mixture, consisting of three components, for state s_1 at time t.

$$b_{jm}(o_t) \sim \mathcal{N}(o_t, \mu_{jm}, \sigma_{jm}) = \frac{1}{\sqrt{2\pi}\sigma_{jm}} e^{-\frac{(o_t-\mu_{jm})^2}{2\sigma_{jm}^2}},$$

where o_t is a scalar and μ_{jm} and σ_{jm} are the mean and standard deviation for state s_j and mixture component m. For a multivariate observation sequence one get:

$$b_{jm}(o_t) \sim \mathcal{N}(o_t, \mu_{jm}, \Sigma_{jm}) = \frac{1}{(2\pi)^{D/2}|\Sigma_{jm}|^{1/2}} e^{-\frac{1}{2}(o_t-\mu_{jm})^T \Sigma_{jm}^{-1}(o_t-\mu_{jm})},$$

where o_t is a vector and μ_{jm} and Σ_{im} are the mean and covariance matrix for state s_j and mixture component m.

The learning of the model parameters thus becomes an estimation of means and covariances for the different Gaussians. The variable $\gamma_t(j,m)$ denotes the probability of being in state s_j at time t with the mth mixture component accounting for the observation o_t. It is defined according [17]:

$$\gamma_t(j,m) = \left[\frac{\alpha_t(j)\beta_t(j)}{\sum_{j=1}^{N}\alpha_t(j)\beta_t(j)}\right]\left[\frac{w_{jm}\mathcal{N}(o_t, \mu_{jm}, \Sigma_{jm})}{\sum_{k=1}^{M} w_{jk}\mathcal{N}(o_t, \mu_{jk}, \Sigma_{jk})}\right]$$

and this in its turn give [17]:

$$\begin{aligned}
\mu_{jm} &= \frac{\sum_{t=1}^{T}\gamma_t(j,m)o_t}{\sum_{t=1}^{T}\gamma_t(j,m)} \\
\Sigma_{jm} &= \frac{\sum_{t=1}^{T}\gamma_t(j,m)(o_t-\mu_{jm})(o_t-\mu_{jm})^T}{\sum_{t=1}^{T}\gamma_t(j,m)} \\
w_{jm} &= \frac{\sum_{t=1}^{T}\gamma_t(j,m)}{\sum_{t=1}^{T}\sum_{k=1}^{M}\gamma_t(j,k)}
\end{aligned}$$

whereas a_{ij} and π_i is calculated as before.

2.4 The Exponentially Weighted Expectation Maximization Algorithm

Many studies on the EM algorithm have been carried out, but few have treated the matter of non-stationarity in time series. For most data this is not a major concern and the traditional EM based HMM have had great success in fields like language and video processing. But when it comes to financial time series, it is not farfetched to assume this could cause problems. [18] proposes in his work an EM algorithm that makes use of a Exponentially Weighted Moving Average (EWMA) for time attenuation. This allows for focus on more recent data, but while keeping information about trends and patterns contained in older data in mind. New emerging trends would otherwise be diluted.

For a more thorough explanation on the exponentially weighted expectation maximization (EWEM) algorithm, see [18].

2.4.1 The Expectation Maximization Algorithm Revisited

$P(O|\lambda)$ is the probability of observing the sequence $O = \{o_1, \ldots, o_T\}$ given the parameter set $\lambda = \{A, B, \Pi\}$. Assuming that the data is independent and identically distributed (i.i.d.) with distribution P, the probability can be written as:

$$P(O|\lambda) = \prod_{t=1}^{T} P(o_t|\lambda) = L(\lambda|O)$$

The function L is called the likelihood function of the parameters given the data. The task for the EM algorithm is to find a λ^* that maximizes the value of L:

$$\lambda^* = \text{argmax}_\lambda L(\lambda|O)$$

Since the log function is monotonous, it is often easier maximize over the logarithm of of the likelihood function since it reduces multiplications to additions. It should also be noted that the EM algorithm only leads to a local maximum.

The target data for the EM algorithm can be viewed as composed of two parts, an observed part, $O = \{o_1, \ldots, o_T\}$, and an underlying part, $Q = \{q_1, \ldots, q_T\}$, i.e. the hidden Markov chain. O is referred to as the incomplete data and $Z = (O, Q)$ as the complete data. By the Bayes rule, the joint probability density function for one of the data in the complete data is:

$$P(z_t|\lambda) = P(o_t, q_t|\lambda) = P(q_t|o_t, \lambda)P(o_t|\lambda)$$

This leads us to the complete data likelihood function for the overall data set based on the joint density function:

$$L(\lambda|Z) = L(\lambda|O, Q) = P(O, Q|\lambda)$$

When estimating the hidden data Q, O and λ can be seen as constants and the likelihood for the complete data can be seen as a function only dependent of Q:

$$L(\lambda|O, Q) = h_{O,\lambda}(Q)$$

The EM algorithm now tries to find the expected value, the E-step, of the complete data log-likelihood $\log P(O, Q|\lambda)$, with respect to the unknown data Q, given the observed part O and the parameters from the last estimation λ. A Q function is defined to denote this expectation:

$$Q(\lambda, \overline{\lambda}) = \sum_{Q} P(Q|O, \lambda) \log P(O, Q|\overline{\lambda})$$

where λ is model parameters from previous estimations and $\overline{\lambda}$ is the re-estimated model. In the M-step we maximize the Q function with respect to $\overline{\lambda}$, and it has been proven by Baum and his colleagues that this leads to an increased likelihood:

$$\max_{\overline{\lambda}}[Q(\lambda, \overline{\lambda})] \Rightarrow P(O|\overline{\lambda}) \geq P(O|\lambda)$$

The two steps are repeated until the probability of the overall observation does not improve significantly. This is what is performed in section 2.2.5.

Rewriting the Q function

An assumption when calculating the Q function, is that all data is of equal importance in the parameter estimation. As stated above this is not true for all data, e.g. financial time series. One therefore have to rewrite the Q function if the EWEM is to be used, and this is done by including a time-dependent variable η which gives:

$$\begin{aligned} \hat{Q}(\lambda, \overline{\lambda}) &= E\left[\log \eta P(O, Q|\lambda)|O, \lambda\right] = \\ &= \textstyle\sum_{Q} P(Q|O, \lambda) \log \eta P(O, Q|\overline{\lambda}) \end{aligned}$$

Since the logarithmic function is monotonous and $0 < \eta \leq 1$, η can be taken out of the log function. The Q function then becomes:

$$\begin{aligned} \hat{Q}(\lambda, \overline{\lambda}) &= E\left[\eta \log P(O, Q|\lambda)|O, \lambda\right] = \\ &= \textstyle\sum_{Q} \eta P(Q|O, \lambda) \log P(O, Q|\overline{\lambda}) \end{aligned}$$

The weight η is a vector of real values between 0 and 1: $\eta = \{\eta_1, \ldots, \eta_T\}$, where T is the number of data in the observation sequence. The whole point of this is that the likelihood for different data in the sequence is discounted by a confidence density.

2.4.2 Updating the Algorithm

With the Q function rewritten one can derive new expressions for the EM algorithm. A full derivation will not be shown here, and the interested reader is referred to [18]. So for the case of a HMM using GMM, where j stands for the state and m for the mixture component, one get:

$$\begin{aligned} \mu_{jm} &= \frac{\sum_{t=1}^{T} \eta_t \gamma_t(j, m) o_t}{\sum_{t=1}^{T} \eta_t \gamma_t(j, m)} \\ \Sigma_{jm} &= \frac{\sum_{t=1}^{T} \eta_t \gamma_t(j, m)(o_t - \mu_{jm})(o_t - \mu_{jm})^T}{\sum_{t=1}^{T} \eta_t \gamma_t(j, m)} \end{aligned}$$

$$
\begin{aligned}
w_{jm} &= \frac{\sum_{t=1}^{T} \eta_t \gamma_t(j,m)}{\sum_{t=1}^{T} \sum_{k=1}^{M} \eta_t \gamma_t(j,k)} \\
a_{ij} &= \frac{\sum_{t=1}^{T-1} \eta_t \xi_t(i,j)}{\sum_{t=1}^{T-1} \eta_t \gamma_t(i)}
\end{aligned}
$$

where $\xi_t(i,j) = P(q_t = s_i, q_{t+1} = s_j | O, \lambda)$ and $\gamma_t(i)$ the probability of being in state s_i at time t. π_i is calculated as before.

2.4.3 Choosing η

As mentioned above, the algorithm uses a EWMA for weighting the different observations. EWMAs are frequently used in technical analysis in finance, originally invented to keep track of time changes in different time series. A short-term EWMA responds faster to changes in the market, while a long-term EWMA takes more consideration to long-term trends. When curves with different lengths cross each other, this usually means a trend change.

The formula for calculating EWMAs for $t \geq 2$ is:

$$Y_t = \rho X_t + (1-\rho) Y_{t-1} \tag{2.23}$$

where Y_t and X_t is the EWMA respectively the data at time t, and ρ the smoothing factor. Replacing older Ys according to equation 2.23, one get:

$$Y_t = \rho X_{t-1} + \rho(1-\rho) X_{t-2} + \rho(1-\rho)^2 X_{t-3} + \rho(1-\rho)^3 X_{t-4} + \dots \tag{2.24}$$

If the last time instant for which data is available is denoted with T, one can in equation 2.24 easily see that $\eta = \rho$ for T and for older data at times t, $\eta = \rho(1-\rho)^{T-t}$. η is obviously a function of the time and summarizing these observations, it can be set accordingly:[3]

$$\eta_t = \rho(1-\rho)^{T-t}, \quad \begin{array}{l} t = 1, \dots, T, \\ 0 < \rho \leq 1 \end{array}$$

The smoothing factor can be seen as factor deciding how big the weights to old data should be. ρ can be expressed in percentage, e.g. $\rho = 0.1$ equals 10 percents. But it can also be expressed in terms of k time periods, where $\rho = \frac{2}{k+1}$. For example, $k = 19$ equals as well $\rho = 0.1$. In this master's thesis, the number of time periods will usually be stated since this might be more intuitive when looking at an EWMA. Figure 2.8 shows an example of how the choice of different smoothing factors will affect data.

[3] The setting of η here differs from what is presented in [18].

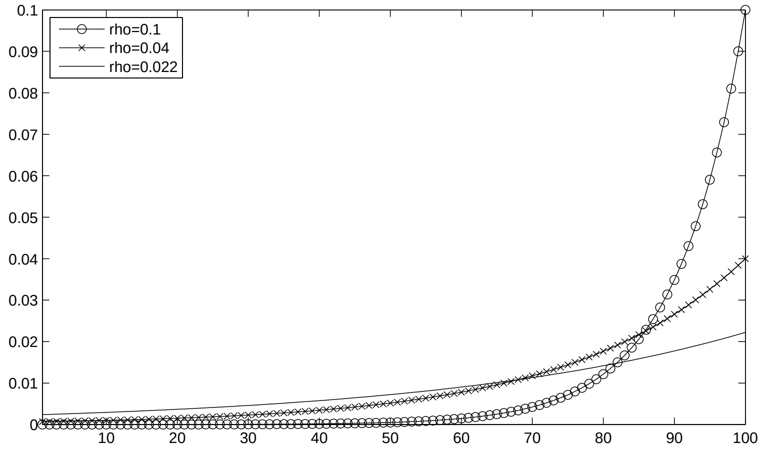
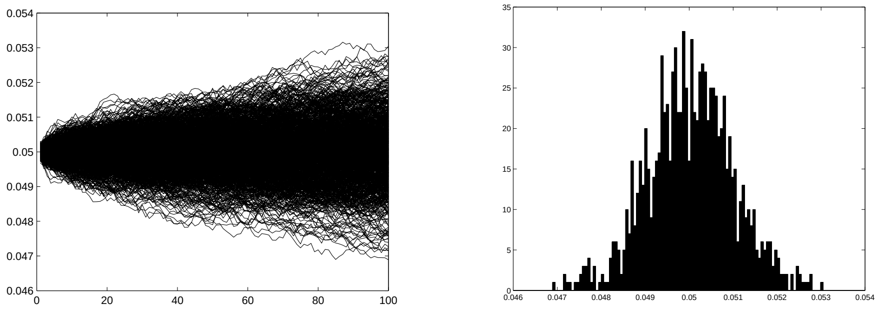


Figure 2.8: The proportion of each of the 100 days used in the estimation procedure. For $\rho = 0.1$ ($k = 19$), the last couple of days is given a lot of attention while older data points hardly have any impact at all in the calculations. Descending to $\rho = 0.022$ ($k = 89$) older data gets less attenuated while days closer in time is given less attention.

2.5 Monte Carlo Simulation

A MC simulation is a procedure for sampling random outcomes of a given stochastic process. It can for example be used when one wants to get hold of the density function for a certain process, as it supplies one with both the mean, as the expected value of the outcomes, and the standard deviation of the underlying distribution. [12] It can also be used if one wants to simulate for example the development for a financial asset under a longer time period. MC simulations tends to be numerically more efficient than other procedures when there are more than three stochastic variables. This is because the time taken to carry out the MC simulation increase approximately linearly with the number of underlying assets.



(a) (b)

Figure 2.9: (a) Monte Carlo simulation using 1000 simulations over 100 time periods. (b) Histogram over the final value for the 1000 simulations.

When simulating the distribution for the currency cross returns, see figure 2.6, a Brownian motion is used for each one of the Gaussians. The price change for a currency cross is thereby given by:

$$dS = \mu_i S dt + \sigma_i S dz$$

where dz is a Wiener process so that $dz = \varepsilon\sqrt{dt}$, $\varepsilon \sim N(0,1)$, and μ_i and σ_i are the mean and standard deviation for each Gaussian mixture, $i = 1,\ldots,M.$ Discretizing the expression and dividing both sides with S, the process for the return is given by:

$$\frac{\Delta S}{S} = \mu_i \Delta t + \sigma_i \varepsilon \sqrt{\Delta t},$$

which can be used to simulate the uncertainty in $E[S_{t+1}]/S_t - 1$.

The accuracy of the result given by the MC simulation depends on the number of trials. The standard error of the estimate is

$$\frac{\sigma_i}{\sqrt{N}}$$

where N is the number of simulations. A 95 percent confidence interval for the rate of return u is therefor given as

$$\mu_i - \frac{1.96\sigma_i}{\sqrt{N}} \leq u \leq \mu_i + \frac{1.96\sigma_i}{\sqrt{N}}.$$

This shows that the uncertainty about the value of the rate of return is inversely proportional to the square root of the number of trials. To double the accuracy of a simulation, one must quadruple the number of trials; to increase the accuracy by a factor 10 one have to increase the number of trials by a factor 100 and so on. [12]

The number 1.96 is given by the the normal distribution, where one for a double sided 95 percent confidence interval have:

$$P(-s \leq S \leq s) = 1 - \alpha = 0.95$$

$$\Phi(s) = P(S \leq s) = 1 - \frac{\alpha}{2} = 0.975$$

$$\Phi^{-1}(\Phi(s)) = \Phi^{-1}(0.975) = 1.96$$

The 95 percent is a commonly used confidence level and will be used through out this master's thesis.

When MC simulation is used, a large number of simulations is needed to obtain good results, i.e. a small σ which gives a narrow confidence interval. To speed up this process and to get even better random generation, different variance reduction techniques are available, and one utilized here is latin hypercube sampling. This method divides the interval $[0, 1]$ into n segments, and in each of these segments one random number is generated from an uniform process. The interval is then inverse transformed to obtain a Gaussian distribution.

2.6 Summary

In this chapter the theoretical foundation of hidden Markov models have been presented together with three fundamental problems, namely computing likelihood, decoding, and learning. They all serve different purposes, depending on what one wants to do, and are often considered the basis of HMM. In the next chapter one will see how they can be utilized for forecasting cross movements on the currency market. It was also shown that the HMM can be applied in different versions, for example taking in multivariate data with continuous emission probabilities.

Extensions to the HMM framework, like Gaussian mixture models, opening up for non-normal distributions which is often assumed for FX data, and an exponentially weighted expectation maximization algorithm, taking time attenuation in consideration, have also been reviewed. This in the hope that they will serve as good complements to the basic framework.

A short introduction to different and commonly used trading strategies, such as carry and momentum, was also made and will be the starting point for the comparative index to be created. Alphas, implying excess return, and betas, indicating market return, was also introduced in this section.

The contents of this chapter should now be fully obtained, as the next chapter, Applying Hidden Markov Models on Foreign Exchange Data, will try to give the reader a clearer picture of how HMM can be used more specific on FX data as well as how the implementation is to be carried out.

Chapter 3

Applying Hidden Markov Models on Foreign Exchange Data

This chapter will explain how HMMs can be used on FX data for one-day predictions, as well as how the implementation is to be carried out. The chapter constitute the core of how one can turn the theory into a functional prediction framework. Data sources, the choices regarding number of states and mixtures as well as window lengths, will be gone through, together with various additions to the model. Different trading signals will also be presented, and how this can affect the results.

3.1 Used Input Data

When applying the HMM framework on financial time series, focus will be on the EURUSD. This is the major currency cross on the global FX market, which guarantees a high liquidity. Two different combinations of input data will be tested; only the EURUSD return and EURUSD returns together with six features, given by Nordea Quantitative Research. Multiple inputs is possible since the HMM framework can handle multivariate data, as seen in section 2.2.6, and it can be interesting to see whether or not they might be of any aid.

These features will not, according to secrecy reasons, be listed explicitly. They will from now on be named $F = \{f_0, f_1, \ldots, f_6\}$, where f_0 is the return on EURUSD. It is important to point out that the six features is calculated from price series as well as for the return for EURUSD and have shown high significance using other mathematical frameworks, like artificial neural networks, for prediction of EURUSD. It should however be noted that these six features not are linearly correlated with the EURUSD return. One reason to use other time series to facilitate the prediction, is that there are sideway movements in the long term trends. It should be possible to smooth these movements using time series with other composition than the one to be predicted.

The size of data set amounts to approximately 1400 daily price quotes from the years 2003-2006, and reallocation will as well take place on a daily basis. The set of data allows for a 1000 days long training period (30/05/2003 – 22/02/2006), a 200 days out-of-sample period (23/02/2006 – 10/09/2006) and an additional amount of data that is needed for the first day's historical time window, thus allowing for a maximum window length of 200 days. The distinction between training and out-of-sample might in this case be somewhat diffuse — the model parameters are re-estimated every day, whether one is in the training or out-of-sample period. This means that the model is constantly adopting and no settings are fixed after the training period. But all pre-testing of the models have been performed during the training period, meaning that choices regarding ”optimal” model settings have been on the basis of the Sharpe ratios and P-values for this time period.

A limited amount of data can be a problem during back-testing, since one usually wants to test a model over long time periods. But as stated in the introduction the huge increase in the total turnover during the last years due to changes in market conditions has most certainly changed the play rules of the FX trading. And since back-testing should be performed on data series reflecting today's market conditions, results based on data from the last couple of years should constitute a valid ground for evaluation.

3.2 Number of States and Mixture Components and Time Window Lengths

The HMM framework is consisting of many different parameters. Two important ones, when using HMM in combination with GMM, is the number of states, N, and the number of Gaussian mixtures, M. As one can see in different articles about the HMM framework there is no analytical way of finding the optimal number of hidden states for a hidden Markov model. [9, 18] One therefore have to find the optimal, in some sense, combination through empirical experiments on historical data. Following this idea, different combinations of hidden states and Gaussian mixtures will be tested to see which one that generates the best forecast of EURUSD. Intuitively, the number of states can be seen as the number of strategies that one have to forecast for the next time step. This number should not be too large, neither too small. Too many states will make little distinction between the different strategies and will increase the need of computer strength. A too small number can on the other hand increase the risk of misclassification. The number of mixture components used in the GMM part of the framework follows the same discussion. The important point is to replicate the skewness and kurtosis of the returns underlying distribution.

Because financial time series differs from other signals, an EWEM will be implemented, described in section 2.4. Different lengths of the data sequence will also be considered, trying to find an optimal length for EURUSD. The length of the window is important according to the discussion in section 2.4.

3.3 Discretizing Continuous Data

As seen in section 2.2.6, one can use the HMM with both discrete and continuous emission probabilities. Both cases will be investigated in this master's thesis, giving an interesting comparison between the two models and their performances.

The major difference is that the data in the first case have to be discretized, which intuitively would mean a loss of information, possibly a lot if the discretizing is performed poorly. Discretizing of the features will always be according:

$$f_i^d = sign(f_i^c)$$

where f_i^d is the discrete value of the continuous variable f_i^c. For the cross feature, f_0, for example, a rise, no matter how big or small, is represented by a 1 and so on.

But using seven data series, where $f_i = \{-1, 0, 1\}$, equals a total number of 2187 possible observations at each point in time.[1] This might be a problem, and therefore a further discretizing will also be tested, reducing the number of possible observations. Looking at the six feature data series, one see that one can sort them into to three groups, or pairs if one so wish. Within each pair, discretizing will be performed accordingly:

- If $f_i^d = 1$ and $f_j^d = 1 \Rightarrow f_i^{d,new} = 1$
- If $f_i^d = -1$ and $f_j^d = -1 \Rightarrow f_i^{d,new} = -1$
- Otherwise set $f_i^{d,new} = 0$

This means that the number of additional features will go from six to three, and using the four features f_0^d, $f_1^{d,new}$, $f_2^{d,new}$, and $f_3^{d,new}$, reduces the number of possible observations to 81. But this further discretizing might lead to even larger information losses.

3.4 Initial Parameter Estimation

The first thing one must do when initializing the model is to supply the algorithm with a parameter set, λ. An easy and straightforward way of doing this is simply to let a random generator choose the probabilities for A, B, and Π, under certain constraints, e.g. $\sum_i \pi_i = 1$. Since the EM algorithm guarantees parameters that correspond to a local maximum of the likelihood function, this does not appear to be a huge problem. One want however to find the global optimum if possible, to ensure the best functionality of the model. Experience has also shown that good initial estimates, especially for B, are helpful in the discrete case and essential in the continuous case[2]. [17] A number of different solutions have been proposed to the problem, unfortunately none of them is a guarantee for finding the global optimum.

A simple K-means segmentation with clustering, presented in [17], will therefore be implemented and tested for the continuous version, while all initializing for the discrete case will be on the base of random numbers. The segmentation works accordingly:

1. A large amount of data is read into the program. The parameters Π, A, Σ, μ, and W are generated using random numbers. The log-likelihood, L_{λ}, for the parameter set λ is calculated.

2. Based on the data, i.e. the observation sequence, O, and the parameters a Viterbi path is generated, giving a state sequence Q. With the Viterbi path given, one can assign each observation, o_t with a state, $q_t = s_i$, thus enabling for grouping all observations $o_1, o_2, \ldots, o_T$ into N groups, where N is the number of states.

3. Each of the N groups from the previous step is partitioned into M segments, where M is the number of mixture components in each Gaussian mixture. The

[1] $3^7 = 2187$

[2] The equivalent to B in the continuous case are the set of mixture weights, W, and μ and Σ.

K-means clustering, that is used in this step, segments D data points into K disjoint subsets, S_j, each containing D_j data points so as to minimize the sum-of-square criterion:

$$J = \sum_{j=1}^{K} \sum_{d \in S_j} |x_d - \mu_j|^2$$

where x_d is a vector or scalar representing the dth data point and μ_j is the geometric centroid or mean of the data points in S_j.

4. One now have a total of NM groups. For each group, reflecting one particular mixture component and state, Σ and μ are calculated in standard manners. The elements in W are estimated accordingly: w_{ij} =(number of data points in group i, j)/(total number of data points).

5. With the re-estimated parameter set $\overline{\lambda}$, the log-likelihood $L_{\overline{\lambda}}$ is calculated. If $L_{\overline{\lambda}} - L_\lambda$ is less than a threshold d, the process is stopped; otherwise one go to 2 and restart the iteration procedure.

It should be noted here that the parameters Π and A do not change during the above described procedure — initiating these with random numbers is believed to be satisfactory. And with the random initiations before the K-means clustering of W, μ and Σ, one is still, to some extent, dependent of random numbers initializing the process. Finally, the amount of data that will be used when clustering have to be quite extensive, to get enough data points for each of the NM groups, increasing the possibility to good estimates. But with larger amounts of data, one also get more redundancy due to trends that might have perished long time ago. [3]

3.5 Generating Different Trading Signals

One of the most important steps in a trading strategy is to create a correct trading signal. Without a sophisticated signal, even a good parameter estimation can result in poor performance. For the discrete case, only one signal will be tested. For the continuous case one have two different choices — one ”standard” and one using MC simulations.

3.5.1 Trading Signal in the Discrete Case

Here, one simply determine the most probable state for tomorrow, $q^*_{t+1} = s_r$, according:

$$r = \text{argmax}_j \{a_{ij}\}$$

where i indicates the most probable state today according to the Viterbi path, i.e. $q_t = s_i$, and $r = 1, \ldots, N$ where N is the number of states in the model. Having done this, one wants to know the most probable observation, $o^*_{t+1} = o^s_{t+1}$. But since B is not available for the time $t+1$, one have to use the last known B, which is available for the time t, relaying on that this will give good enough estimates. This gives:

$$s = \text{argmax}_k \{b_r(o^k_t)\}$$

[3] No exponential weighting of observations was conducted during the initializing.

with $k = 1, \ldots, D$, where D is the number of possible observations. On the basis of o^*_{t+1}, the trading signal is determined. The simplest way is just by looking at f_0^d; if $f_0^d = 1$, go long, and if $f_0^d = -1$, go short.

A last option one have, which could be used to increase the probability of correct decisions, is to look at the probabilities a_{ir} and $b_r(o_t^s)$. One could argue, that the higher they are, the more certain the model is of the next state and the observation. Using a threshold, more unlikely observations can be sorted out.

3.5.2 Standard Signal in the Continuous Case

When deciding if to go long or short when dealing with continuous input data, the probability density function is used, given by the Gaussian mixture at the most probable state in the next time step, q^*_{t+1}, which is determined as above. Having obtained μ_{jm}, where m denotes the mixture component, $m = 1, \ldots, M$, and j the most probable state (i.e. $q^*_{t+1} = s_j$), a value for the return is given by:

$$g = \sum_{m=1}^{M} w_{jm} \mu_{jm}$$

where w_{jm} is the weight for a certain mixture component and state. If one instead of mixture only have a single Gaussian, g is simply the value given in the estimation procedure. $g > 0$ indicates that one should go long, $g < 0$ short, and $g = 0$ neutral.

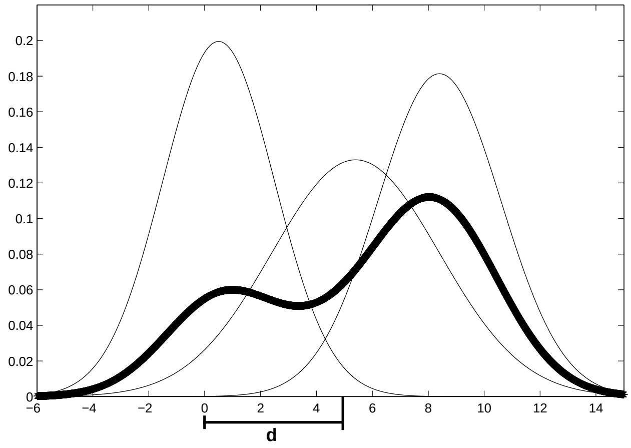


Figure 3.1: A Gaussian mixture with a threshold d that the predicted return (in this case positive returns) has to exceed in order to execute a trade. The Gaussians in the picture are not extracted from one the HMMs, and the x-scale is therefore not proportional to the usual values of the EURUSD returns.

As described in section 2.3.1, one have the possibility to filter trades that returns less than the spread. In this case, a threshold d is set, see figure 3.1, and if the absolute value of the returns does not exceed this, then the trade will not be executed. With the notation used above, this means that $g > d$ indicates that one should go long, $g < -d$ short, and $-d \leq g \leq d$ neutral. d will during testing be set to the mean of previous day.

3.5.3 Monte Carlo Simulation Signal in the Continuous Case

As mentioned in section 2.5, one can easily through MC simulation get the complete probability distribution for the underlying asset. One can then study the probability mass and how it is distributed, instead of just looking at the weighted means as in the the last section. The procedure is simple; through MC simulation each one of the M Gaussians is generated on the basis of μ_{jm} and Σ_{jm}, where m denotes the mixture component, $m = 1, \ldots, M$, and j the most probable state (just as in the previous case, the most probable state is determined as in section 3.5.1). The Gaussians are then weighted together, using w_{jm}, to one density function.

The trading decision is then taking in a similar way as above, but focus now lays on the probability mass instead of the mean — if more than 50 percent of the mass lies over zero, this means to go long, and if more than 50 percent lies bellow zero, one should go short. In the event that the probability mass is equally distributed on both sides of zero, the signal is go neutral. Having more than 50 percent of the probability mass above zero equals having $g > 0$ in the standard trading signal, and these two signals should therefore not differ from one another. The strength however with having the probability mass, is that one now can apply a probability measure to a up or down in an intuitive way. Instead of trying to relate the distance between μ and zero and how probable this is, it is simple to see for example if 55 percent of the probability mass lies above, or below, zero.

And exactly as in the last section, one can filter out low-yielding trades by adding a threshold d. Here, however, more than 50 percent, or some other probability, has to lie above, or below, for a trade to take place.

3.6 Variable Constraints and Modifications

Some features was added to the program, to avoid common pitfalls in the estimation process.

- Before sending multivariate data in to the EM algorithm, it should be ensured that the data from the different sources are normalized. This is simply done by extracting the mean from each data series and dividing it with its standard deviation. Non-normalized data otherwise constitutes a problem when different data series are compared.

- One problem in model building is overparametrization. This occurs when the number of free parameters in the model is very large and the training data is sparse. Except regulating the number of states and mixture components, some restraints are set to A and W.[4] Some events might be occurring very seldom in the training data, meaning that when re-estimating there might be probabilities tending towards zero. To avoid this, the elements in A and W that are below 10^{-4} are always set to this threshold.

- All covariance matrices are diagonal, since allowing for full covariance matrices gives positive log-likelihoods. This implies that the time series are uncorrelated, which is not entirely true since they show small linear correlations in between.

[4] Other suggested ways of dealing with overparametrization is for example *parameter tying*. This is however only applicable when a parameter is known to be same in two or more states, e.g. the observation density.

But according to [17] this is preferable, rather than using full covariance matrices, due to the difficulties in estimating good off-diagonal components in Σ.

- Furthermore, at each iteration $\lambda_{min}I$ is added to Σ, which is done to avoid singular covariance matrices which in its turn generates infinite likelihoods. λ_{min} is the smallest eigenvalue of Σ.

3.7 An Iterative Procedure

The following steps refer to the case where the HMM have continuous emission probabilities together with Gaussian mixtures. The procedure for other versions of the HMM is however very similar, following the same steps and only differing in some of the points.

Step 0: Initialization One starts by selecting the number of historical days to use for the re-estimation processes and if one wants to use the EWEM algorithm, and the number of states and mixture components to be used. Furthermore, one also have the choice between using the K-means segmentation for initializing the parameter set λ or if one simply wants to use a random generator for this.

Step 1: Updating the model The observation sequence O and parameter set λ, either given from the initializing process or from the last estimation, is taken and the EM algorithm is run, updating all parameters, giving us $\overline{\lambda}$. This equals what is done in Problem 3 in section 2.2.5.

Step 2: Generating a trading signal Given $\overline{\lambda}$ and the observation sequence O, the Viterbi path is determined, see Problem 2 in section 2.2.5, which in its turn gives the trading signal, according to the description above. If $t = T$ one stop; otherwise O is updated with today's observation and one return to step 1.

Throughout the entire work, MATLAB will be used as testing environment. A HMM Toolbox, containing some of the basic functions, will also be utilized.[5]

3.8 Evaluating the Model

The performance of the HMM is to be evaluated in three ways. First, the capability of correct predictions is statistically tested under a null hypothesis. Second, the risk taking is evaluated with measurements as Sharpe ratio, maximum drawdown and value-at-risk. And third, a beta index, reflecting market return, has been created for comparison.

3.8.1 Statistical Testing

In statistics, a null hypothesis is a hypothesis that is set up to be refuted to support an alternative hypothesis. When used, it is assumed to be true until statistical evidence indicates otherwise. In this case the null hypothesis is set up according:

$$H_0 : \text{the model is a random generator}$$

[5] The toolbox is written by Kevin Murphy, assistant professor at University of British Columbia, and has been approved by The MathWorks.

or stated mathematically:

$$H_0 : HR > 50\%$$

where HR is the hit ratio of the model. The reason for choosing 50% is that only the long/short positions are taken in consideration, and not the neutral. This leads to that the probability of a correct guess is 50%.

A P-value, which is the probability of obtaining a result at least as extreme as a given data point under the null hypothesis, must be calculated. Taking only the long/short positions into account, makes it possible to use the binomial distribution. The binomial distribution gives the discrete probability distribution $P_p(n|N)$ of obtaining exactly n successes out of N Bernoulli trials, where the result of each Bernoulli trial is true with probability p and false with probability $1-p$. The binomial distribution is given by:

$$P_p(n|N) = \binom{N}{n} p^n (1-p)^{N-n}$$

The probability of obtaining more successes than the n observed in a binomial distribution is:

$$P = \sum_{k=n+1}^{N} \binom{N}{k} p^k (1-p)^{N-k}$$

If P, i.e. the P-value, is less or equal than a chosen significance level, α, the null hypothesis, H_0, is rejected. If for example α is set to 0.05 and a P-value of 0.025 is obtained for the model, using $p = 0.5$, the null hypothesis — that the outcome can be ascribed to chance alone — is rejected.

It should be noted that the P-value is not to be seen as a probability of the model being merely a random generator, it should only be used for testing the null hypothesis. And obtaining a P-value greater than α does not say that the model *is* a random generator, only that we can not reject it as not true.

3.8.2 Sharpe Ratio

When evaluating the payoff from an investment, it can be interesting putting the rate of return against the amount of risk. A measure that does this is the Sharpe ratio, which has a strong connection to theory of CAPM. The Sharpe ratio for a financial portfolio in general is calculated according:

$$\frac{\mu_p - r_f}{\sigma_p}$$

where μ_p and σ_p are the return and standard deviation for a portfolio p, and r_f the risk-free rate, e.g. the rate for a specific government bond. A high Sharpe ratio indicates a high return in relation to the risk, which is desirable.

But as discussed in section 2.1.1, the risk-free rate is usually left out when dealing with FX investments. The Sharpe ratios in this master's thesis is calculated as:

$$\frac{\mu_p^*}{\sigma_p}$$

where μ_p^* is the mean return for the FX portfolio with respect to the spread.

3.8.3 Value at Risk

When investing in a certain asset or portfolio of assets it is important to know the risk of the investment. One frequently used risk measure is value-at-risk (VaR). When using VaR, one is interested in finding the value, VaR_p, which holds for the following statement: *it is (1-α) percent certain that one will not loose more than VaR_p units in the following N days.* Here, VaR_p is a function of two variables, namely the time horizon (N) and the confidence level α. The confidence level is represented by λ_{α} that satisfies $P(X \geq \lambda_\alpha) = \alpha$ where $X \sim N(0,1)$. Now one can define VaR for a given portfolio of assets, p, as follows

$$VaR_p = \lambda_\alpha V_p \sigma_p \sqrt{N}$$

where σ_p and V_p are the portfolio's volatility on a daily basis and total value respectively. [12]

The volatility can be calculated using different methods, e.g. weighted scheme (WS) or EWMA. The WS is a simple way of describing the volatility and can be formulated as

$$\sigma_p^2(n) = \sum_{i=1}^{m} w_i u_{n-i}^2, \sum_i w_i = 1 \, w_i \geq 0 \forall i$$

where w_i is the weight for the specific return u_{n-i} at day n. The length of the window is m, which consider the last m returns. If all weights are set equal, one get a standard moving average with length m.

If using an EWMA there is a small modification made to the WS. The weights are exponential which makes the volatility a function of not only the return. It is now also a function of the calculated volatility for the last time period. It can be expressed as

$$\sigma_p^2(n) = a\sigma_p^2(n-1) + (1-a)u_{n-1}^2$$

where $a \in [0,1]$ is the decay factor, which determine the weight put on the most recent volatility estimate. [12]

When calculating VaR the important thing when choosing the way for estimating the volatility is that the loss exceeds VaR as close to α percent as possible. By testing different lengths of moving averages and decay factors one should chose model for estimation after the above stated criteria.

3.8.4 Maximum Drawdown

The maximum loss from a market peak to the following valley, often called the maximum drawdown (MDD), is a measure of how large one's losses can be. Large drawdowns usually leads to fund redemption, and so the MDD is the risk measure of choice for many money management professionals. A reasonable low MDD is therefore critical for the success of any investment strategy, certainly if its investors has a short investment horizon. [23] In figure 3.2 one can see the maximum drawdown for EURUSD for the period which has been used during testing. The cumulative loss is almost 20 percent, which implicates a loss of almost 16 percent of the portfolio value.

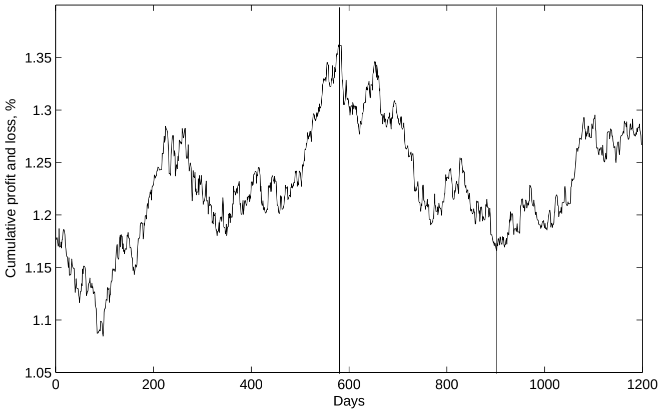


Figure 3.2: The maximum drawdown for the EURUSD during the fixed back testing period.

3.8.5 A Comparative Beta Index

To have something to compare the payoffs from the HMM trading strategies, a comparative index has been created. This should be seen as a market return on FX and thus reflecting a beta strategy, see section 2.1.1. This index will also form the basis when calculating the excess return, α, comparing the realized portfolio return with the one given by CAPM.

Three trading strategies were discussed in the last chapter — carry, momentum and valuation. Valuation was however omitted due to lack of satisfactory background PPP data. According to the carry strategy, one should go short in low-yielding currencies and long in high-yielding, whereas using momentum, one calculates and compares two moving averages to see if there has been a change in trend. For a more detailed explanation of the strategies see section 2.1. For the carry, reallocation was made every three months, using 3m interbank offered rates (e.g. LIBOR, STIBOR, NIBOR) for determining which currencies to long and short in. For the momentum strategy, reallocation was performed more frequently, on a monthly basis, since trend changes might occur more often than every third month.

The currencies selected to be available for investment, are those belonging to G10. This group of currencies are highly liquid, thus making them a good selection for a market index since they are ”easy” to trade with. The ten currencies are: Euro (EUR), US Dollars (USD), Japanese Yen (JPY), British Pound (GBP), Swiss Franc (CHF), Canadian Dollars (CAD), Swedish Kronor (SEK), Norwegian Kronor (NOK), Australian Dollars (AUD), and New Zealand Dollars (NZD). For both strategies we chose the two currency crosses that, according historical data, would yield the most. All crosses was stated versus the Euro, all other crosses have been created from this.

When calculating the yield, the effect of interest payments was also taken into account, except from the profit or loss due to changes in the exchanges rates. This effect was however almost negligible for the momentum strategy. Furthermore, an approximative transaction cost of 0.1 percent per trade was also added. The interest rates used were the 1m and 3m interbank offered rates, as mentioned above. [6] All data

[6] For the AUD and NZD, deposit rates had to be used for the 1m rates, since there were no LIBOR rates

used was provided by Nordea.

The index had a total return of 15.9 percent, corresponding to an average annual return of 4.8 percent. The annualized volatility amounted to 5.8 percent, thus yielding a Sharpe ratio of 0.83. These results are coherent with other market indices, e.g. the DBCR Index. In figure 3.3 the return of the index is plotted, and it starts in June 2003 and ends in September 2006, which is the same time period that will be used when the HMM is to be tested.

The without question best strategy for this particular time period was the carry, yielding the highest return as well as the highest Sharpe ratio. One could alter the results by removing one of the strategies, making the index perform better or worse. This will however not be done, since a index should reflect a set, i.e. at least two, strategies and not one specific.

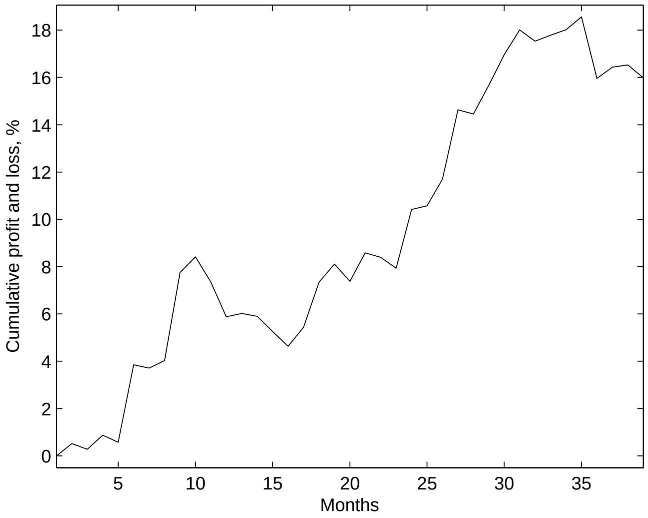


Figure 3.3: Total return for the comparative beta index.

available until 2003.

Chapter 4

Results

In this chapter the results from the back-testing is presented and it is divided into two sections, one for the case with discrete emission probabilities and one for the continuous case. During the sequences of test, various window lengths, number of states and Gaussian mixtures, and trading signals will be tested, as well as special applications like the K-means clustering. In figure 4.1 the development of the EURUSD exchange rate during the back-testing period is plotted, the dashed line indicates where the out-of-sample period starts.

In section 4.1 – 4.2, all testing has been performed with the spread excluded. Not knowing exactly how much of the profits the spread would take, it was easier to see if the model could find any patterns at all in the data, excluding the spread. The issue regarding the spread is addressed again in section 4.3. This means that all Sharpe ratios in the first two sections are calculated without consideration to the spread, and should therefore only be used to compare the HMMs in between.

A problem that comes with large sets of information, is how to find the relevant information. [1] In this chapter some, but far from all, tests are presented. In general the best performing models together with small adjustments are chosen, which certainly affects the presented material in such a way that the HMM seems to perform better than it maybe does.

But during the rather extensive testing no extreme negative results where generated, meaning that there where no negative payoffs that to the amount exceeded the payoffs of the best performing models. Negative payoffs usually amounted to a maximum of minus 10 – 15 percent.

4.1 The Discrete Model

A set of HMMs with discrete emission probabilities are reviewed, first where only the EURUSD cross is used as input data and then where the features also have been applied to the model.

[1] This problem is addressed in *data mining*, which is the principle of sorting through large amounts of data and picking out relevant information.

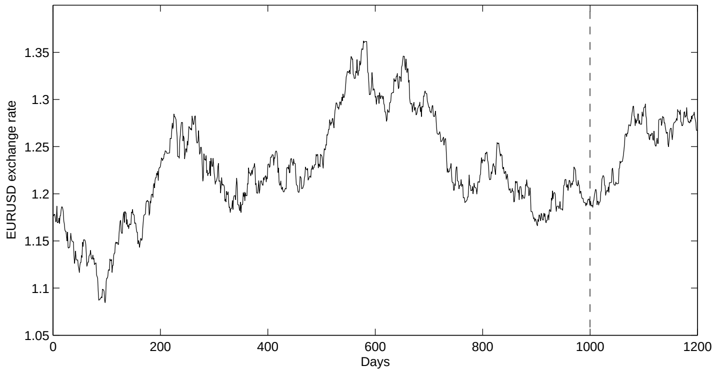


Figure 4.1: Development of the EURUSD exchange rate during the back-testing period.

4.1.1 Using Only the Currency Cross as Input Data

Trying with different window lengths, the number of states is initially set to three, which was shown to be the optimal number. For a window of length 30 days, the total payoff amounts to over 40 percent, equaling an average annual return over 10 percent. But the P-value is however not small enough to refute the null hypothesis on a 95 percent confidence level. Increasing the window length to 65 days, the model performs extremely poor yielding -17.4 percent over the time period, but further increasing the window up to a 100 days, gives a model that returns about the same as in the first case during the training period, but substantially worse in the out-of-sample period. The null hypothesis is however refutable on the 95 percent level and the Sharpe ratio amounts to 0.91. For test results, see figures 4.2 – 4.4. Further test with different window lengths shows that there is no other model settings that gives better results.

Increasing or decreasing the number of states from three using the 100 days window, which gave the best results as just seen, the model is most of the time close to something that looks like a random generator. The payoffs are very volatile, the total returns often negative, and the P-values high. Trying with other time window lengths than 100 days, for the cases with two, four, five, . . . states, yields nothing.

Turning again to the case with three states and a 100 days window, the best so far, a threshold is set to the probability $P(q_{t+1} = s_i|q_t = s_j) \cdot P(b_i(o_t)|q_t = s_i)$. This means that the model should be more certain about the outcome of tomorrow, thus giving more accurate predictions. Using a threshold of 0.4, the average annual return increases from 7.2 to 11.3 percent, see figure 4.5, and the Sharpe ratio from 0.91 to 1.53 while the number of trades decreases. The optimal size of the threshold considering the payoff was found in an iterative process, trying for different values, where the hit ratio peaked for 0.4.

4.1.2 Adding Features to the Discrete Model

Adding features to the discrete HMM does not improve results. On the contrary, they seem to worsen things — the trajectories are even more volatile than before and the payoffs never exceeds what could be considered as acceptable. In figure 4.6, one can see the results when using a 100 days window, three states and six features. The further discretizing, proposed in section 3.3, does not help either; the model still performs very poorly, see figure 4.7.

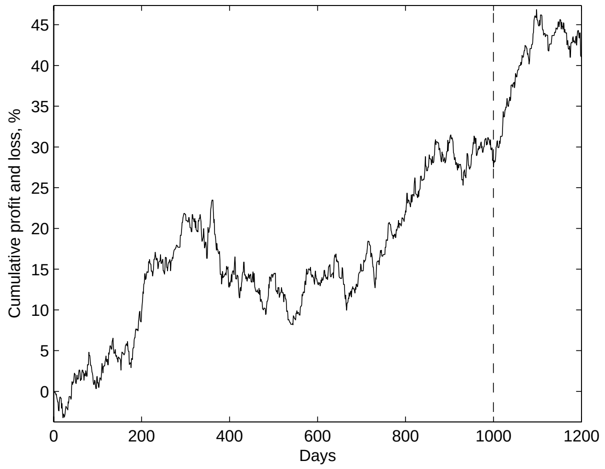


Figure 4.2: Using only the cross as input data with a 30 days window and 3 states.

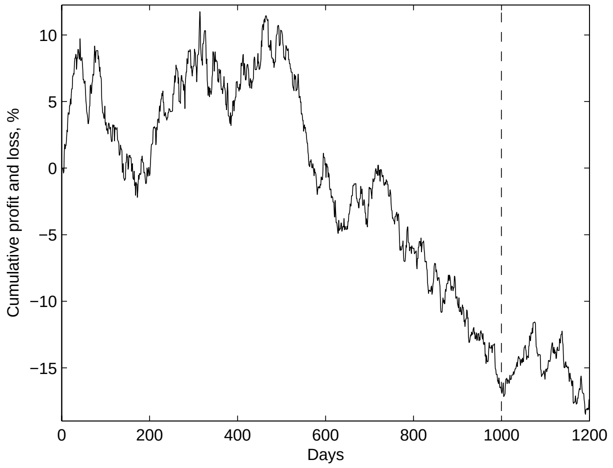


Figure 4.3: Using only the cross as input data with a 65 days window and 3 states.

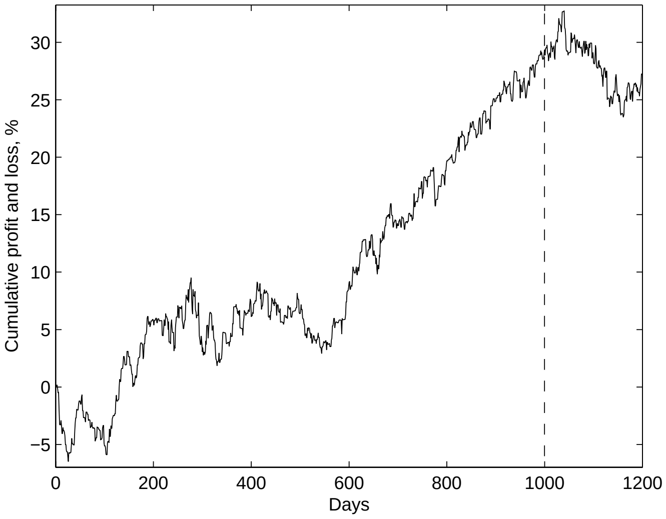


Figure 4.4: Using only the cross as input data with a 100 days window and 3 states.

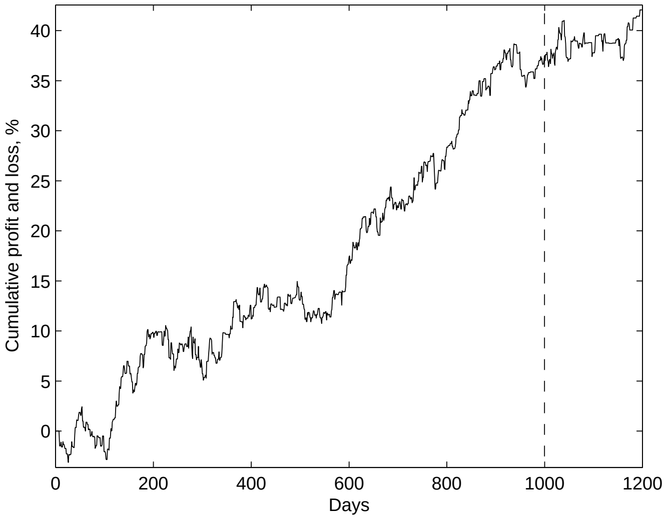


Figure 4.5: Using only the cross as input data with a 100 days window and 3 states, as in figure 4.4, but where a trade only is executed if $P(q_{t+1} = s_i|q_t = s_j) \cdot P(b_i(o_t)|q_t = s_i) > 0.4$.

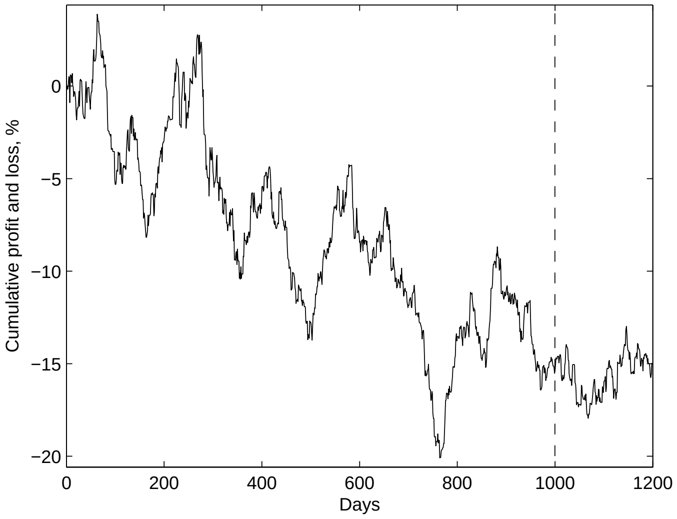


Figure 4.6: Using the cross and the 6 features as input data with a 100 days window and 3 states.

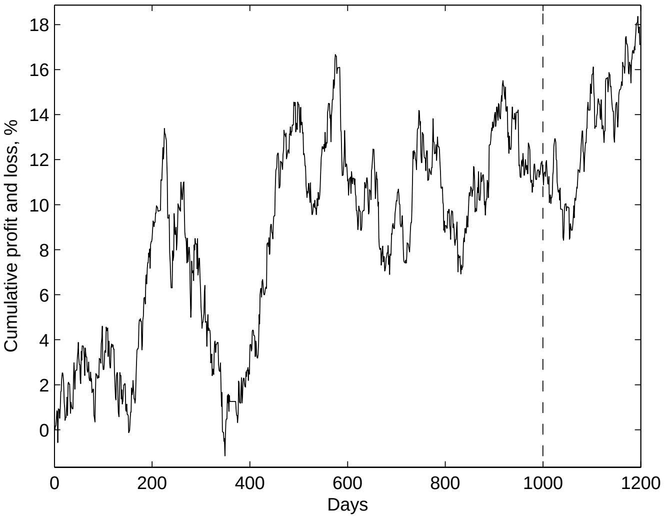


Figure 4.7: Using the cross and the 3 features as input data with a 100 days window and 3 states.

4.2 The Continuous Model

The second model is a continuous version of the discrete HMM, which was considered in the previous section. It is more complex because of its use of GMMs as continuous emission probabilities, which makes it possible to use continuous scaled input data. All tests were carried out using four hidden states together with two mixture components, which through empirical studies was found to be the most profitable set up.

4.2.1 Prediction Using a Weighted Mean of Gaussian Mixtures

First of all the continuous model was tested on the simplest input data possible, namely the cross with features excluded. The best test results were found when using 20 days of input data where a total return of 22.7 percent was obtained during the training period. As seen in figure 4.8, the model generates almost as much as 30 percent during the first two years but is less accurate to find return during the last period. After a drawdown during the out of sample period the total rate of return ends at a level of 17.5 percent. If the window size instead is set to 30 days together with four states and two Gaussian mixtures, a total return of 20.6 percent is obtained during the in sample period, seen in 4.9. For both these tests one can see that the trajectories are very volatile, which implicates low Sharpe ratios.

To add stability to the trading strategies, in some sense, the six features where added. By testing 20 and 30 days of input as for figures 4.8 - 4.9 the result was found to be worse than using only the cross as input. Therefore the tests presented in this section will be using a historical window size larger than 50 days, which has been shown more accurate when using multivariate data.

Using 50 days of input the model generates a return of 22.4 percent together with a Sharpe ratio of 0.58. The trajectory, plotted in figure 4.10, is highly volatile and faces a large drawdown at 480 days, which leads to an decrease in the overall payoff with more than 20 percent. There is also a large decrease between day 700 and 1000.

When 75 days are used instead of 50 one can see a more stabile trajectory, shown in figure 4.11. The obtained Sharpe ratio is now 1.29 together with a total return of 43.4 percent.

For 100 days of input the model performs very well during most of the whole period, except between 750 and 950 days, and generates an annual rate of return of 18.1 percent. A total rate of return of 72.8 percent is generated, which together with an annual volatility of 9.8 percent gives a Sharpe ratio of 1.91. One can see in figure 4.12 that the model performs as well out of sample as during the training period. The P-value is 0.017, which indicates that the null hypothesis stated in section 3.8.1 can be discarded.

Using even more data as input, as much as 125 days, does not improve the result further. As shown in figure 4.13 one can see that the model is generating a stabile trajectory for the first 700 days with no larger drawdown. During the last part of the test period the rate of return is constant until reaching the out of sample period where the cumulative profit decreases with 10 percent during the last 200 days ending up at a level of 42.1 percent. Together with a annual volatility of 8.4 percent the obtained Sharpe ratio is 1.39.

When adding the EWEM on the standard HMM the performance shown in figure 4.14 is obtained. For a moving average over 75 days and a period of historical data of 100 days the total rate of return is less than for the model generating the trajectory shown in figure 4.12. The total rate of return obtained after the whole period is 47.4

percent. Looking at the volatility one finds that it is lower than when not using EWEM, 8.8 percent annually instead of 9.8.

It has earlier been discussed how important the initial parameters are for the stability of a continuous HMM. When using K-means clustering with 100 days of input data the return develops as shown in figure 4.15. One can see that the trajectory has two large drawdowns, one at 200 days and one at 800 days. The total return for the whole period is 46.2 percent, which together with a volatility of 9.9 percent gives a Sharpe ratio of 1.30.

4.2.2 Using Monte Carlo Simulation to Project the Distribution

As presented in section 3.5 one can use MC simulation to project the underlying distribution of the estimated return, taking advantage of the information such as mean and standard deviation given from the parameter estimation. When using the set up that generated the highest overall return and Sharpe ratio using the standard signal one can in figure 4.16 see that the return is lower using MC simulation. The total return is 63.2 percent comparing with 72.8 percent. The volatility for the two tests are comparable why the Sharpe ratio is lower for the model using MC simulation, 1.69 compared to 1.91. The model using MC simulation has a low P-value, 0.022, which makes it possible to reject the null hypothesis.

It is easy to set a level of the required certainty using MC simulation. When using a minimum level of probability set to 55 percent, the cumulative profit develops as shown in figure 4.17. Comparing this result to the one given in figure 4.16 one can find many interesting results. The total rate of return is lower when using a minimum probability, 52.2 versus 63.2 percent. On the other hand the hit ratio is higher and the P-value together with the number of trades lower when a minimum probability is used. The hit ratio is now as high as 54.1 percent which gives the model a P-value of 0.014.

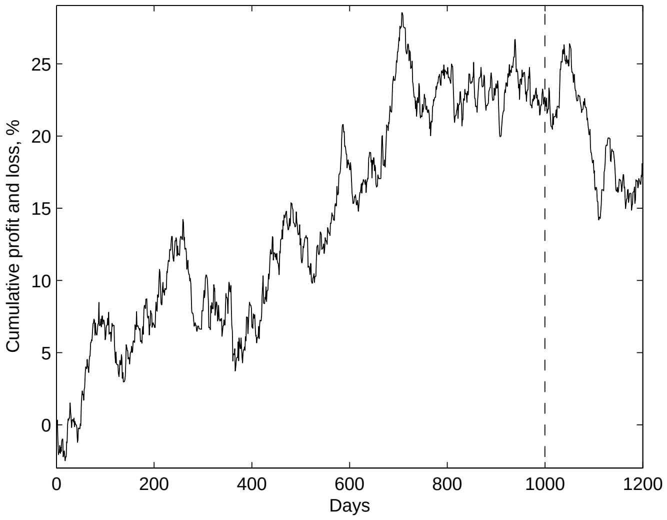


Figure 4.8: Using only the cross as input data with a 20 days window, 4 states and 2 mixture components.

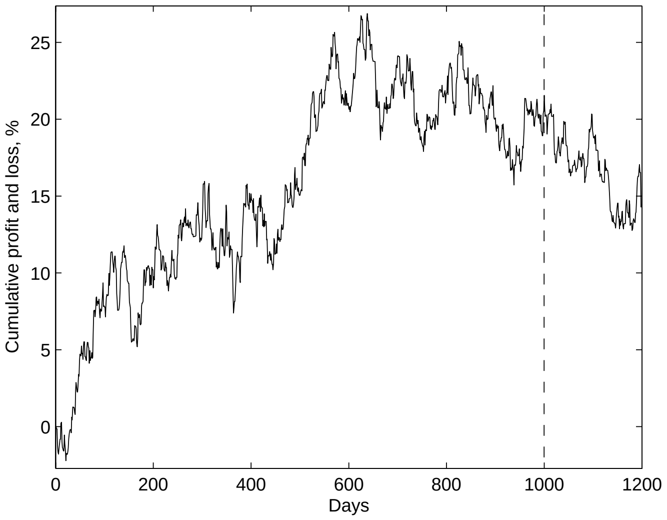


Figure 4.9: Using only the cross as input data with a 30 days window, 4 states and 2 mixture components.

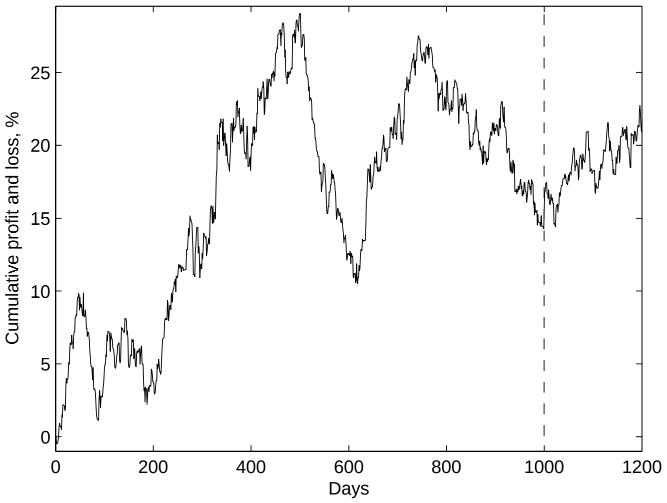


Figure 4.10: Using features as well as the cross as input data with a 50 days window, 4 states and 2 mixture components.

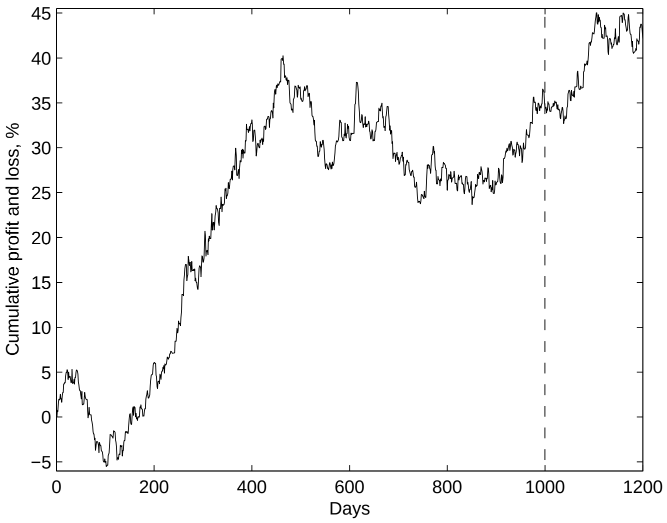


Figure 4.11: Using features as well as the cross as input data with a 75 days window, 4 states and 2 mixture components.

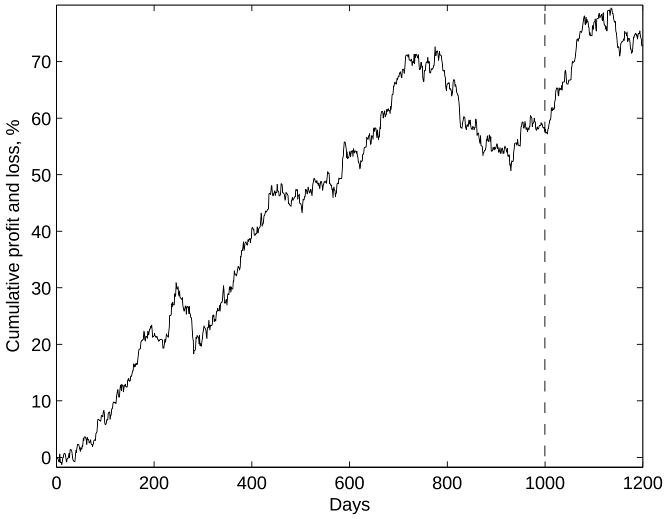


Figure 4.12: Using features as well as the cross as input data with a 100 days window, 4 states and 2 mixture components.

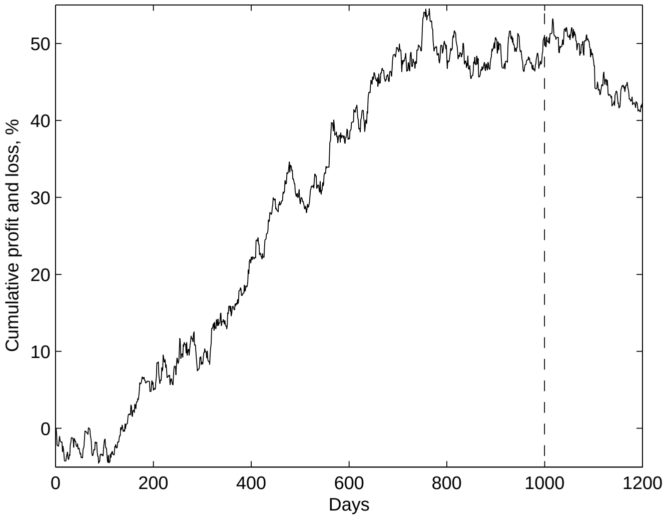


Figure 4.13: Using features as well as the cross as input data with a 125 days window, 4 states and 2 mixture components.

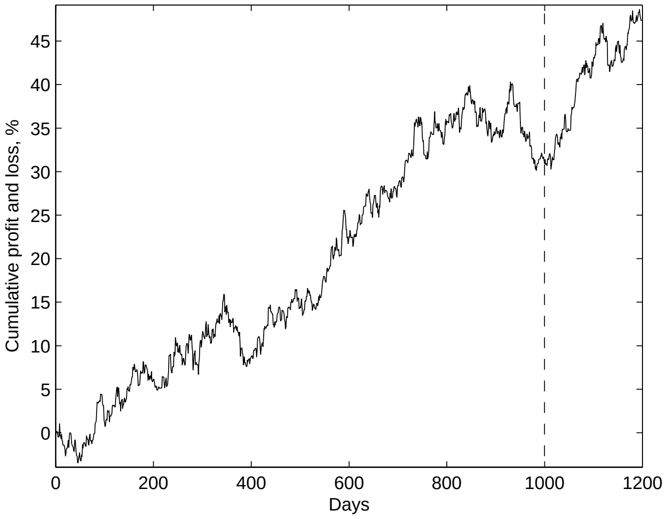


Figure 4.14: Using features as well as the cross as input data with a 100 days window, 4 states and 2 mixture components. To weight the input we use EWEM with a 75 days moving average.

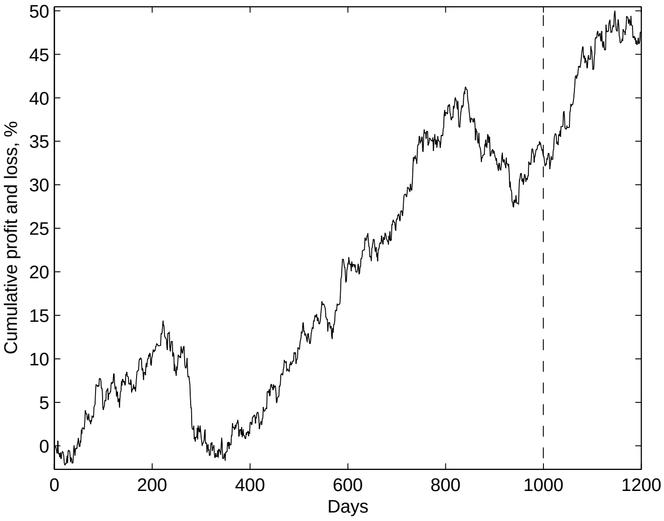


Figure 4.15: Using features as well as the cross as input data with a 100 days window, 4 states and 2 mixture components. To initiate the parameters K-means clustering is used.

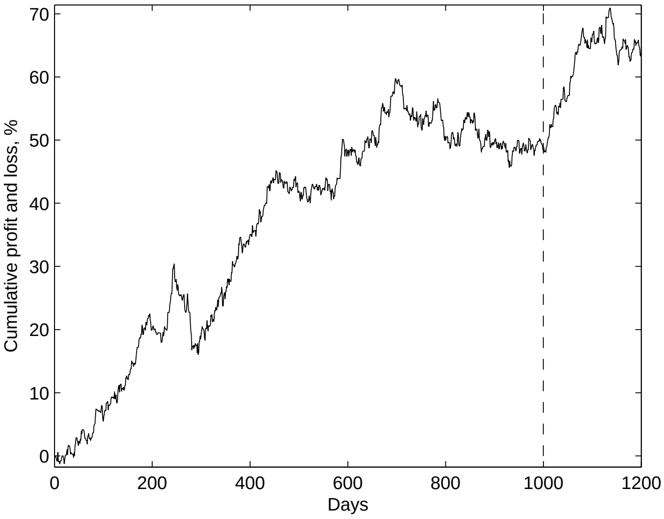


Figure 4.16: Using features as well as the cross as input data with a 100 days window, 4 states and 2 mixture components. As trading signal we use the probability mass simulated through Monte Carlo simulation.

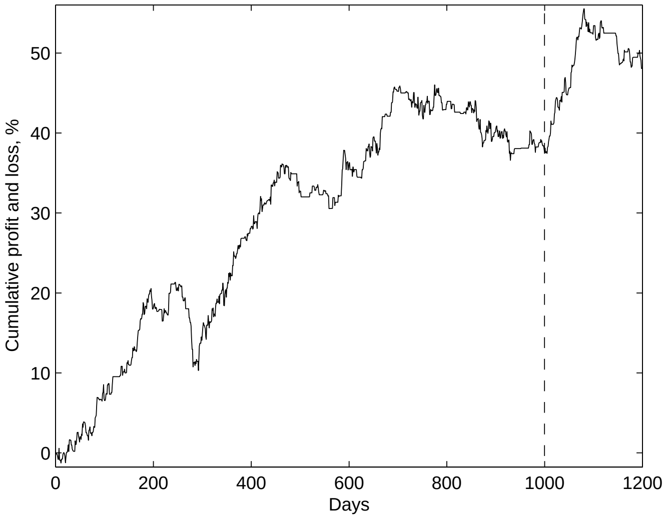


Figure 4.17: Using features as well as the cross as input data with a 100 days window, 4 states and 2 mixture components. As trading signal we use the probability mass simulated through Monte Carlo simulation. A minimum level of 55 percent is set on the probability mass.

4.3 Including the Spread

Up until now all testing have been performed with the spread excluded, since it could be easier to see if the model was capable of finding any patterns at all in the analyzed data. But in figure 4.18 and 4.19, the payoff curves are plotted for the best performing discrete and continuous models, and where the spread has been considered. The best performing models has been chosen considering the Sharpe ratios, since it also takes the accumulated payoff in consideration, compared with risk measures.

As one can see, they still perform well but with lower rates of return, as expected. In the discrete case, the total payoff amounts to 33.3 percent for the whole time period, which equals a Sharpe ratio of 1.25. For the continuous model the Sharpe ratio decreases from 1.91 to 1.81. At an annual basis the model generates 17.0 percent, which gives a total return of 67.7 percent.

The calculated βs for the discrete as well as the continuous model were close to zero, giving αs close to the average annual return. This also means that the correlation between these two strategies and the market index are close to non-existing and this is probably due to that the EURUSD only constitutes a relatively small part of the market index. The high level of the excess returns is also dependent on the fact that no risk free rate is taken into consideration.

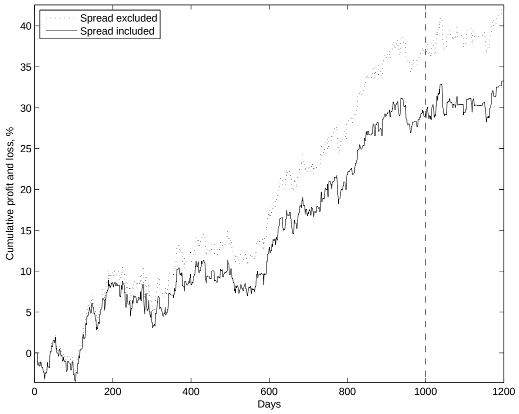


Figure 4.18: Payoff for the best performing discrete model (only using the currency cross, a 100 days time window, 3 states, and a 0.4 threshold) with the spread included. As a comparison, the original trajectory is plotted as the dotted line.

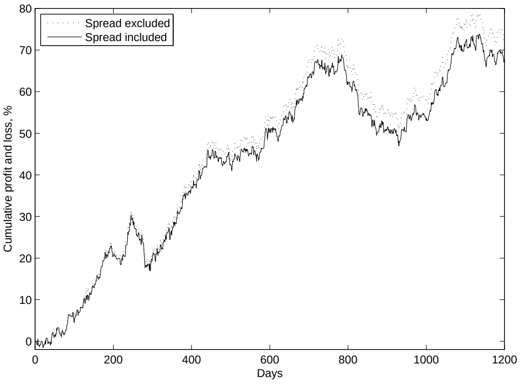


Figure 4.19: Payoff for the best performing continuous model (using the currency cross together with features, a 100 days time window, 4 states and 2 Gaussian mixtures) with the spread included. As a comparison, the original trajectory is plotted as the dotted line.

In section 3.8 one can together with the statistical test and Sharpe ratio also find VaR and MDD. To evaluate a strategy it is important to consider the risk involved, why these measures will be calculated for the best discrete and continuous model respectively. For the discrete model the VaR is on average 0.65 percent of the portfolio value on a daily basis, which gives 1.7 percent a week during the test period. The maximum drawdown is 5.5 percent cumulative, which implicates a maximum loss of 5.9 percent of the portfolio value. The continuous case has a higher VaR, on average 0.81 percent of the portfolio a day and 2.1 percent a week. The maximum drawdown is cumulatively 21.3 percent, which constitute 13.5 percent of the portfolio value. This should be compared to the cumulative MDD for the currency index created in this master's thesis, which is approximately 5 percent.

4.3.1 Filtering Trades

As stated in section 2.3.1 one can use a threshold, d, to filter trades which is predicted to have a payoff that do not exceed d. In figure 4.20 the cumulative profit for the standard trading signal when using a threshold equals today's spread is presented. One can see that the filter is too strict, in some sense. It filters out trades in a way which leads to a decreasing overall rate of return. The Sharpe ratio decreases from 1.81 to 1.63 and the total rate of return from 67.7 to 54.1.

When using the MC simulation as a part of the trading signal it is easier to relate the threshold to a probability mass, as described in 3.5.3. When setting a threshold equal to today's spread and at the same time set the required probability mass to be bigger than 55 percent the trajectory develops as shown in figure 4.21. One interesting detail is that the model using a threshold does outperform the model without a threshold during the last 500 days. Still, the Sharpe ratio has decreased depending on the lower final return.

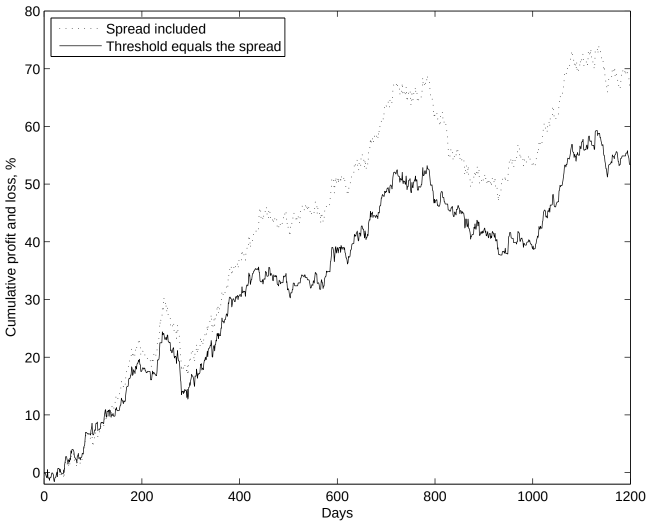


Figure 4.20: Payoff for the best performing continuous model (using the currency cross together with features, a 100 days time window, 4 states and 2 Gaussian mixtures) with the spread included. A threshold is set to equal the spread.

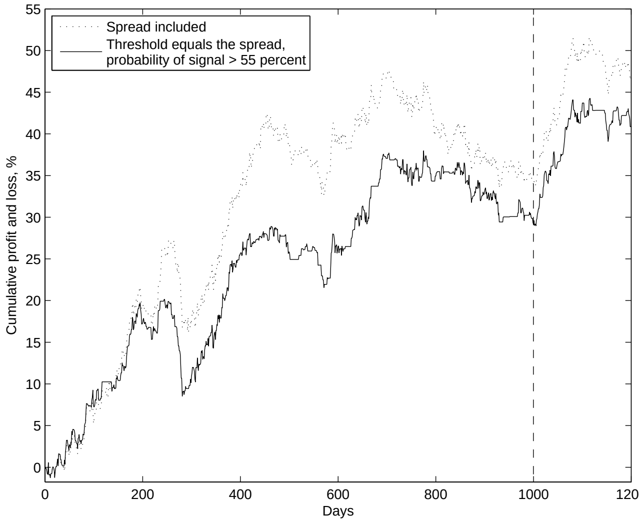


Figure 4.21: Payoff for the best performing continuous model with MC simulation (using the currency cross together with features, a 100 days time window, 4 states and 2 Gaussian mixtures) with the spread included. A threshold is set to the trading signal, a minimum of 55 percent for the probability mass.

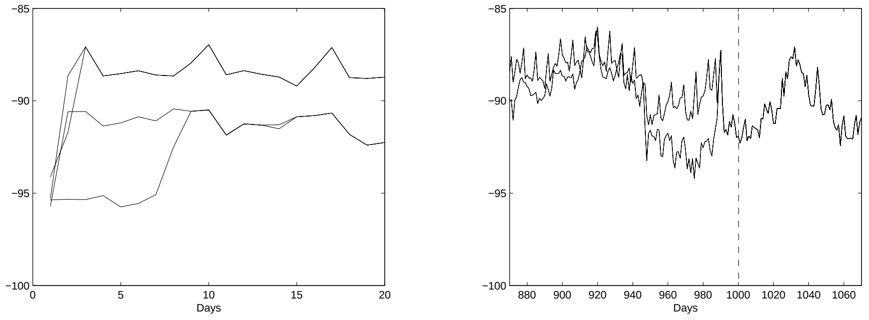
4.4 Log-likelihoods, Random Numbers and Convergence

All models, even the one where the K-means clustering is performed, rely on the built-in random generator in Matlab since the model parameters, $\lambda = \{A, B, \Pi\}$, are initiated by some random numbers. Previously when testing with different models, the same random numbers[2] have been used to ease a comparison, but an interesting question is how much this effects the finding of different optimums. To answer this question the maximum likelihood method is addressed.

Each time a re-estimation of λ is performed a log-likelihood is obtained. The value of the log-likelihood corresponds to a certain local maximum on the surface where the optimization is performed. As a result, a number of 1200 log-likelihoods is obtained during testing, equaling the number of days in the period. Testing with different random numbers but with the same settings regarding window size, numbers of states, trading signal, etc., gives a number of log-likelihood sequences. Four different sequences, generated from four different sets of random numbers, are plotted for the case of discrete emission probabilities in figure 4.22 and for the continuous case seven different sequences are generated, see figure 4.23.

As one can see in figure 4.22(a) all log-likelihood sequences start in different optimums in the discrete case. But after only 20 days, four sequences has become two and when approaching the out-of-sample period the two latter converges into one, see figure 4.22(b). This means that starting with four different parameter sets, $\lambda_1 \neq \lambda_2 \neq \lambda_3 \neq \lambda_4$, ends with four identical parameter sets after 1000 days,

[2] An initial statement in the code lets us control which random numbers that are used each time.

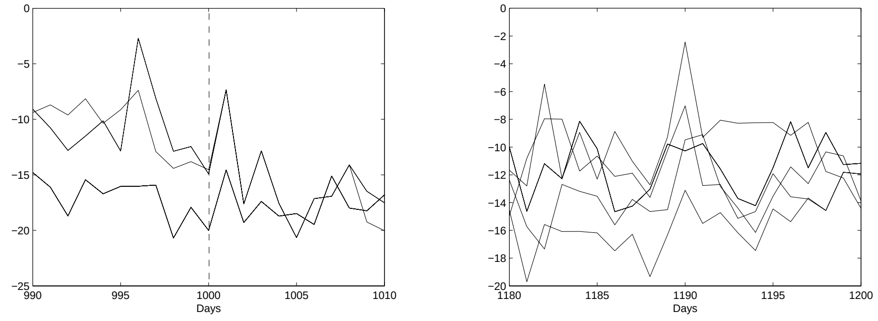


(a) (b)

Figure 4.22: (a) Log-likelihood sequence for the first 20 days for the discrete model (b) Log-likelihood sequence for days 870 to 1070 for the discrete model

$\lambda_1 = \lambda_2 = \lambda_3 = \lambda_4$. As a consequence, the payoff curve is for example identical after 1000 days for all four random initiations.

But for the continuous case, convergence is less evident. The initially seven log-likelihood sequences has only converged to a number of two-three paths after 1000 days, see figure 4.23(a), and close to 1200 days the sequences has diverged into a number of four-five, see figure 4.23(b). Applying the K-means clustering gives similar results.



(a) (b)

Figure 4.23: (a) Log-likelihood sequence for days 990 to 1010 for the continuous model (b) Log-likelihood sequence for the last 20 days for the continuous model

All aforementioned trials have been carried out using only the currency cross and a 100 days time window in the discrete case and a 20 days time window in the continuous case, this since the log-likelihood sequences in the latter case does not exhibit convergence otherwise. This particular behavior, with faster convergence the smaller the time window is, is valid also for other cases, and the opposite is obviously also true.

Adding features reduces the convergence pattern — for the discrete case an initial number of eight sequences only converges to four within 1000 days and similar results are given for the continuous case.

Another interesting observation is that even if one sequence has a generally higher log-likelihood compared to the others, i.e. a higher probability for the observed observations given λ, this does not mean that the payoff nor the hit ratio are automatically higher.

Chapter 5

Analysis

In this chapter the overall performance will be discussed on the basis of different aspects, found through the tests made in chapter 4. This will then form as a basis for the conclusions in the next chapter.

5.1 Effects of Different Time Windows

The number of historical days that the model make use of plays an important role. Using for example a 30 days window, using only the currency cross as input data, both the discrete and continuous model is functioning well. But the results are not exclusive in the discrete case where a 100 days window also manages to generate a high payoff during the back-testing period. Some time frames must obviously contain the ”right” information which can make the discrete HMM work for both large and small windows. It is therefore difficult to say if the discrete sees the same patterns as the continuous in the historical data.

The discrete HMM does not function at all when adding features, and the reasons for this is discussed in section 5.3. In the continuous case, the window has to be extended to make the model work properly. Since information is added, and not removed, one could intuitively believe that it would work at least as well as in the previous case. One explanation, for needing more data, could be that more parameters have to be estimated as μ and σ goes from scalar values to 7x1 and 7x7[1] matrices. Even though the amount of information in a specific time window also increases with a factor 7, even more data might be required. Another explanation could be the non-linear covariations between the features and the EURUSD cross, that might occur over other, in this case longer, time periods.

The implementation of the EWEM algorithm was made to cope with the non-stationarity in the time series. During testing one could see that it was only for $k = 75$ ($\rho = 0.03$) that the model, making use of EWEM, generated a good payoff and where the null hypothesis could be rejected. But the result was however not as good as in the case where EWEM was omitted, ceteris paribus, and the observations made here do not rule in favor for the EWEM. One possible reason for this, since the EWEM previously has worked fine on equity data, is that FX data simply has other, or less time dependent, characteristics.

[1] Since diagonal covariance matrices are used, it is actually only seven values that have to be estimated.

5.2 Hidden States and Gaussian Mixture Components

As found during the empirical study, the combination of the number of states and mixture components is important for the overall performance. Through numerous tests, it has been shown that different set ups have been optimal for the discrete and the continuous case. For the discrete models three states was found to be the best number at the same time as four states together with two mixture components gave the best overall performance for the continuous framework.

When comparing the two discrete cases, single-variate with multivariate, it has through empirical studies been found that the same number of states has been the optimal way of describing the underlying Markov process. The same result has been found for the continuous models, both the single-variate and the multivariate. The same number of states is optimal for replicating the underlying historical data series. This implicates that the number of time series used do not influence the chosen number of hidden states, i.e. the degree of freedom needed for the hidden Markov model is the same when replicating one as well as seven input series. This could have something to do with the nonlinear correlation between the time series. They do support each other in some sense and might therefore be replicated by the same number of hidden states. Therefore one should not need more states to replicate the multivariate case than the single-variate.

If comparing the discrete case with the continuous, one can see a different result, namely that the optimal number of hidden states for the continuous model is higher than for the discrete set up. The need of a higher degree of freedom for the continuous Markov process should therefore be implicated by the characteristics of the input data. In the discrete case, ups and downs are simply represented by binary digits while they in the continuous case also contain information of how big or small past movements has been. This larger amount of information in each data point, might be one of the reasons which makes the continuous Markov model dependent on a higher number of possible hidden states.

When looking at the task of finding the most suitable number of Gaussian mixtures for the continuous framework it has been found that the use of two Gaussians is the best way of describing the underlying distribution of the input data. This implicates that the replicated data series is not normally distributed. Instead it is generated by a mixture of two Gaussians where the weights indicates significant skewness and kurtosis, which is in line with what one could expect initially.

5.3 Using Features as a Support

Adding features does not improve results in the discrete case. The explanation for this should be the extreme increase — with a factor of over 700 going from one to seven time series — in the number of possible observations at each time instant. Even if some observations might be more reoccurring, the estimates get highly unreliable. Doubling the time window does not help much neither, more data is needed for more accurate estimates. But further increasing the time window, would most probably worsen the ability of the model to comprehend short-term trends. Reducing the number of features from six to three does not either work, and the reasons for this can be two: first the number of possible observations might still be too large, and second the information loss when further discretizing the data might have been too extensive.

The continuous model does however seem to find use for the additional data. This does not apply to all model set ups, as seen in chapter 4, but the best results spring from models making use of additional time series. Apparently the larger amount of information that comes with the features, as mentioned in the previous section, improves the parameter estimations. Whereas the discrete model has a finite codebook, strongly linked with the probability $P(o_t|q_t)$ and the prediction, the continuous using Gaussian mixtures does not depend on specific observation probabilities. With its estimations of W, μ, and Σ, it is more flexible in its structure. Furthermore, these results also implicates that the supplied features, that has previously worked well with other frameworks, also functions together with HMM.

5.4 The Use of Different Trading Signals

In the discrete case, the trading signal generating has been pretty straightforward. Either with or without additional features to the currency cross, the most probable cross return for the next day was the only determinant. Including a threshold for more accurate predictions gave ambiguous results; up to 0.4 the hit ratio increased from 54.4 percent to 57.0 percent, just as one would expect, but further increasing it lead to a decrease in the hit ratio. One source of error could simply be that the model is in the ”wrong” state at time t, meaning that the high probabilities for the next state and the most likely observations refers to something that is not correct. This however does not explain the peak in the hit ratio for the threshold 0.4.

Two trading signals has been used in the continuous case for generating a trading signal. The overall performance for the two strategies has varied from test to test, why it is important to find the underlying cause of this bias. The two signals is similar in the way they use the mean of the Gaussian for predicting the rate of return for the next time period, where it for the MC simulation is used, together with the volatility, to find the underlying distribution.

Using the weighted mean of the Gaussian mixtures is the most intuit of the two methods, it yet has its limitations. The only information given is the mean of the Gaussian mixture, which does not in a clear way describe the form of the underlying distribution. When implementing the MC simulation this problem was to be taken care of. The probability mass has been simulated, making it possible to unveil the complete underlying distribution, with mean, standard deviation and probability mass. By gaining this information one can easy define a level of certainty that should be used on the trading signal. Throughout the empirical studies it has been found that the hit ratio increases up to a certainty level of 55 percent. This is not equivalent to an increased payoff though. The reason why the hit ratio does not increase further when the certainty level is increased, which would be the most intuitive behavior, is that the model does not make estimations that are accurate enough.

The findings in this master's thesis has shown that the filtering process is too strict in some sense, filtering out profitable trades as well as nonprofitable. This is in line with the results found in section 2.3.1. The underlying cause is the same as just mentioned; due to the inaccuracy of the model, filtering is incorrectly executed.

In general, the use of MC simulation has given lower rates of return and hit ratios though. One possible and also probable answer to this is the way the mean for tomorrow has been simulated. Because this trading signal do not directly take the mean in consideration, but instead the probability mass, there have certainly been a information loss during the simulation process. A Brownian motion has been used for the simulation of the predicted mean. The interesting part in this case is to project the two underlying distributions, not to simulate the future development of EURUSD, which makes the Brownian motion applicable. Still the simulation might have added or removed skewness or kurtosis to the distributions, changing the probability mass for the Gaussian mixture. As many as 100 000 simulations has been used, trying to minimize the bias, but still there is one present.

Therefore one can say that both positive and negative results has been found for the trading signal using MC simulation. Positive results such as the simulated probability mass, and negative such as the information loss when simulating the underlying distributions. Since the model performance is worsen when using MC it should not be used if the certainty level is not larger than 50 percent.

5.5 Optimal Circumstances

The results given in chapter 4 does not show that there is an incorrect bias between the model's performance in different market characteristics, such as increasing or decreasing trends. One can on the other hand see differences between periods with and without clear trends — the models seems to have a hard time coping with changes, as the one just before day 700 in the training period. This result is not shown in every trajectory, but in many of them generated by the continuous model. Therefore it might be hard to do a more deep going analysis of this phenomena. Similar results has been found in previous works though, when HMMs has been used for prediction of movements in equity indices, such as S&P 500.

This phenomena has also been seen in the transition matrices. When these type of changes appear, the probabilities do vary a lot and quickly before they settle according to the new pattern.

5.6 State Equivalence

The states in an ordinary Markov chain corresponds to events observable in the world, as one could see in chapter 2, e.g. sunny, cloudy and rainy, and no further interpretation is called for. But when it comes to a HMM it is more difficult since the states are hidden. But as mentioned, in both chapter 2 and 3, the states could correspond to different regimes, e.g. high or low volatility, or strategies. But the pattern of the during testing generated Viterbi paths, does not point in any of these directions. A time period, where the model would stay in one state for several days or weeks, could be followed by one where the model would constantly switch state. It is therefore not possible to tie the states to something observable, and one simply have to satisfy with that they corresponds to some unknown underlying source of information that is linked to the EURUSD cross and the features.

5.7 Risk Assessment

In section 4.3 VaR and MDD was calculated for the best continuous and discrete model set ups. The measures shows that the continuous model contains a higher risk than the discrete, both having a higher average VaR as well as MDD. The biggest difference lies within the MDD though. Considering the continuous case one can see that the cumulative MDD is 31.5 percent of the total return during the whole back testing period. For the discrete model it is only 17.8 percent. This indicates that the risk for sustainable drawdowns is higher using the continuous HMM framework for forecasting movements in EURUSD, when using MDD as a measure of risk. The created benchmark index has a MDD compared to the total rate of return which lies between the HMMs' values.

At the same time as the risk is higher using MDD as a risk measure for the continuous model, one have to bear in mind that the potential for better overall return is more likely as well. Looking at the Sharpe ratio one can see that it is higher for the continuous model, which indicates that the rate of return in comparison to the risk taken, if risk is measured as volatility instead of MDD, is higher as well.

According to the high level of risk it is important to find ways in which it can be decreased, especially when using the continuous HMM. This can e.g. be done through diversifying the investment by using other, uncorrelated, strategies as a complement to the one presented in this master's thesis. In this way one could decrease the risk of large MDDs as well as high volatility for the total portfolio of investments.

5.8 Log-likelihood Sequence Convergence

From the observations made in section 4.4, three factors which influence the results have been distinguished when testing different set ups:

- Using a discrete or a continuous model.
- Using features or not.
- Length of the time window.

The effect that the random numbers have on the test result have to be eliminated to get a reliable testing in the out-of-sample period. For the discrete model using only the cross as input data, one is able to do this since all log-likelihood sequences converge in the training period. Except from modifying the model with respect to the three factors above, two ways of incorporating with non-convergence is available:

- Extension of the training period.
- Using initial estimates for the HMM parameters.

Due to the somewhat limited amount of data, it is not possible to largen the back-testing period to see where, or if, all sequences converge. It is reasonable to believe that for the discrete multivariate case, full convergence will take place further ahead in time like for the single-variate. But for the continuous case, it is more difficult to say anything about the log-likelihood sequences. Since two sequences could converge and then later on diverge, it seems rather unlikely that they ever would fully converge. Something else that speaks for this is the implemented K-means clustering, that was applied in the hope that it would steer the initial model parameters, regardless of the random initiation, towards one single set of model parameters, and thus evading the problem of sparse training data. Since the number of paths for the log-likelihood sequences were the same as before and full convergence non-existent, although the purpose was to go round this problem with the K-means clustering, it could indicate that convergence never would take place. Another explanation though could be that the segmentation method was not solid enough to generate initial estimates that was good enough.

To summarize why conversion or non-conversion is found, it is dependent on the model complexity. Looking at the three factors stated above, one have on one end a discrete single-variate model and on the other a continuous multivariate model. With an increased complexity, one get a more difficult optimization problem with more local maximums on the analyzed surface. Hence the number of possible paths between different optimums increases as well. And if the paths never were to converge for the continuous model, then the initialization of λ would be just as crucial as described in previous chapters. But not due to sparse training data, but due to the fact that different log-likelihood sequences generates models that returns different. If one is not able to find a rigid method for initiation, it is very important that one back-tests the model with many different initial parameter values to find a solution that seems to work for close-by historical time periods.

Chapter 6

Conclusions

The objective of this final chapter is to summarize our findings on the use of HMM as a FX prediction tool. The questions that were stated in the beginning are once again addressed and some final remarks on future research rounds up the chapter.

6.1 Too Many Factors Brings Instability

The hidden Markov model framework has previously been applied with great success in finance but never on foreign exchange data, at least to our knowledge. The results from our investigation has been somewhat mixed — for some settings we have been able to generate payoffs with accompanying Sharpe ratios well above the one for our created currency index, but for some cases the predictor been close to what looks like a random generator. To summarize the analysis carried out in chapter 5, we note that there are many factors that have to be considered when setting up a model, e.g. window length and number of states. This, together with the fact that small model adjustments sometimes leads to varying results, indicates a high instability of the model.

If this is due to the difficulties connected with the non-stationarity of data or some other aspect of FX data that complicates the time series analysis is hard to say. One of the conclusions we however draw is that a continuous model is preferable to the discrete. The major reason for this is due to the characteristic of the input data plus the more flexible structure associated with the continuous model. Each data point contains more information, which of course always is desirable, and furthermore, using a continuous model opens up for more advanced data handling, e.g. Monte Carlo simulation which we implemented. But further development is required as well as further testing. Something that was of special interest when we started off, was how the Gaussian mixture model and exponentially weighted expectation maximization algorithm was to affect results. The first showed to be essential for good estimates and a well functioning prediction capability, whereas the other did not act as an enhancement to the model.

One question that we asked ourself in the introduction was: *should Nordea use hidden Markov models as a strategy for algorithmic trading on foreign exchange data?* The answer to this is simple: no. At least not in this form. Even if some generated higher Sharpe ratios than the index, the lack of stability is too extensive. If, however, one of the models, generating a good Sharpe ratio in combination with a low P-value, was to be used, our recommendation is that it should be so in combination with other, uncorrelated, strategies to decrease the risks.

Throughout the investigation a set of delimitations has been considered. This has, most certainly, effected the conclusions drawn from this master's thesis. Most of all it effects the possibility of generalizing the conclusions, making them statable for every currency pair and time frame. Since our focus has been on EURUSD during a fixed period of time with one set of features the results should be seen as exclusive for this set up.

With this we conclude our evaluation of hidden Markov models as a tool for algorithmic trading on foreign exchange data. In the next section we will present some interesting topics that can be further investigated in order to create more stable and reliable models based on the framework of HMMs.

6.2 Further Research Areas

When it comes to future work it is first and foremost important to get a solid model which gives reliable predictions. Having obtained this, one can extend the model with other applications that can optimize the portfolio payoff.

Under *model improvements* we sort those things that can actually make the FX movements prediction more accurate. As we have seen in this master's thesis, the HMM framework seems to perform very well for some settings and time periods. But the instability needs to be addressed, and possible solutions might lie in the two suggestions below.

Substituting the ML method: We have in our tests seen that the performance during the test period, for the discrete case as well as the continuous is highly dependent on the initiated random numbers. The weakness lies within the way of using ML as an estimation method for finding the optimal set of model parameters via maximizing the log-likelihood. When using ML one assumes that the training period is large enough to provide robust estimates. We have during our test session seen that the continuous model do need a lot more training data than the discrete model. To find a way around the dependence of historical data one could use Maximum a Posteriori (MAP), mentioned in section 2.2.5, which could be a more efficient method for parameter estimation. The MAP estimation framework provides a way of incorporating prior information in the training process, which is particularly useful when dealing with problems posed by lack of training data. MAP has been applied successfully to acoustic modeling in automatic speech recognition when using a continuous HMM with GMMs as emission probabilities. It has also been proved helpful when smoothing parameters and adopting models in the discrete case, when using a discrete set of emission probabilities.

Improving initial estimates: As we noted, when looking at convergence for the log-likelihood paths, it is of utter importance that good initial parameter estimations are obtained. The implemented K-means clustering did not work properly, so the problems withstands. Other proposed ways of initiating the parameters includes manual segmentation of the observation sequence into states, which however assumes that one can tie certain observations to certain states, and maximum likelihood segmentation of observations with averaging. If these, or other methods not known to us, was to work, this would most probably increase the functionality of the HMM.

When it comes to *extensions to the HMM*, two methods could be tried for optimizing the overall return.

Portfolio of HMMs: Instead of just trading on the EURUSD cross, one could include other crosses, such as the EURSEK or EURNOK. Since different currency crosses might depend on different causes and have different trend patterns, one would have to use several HMMs, one for each cross. These could then treat different features while working under completely different model set ups. And when working with a portfolio of investments, one could also make use of stochastic programming and utility functions. This would open up for better risk management, where trading decisions could be made with respect to the risk aversion of the trader, and portfolio diversifying, eliminating non-systematic risk as mentioned in section 5.7.

Using leverage: As briefly described in section 2.3.1, one can take advantage of leverage when using GMM. Back-testing with different thresholds, one can obtain substantially larger profits. But the risk of large drawdowns due to too few trades has also increased. Using leverage could furthermore be a part of the aforementioned HMM portfolio, where the risk thinking is more incorporated.

Bibliography

[1] Dasgupta, S., (1999), *Learning Mixtures of Gaussians* Proceedings of Symposium on Foundations of Computer Science (FOCS).

[2] Dempster, M.A.H. et al., (2001), Computational Learning Techniques for Intraday FX Trading Using Popular Technical Indicators, *IEEE Transactions on Neural Networks*, **12**, 744-754.

[3] Eriksson, S. and Roding, C., (2007), *Algorithmic Trading Uncovered - Impacts on an Electronic Exchange of Increasing Automation in Futures Trading*, Royal Institute of Technology, Stockholm.

[4] Evans, M., (2006) *Foreign Exchange Market Microstructure*, Georgetown University and NBER.

[5] Giridhar, (2004), *Market Microstructure*, Wipro Technologies.

[6] Hafeez, B., (2007), *Deutsche Bank - Benchmarking Currencies: The Deutsche Bank Currency Returns (DBCR) Index*, Research Paper from Deutsche Bank's FX Strategy Group.

[7] Gauvain, J. and Lee, C., (1994) Maximum a Posteriori estimation for Multivariate Gaussian Mixture Observations of Markov Chains, *IEEE Transactions on Speech and Audio Processing*, **2**, 291-298.

[8] Harris, L., (2002), *Trading and Exchanges: Market Microstructure for Practioners*, Oxford University Press.

[9] Hassan, R., Nath, B. and Kirley, M., (2007), A Fusion Model of HMM, ANN and GA for Stock Market Forcasting, *Expert Systems with Applications*, **33**, 171-180.

[10] Hassan, R. and Nath, B., (2005), *Stock Market Forecasting Using Hidden Markov Model: A New Approach*, Proceedings of the 2005 5th International Conference on Intelligent Systems Design and Applications.

[11] Hendershott, T., (2003), *Electronic Trading in Financial Markets*, IEEE Computer Science.

[12] Hull, J.C., (2006), *Options, Futures, and Other Derivatives, 6th Edition*, Pearson Prentice Hall, London.

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

[14] Landén, C., (2000), Bond Pricing in a Hidden Markov Model of the Short Rate, *Finance and Stochastics*, **4**, 371-389.

[15] Lindemann, A. and Dunis, C.L., (2003), *Probability Distribution, Trading Strategies and Leverage: An Application of Gaussian Mixture Models*, Liverpool Business School, CIBEF and School of Computing and Mathematical Sciences, Liverpool John Moores University.

[16] Mamon, R.S. and Elliott, R.J., (2007), *Hidden Markov Models in Finance, 1st Edition*, Springer Science Business Media.

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

[18] Yingjian, Z., (2004), *Prediction of Financial Time Series with Hidden Markov Models*, Simon Fraiser University, Burnaby.

[19] Bank of International Settlement, (2007), *Triennial Central Bank Survey - Foreign exchange and derivatives market activity in 2007*.

[20] Bank of International Settlement, (2004), *BIS Quarterly Review – International banking and financial market developments*.

[21] Dictionary (Algorithm), www.bartleby.com, 01/11/2007.

[22] FX Poll, www.euromoney.com/article.asp?PositionID=&ArticleID=1039514, 14/09/2007.

[23] Maximum drawdown, www.risk.net, 11/12/2007.

LINKÖPING UNIVERSITY
ELECTRONIC PRESS

LINKÖPINGS UNIVERSITET