Algorithmic evaluation of Parameter Estimation for Hidden Markov Models in Finance

LINUS LAURI

Master of Science Thesis
Stockholm, Sweden 2014

Algorithmic evaluation of Parameter Estimation for Hidden Markov Models in Finance

LINUS LAURI

Master's Thesis in Mathematical Statistics (30 ECTS credits)
Master Programme in Mathematics (120 credits)
Supervisor at KTH was Boualem Djehiche
Examiner was Boualem Djehiche

TRITA-MAT-E 2014:12
ISRN-KTH/MAT/E--14/12--SE

Royal Institute of Technology
School of Engineering Sciences

KTH SCI
SE-100 44 Stockholm, Sweden

URL: www.kth.se/sci

Abstract

Modeling financial time series is of great importance for being successful within the financial market. Hidden Markov Models is a great way to include the regime shifting nature of financial data. This thesis will focus on getting an in depth knowledge of Hidden Markov Models in general and specifically the parameter estimation of the models. The objective will be to evaluate if and how financial data can be fitted nicely with the model. The subject was requested by Nordea Markets with the purpose of gaining knowledge of HMM's for an eventual implementation of the theory by their index development group. The research chiefly consists of evaluating the algorithmic behavior of estimating model parameters. HMM's proved to be a good approach of modeling financial data, since much of the time series had properties that supported a regime shifting approach. The most important factor for an effective algorithm is the number of states, easily explained as the distinguishable clusters of values. The suggested algorithm of continuously modeling financial data is by doing an extensive monthly calculation of starting parameters that are used daily in a less time consuming usage of the EM-algorithm.

Keywords: Hidden Markov Models, Parameter Estimation, Expectation Maximization, Direct Numerical Maximization

Contents

Chapter 1

Introduction

1.1 Introduction

A common feature of financial time series is that the relations and patterns within the series change from time to time. By this, it is hard to evaluate or infer patterns and predictions by models only relating data in a single regime. For example, one might have to take the different aspect of the economic cycle in mind when modeling time series related to finance. Volatility can be very different in different states of the economy, e.g. recession or high growth. Hence, financial time series analysis by using a regime-switching model like hidden Markov model is a quite intuitive way to go.

The applications of Hidden Markov Models (HMM's) are especially known in signal processing like automatic speech (Rabiner, 1989) and face recognition (Park & Lee, 1998) and biological sequence analysis (Churchill, 1992). Although the usage of the HMM theories haven't been as widespread in financial time series analysis and econometrics, the regime shifting approach has got an increasing amount of interest past years.

As the name suggests the Markov behavior in an HMM is hidden. By hidden it's meant that the Markov chain is not by itself retrievable but instead masked with additional white noise. In this sense there is no way of surely knowing the path of the underlying Markov chain, although several algorithmic approaches can be used to evaluate the most likely path. The two major approaches of parameter estimation by maximum likelihood for HMM's will be under scope in this thesis, direct numerical maximization and the expectation maximization algorithm.

The thesis is made on behalf of Nordea Markets' index development group, as a first step to evaluate the possibilities of using the theory of HMM's for an index. The aim is to get an in-depth understanding of HMMs and its applications to model financial time series. The main focus will be on investigating the optimization and performance of the parameter estimation of HMMs.

1.2 Delimitations

There are several different distribution mixtures to use when using the theory of HMM's. In this thesis we have limited the analyzed distributions only to involve normal HMM's with a discrete time step. Also, we only alter the standard deviations in the distributions, disregarding that also the mean could possibly be changing. By doing this we only make our regime analysis on different states of volatility and not of changes in trend regimes. The choice of only using discrete time steps for the HMM stems from the original problem of evaluating daily data.

Chapter 2

Theoretical background

To get a solid theoretical background of HMM's a few concepts will be presented in this chapter. First of, the most simplistic theory necessary for understanding HMM's, independent mixture distributions will be covered. Also, and obviously, part of the chapter will include some background on regular markov chains. From these concepts the theory of HMM's will be much easier to grasp.

2.1 Independent Mixture Models

Many time series have an unobserved heterogeneity i.e. within the time series there could be clusters of different kinds of data. Each of these clusters or groups comes from the same distribution, different from the distributions generating the other groups. A common way to simulate these overdispersed observations is to use independent mixture models. As suggested by the name the generating distribution consist of a mixture of independent distributions, or more specifically; conditional distributions. Since we only will deal with discrete mixtures regarding HMM's we only go over the concepts of discrete mixtures. The interested reader may find information on continuous mixtures in e.g. Böhning et al. (1999).

With the mixtures being discrete it's although important to notice that the conditional distributions may, and throughout this thesis will, be continuous. The model consists of some (discrete) distribution to generate what type of distribution the final draw will be made from. By this, the parameters of the conditional distributions in the second step will be generated by the sample space of the first distribution. Say for example that a three state mixture model of different normal distributions is under scope. The model is then characterized by the three random variables X_1, X_2 and X_3.

Random Variable	pdf
X_1	$f_1(x)$
X_2	$f_2(x)$
X_3	$f_3(x)$

Further, the mixture distribution will be given by some random function with a sample space of order two e.g.

$$Y = \begin{cases} 1 & \text{with probability } \pi_1 \\ 2 & \text{with probability } \pi_2 \\ 3 & \text{with probability } \pi_3 \end{cases}$$

$$\pi_1 + \pi_2 + \pi_3 = 1$$

Bulla (2006) exemplifies the procedure of a two-component distribution mixture by letting Y be the tossing of a coin to asses what random variable an observation is a realization of. Figure 2.1 explains this further by visualizing the process of a two state mixture model above. It's important to bear in mind that the realization of Y is generally not observable, it is only the observation generated by one of the random variable that can be observed.

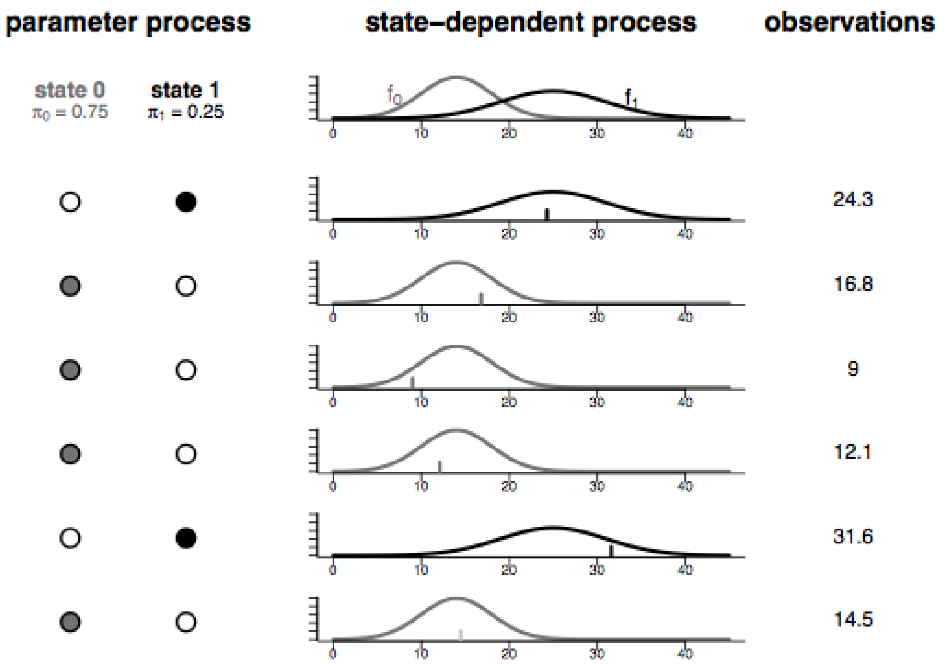


Figure 2.1: Visualization of independent mixture model

By the explanation of Y and X the probability density function of the mixture can easily be calculated.

$$f(x) = \pi_1 f_1(x) + \pi_2 f_2(x) + \pi_3 f_3(x)$$

As seen the extension to J components is quite straightforward. As for the three state case we let $\pi_i, i = \{1, 2, .., J\}$ be the probability of the discrete mixture. Each sample i will result in different distributions with f_1, f_3,.., f_J as their distribution functions.

$$f(x) = \sum_{i=1}^{J} \pi_i f_i(x)$$

More details on independent mixture models could be found in MacDonald and Zucchini (2009).

2.2 Markov chains

Roughly speaking, a Markov chain is a stochastic process where the most recent value at time t is the only relevant information for the future value at $t + 1$. That is, regardless of the process' history. There is a lot of literature available on the subject of Markov chains, why we will try to keep it somewhat brief in this section. A more general account is found in Norris (1998) or Parzen (1962).

First of, a comment about the usage of chain instead of process must be made. The name Markov chain is often used when dealing with a discrete-time Markov process with a discrete state space. Consider a sequence of discrete random variables $\{Y_t : t \in \mathbb{N}\}$. The state space of the sequence consist of $i = \{1, 2, .., J\}$ and is said to be a Markov chain if it satisfies

$$P(Y_{t+1} = y_{t+1} \mid Y_t = y_t, Y_{t-1} = y_{t-1}, .., Y_0 = y_0) = P(Y_{t+1} = y_{t+1} \mid Y_t = y_t)$$

for all $t \in \mathbb{N}$.

This is called the Markov property and can be regarded as a first relaxation of the assumption of independence [15]. As seen in Figure 2.2 the name comes from the fact that the random variables are only dependent of the prior value of the chain, which makes the model mathematically convenient.

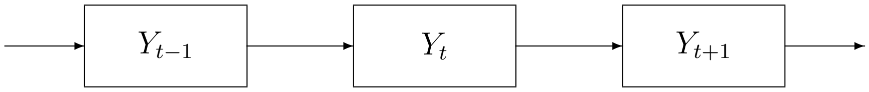


Figure 2.2: Markov chain structure

The probability of moving from one state to another is called the transition probability. These probabilities are presented in a $J \times J$ transition probability matrix (TPM) with elements $p_{ij} = P(Y_{t+1} = j \mid Y_t = i)$ and $\sum_{j=1}^{J} p_{ij} = 1$ for all $i \in \{1, 2, .., J\}$. If the TPM is independent of t we speak of the Markov chain as homogeneous. Thus, the TPM contains the one-step transition probabilities and short-term behavior of the Markov chain. For homogeneous Markov chains we have the Chapman-Kolmogorov equations that boils down to $\boldsymbol{\Gamma}(k) = \boldsymbol{\Gamma}(1)^k$. Thus the k-step transition probabilities could be explained as

$$\boldsymbol{\Gamma}(k) = \begin{pmatrix} p_{1,1}(k) & \cdots & p_{1,J}(k) \\ \vdots & \ddots & \vdots \\ p_{J,1}(k) & \cdots & p_{J,J}(k) \end{pmatrix}$$

Apart from the TPM the unconditional probabilities $P(Y_t = i)$ is often of great interest when it comes to Markov chains. These are the probability that the Markov chain is in a given state i at a given time t and are denoted by the vector.

$$\boldsymbol{u}(t) = (P(Y_t = 1), .., P(Y_t = J)), t \in \mathbb{N}$$

The first of these vector, the one connected to $t = 1$ is often expressed as the initial distribution $\boldsymbol{\pi}$ of the Markov chain. Also, it's worth nothing that all the unconditional probabilities can be calculated recursively by the initial distribution and the TPM for a homogeneous Markov chain.

$$u(1) = \boldsymbol{\pi}$$

$$u(t) = \boldsymbol{u}(t-1)\boldsymbol{\Gamma}(1)$$

By the TPM we know that if an element $p_{ij}(k)$ is non-zero the state j is accessible from state i in k steps. If the element $p_{ij}(k)$ is non zero for any k we write $i \to j$. Furthermore, if the element $p_{ji}(k)$ of the TPM is non zero for any k we have that $j \to i$. If both $i \to j$ and $j \to i$ we write $i \leftrightarrow j$ and says that the two states i and j communicates with each other. We call the Markov chain irreducible if all states i and j communicates with each other, i.e. $i \leftrightarrow j$ for all $i, j \in \{1, 2, .., J\}$.

A Markov chain that is both irreducible and homogeneous is said to be stationary with a stationary distribution δ, defined by

$$\boldsymbol{\delta} = \boldsymbol{\delta}\boldsymbol{\Gamma}$$
$$\boldsymbol{\delta}\mathbf{1}' = 1$$

Where the first of these express the stationarity and the second the fact that sigma is a probability distribution.

2.3 Hidden Markov Models

Sometimes it happens that the clusters or groups discussed in the section of independent mixture models are correlated to each other. In such cases the independent approach of modeling time series wouldn't be the best approach. By using an independent mixture model we would miss the dependence structure and a lot of the information available in the data. The simple independent case would lead to overdispersion when fitting the sample mean. A solution to this problem would be to relax the assumption of independent subgroups in the time series. An example of such a relaxation, and actually one of the most simplistic one, would be to use a Hidden Markov Model (HMM). The statement that this would be a simple way of allowing serial dependence stems from the fact that the usage of a Markov chain as the relation between the subgroups, or states, is quite mathematical convenient [15]. This by the definition of a Markov chain and that it fulfills the Markov property of only depending on the prior state. Further on in this section we will look into the basic of the HMM framework, for the interested reader MacDonald (1997) and Cappé et al. (2005) are suggested to cover a larger portion of the theory.

Cappé et al. (2005) explaines HMM's as a Markov chain observed in noise, which also coincide with the explanation above of using a Markov chain for the dependence in a mixture model. The Markov chain is underlying and unobserved, hence the hidden part, and it states are connected with different distributions generating the observations. The directed graph in Figure 2.3 gives a good visualization of the basic structure of a HMM and that the observations are serially dependent.

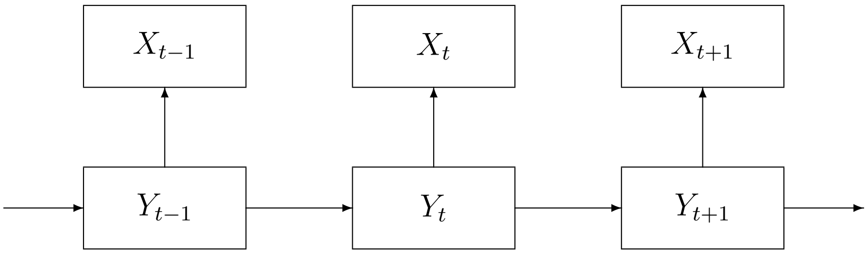


Figure 2.3: Hidden Markov Model structure

By using a HMM to model the dependent structure in time series is a good way to really simplify complex time series, since the Markov property is as mentioned before mathematically convenient. The hard part of this model is, as the name suggests, the hidden part of the model. As in the case of independent mixture models, the generating space is commonly unknown. Thus we only have the given observations but can't say for sure what random variable that it is a realization of. We remember the figure from Bulla (2006) that explained the generating process for independent mixture models above; Figure 2.4 is an extension that models the dependency structure in the observation generating process.

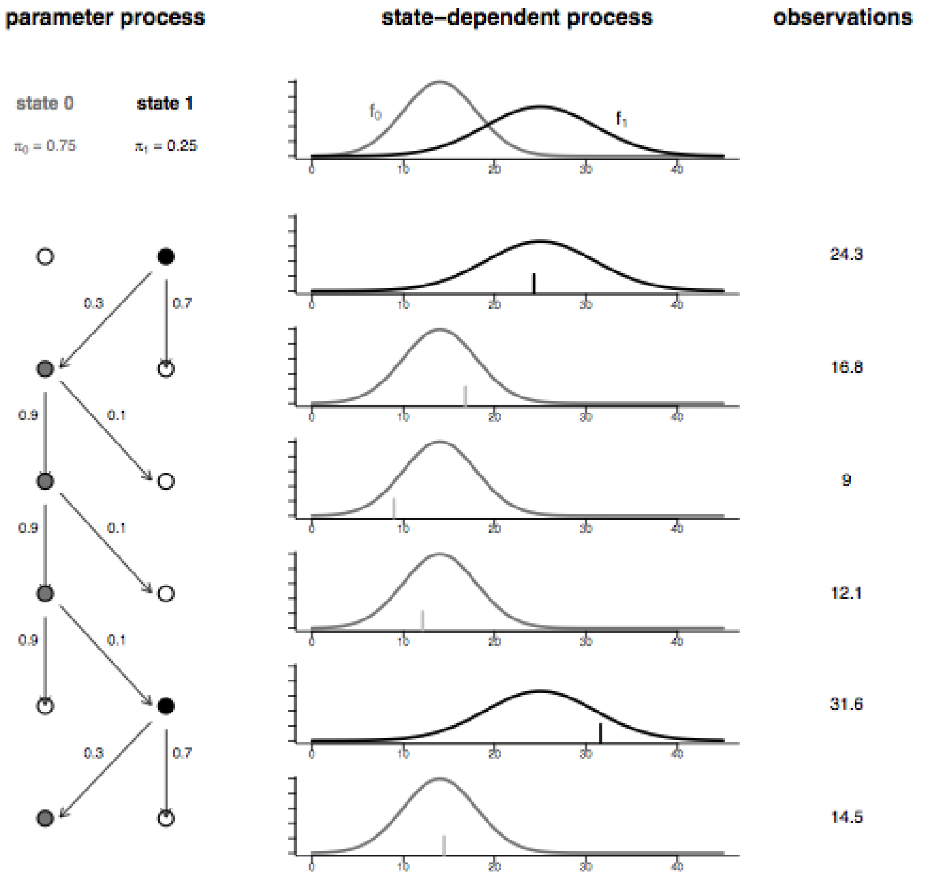


Figure 2.4: Visualization of independent mixture model

Going forward, we will have to introduce some denotations. Looking back to Figure 2.3 above $\{X_t\} = \{X_t, t = 1, 2, 3..\}$ denotes a sequence of observations and $\{Y_t\} = \{Y_t = 1, 2, 3..\}$ a Markov chain with states $i \in \{1, 2, .., J\}$. By the definition of a Markov chain the $\{Y_t\}$ follow the Markov property, i.e. $P(Y_{t+1} = y_{t+1} \mid Y_t = y_t, Y_{t-1} = y_{t-1}, .., Y_0 = y_0) = P(Y_{t+1} = y_{t+1} \mid Y_t = y_t)$. Although the relaxation of the independence structure is made for the states, explained as subgroups above, the state dependent observations are independent of previous observations. We have

$$P(X_{t+1} = x_{t+1} \mid X_t = x_t, .., X_0 = x_0, Y_t = y_t, .., Y_0 = y_0) = P(X_{t+1} = x_{t+1} \mid Y_t = y_t)$$

We call such a pair of processes $\{Y_t, X_t\}$ a J-state HMM. This somewhat explains the statement that a HMM is observed in noise, since it is a combination of two processes; a Markov chain $\{Y_t\}$ and some random state dependent noise $\{X_t\}$ on top of this Markov chain.

Many of the concepts from Markov chains translates into the world of HMM's. In Figure 2.4 we had a stationary distribution of $\boldsymbol{\sigma} = (0.75, 0.25)$ and a transition probability matrix of $\boldsymbol{\Gamma} = \left(\begin{smallmatrix} 0.9 & 0.1 \\ 0.3 & 0.7 \end{smallmatrix}\right)$. The probability density function of $\{X_t\}$ in state i of the Markov chain is given by

$$f_i(x) = P(X_t = x \mid Y_t = i) \tag{2.1}$$

These J distributions are referred to as the state-dependent distributions of the HMM [15]. In the case of a normal distributed HMM the state dependent distributions would be given by

$$f_i(x_t) = \frac{1}{\sqrt{2\pi\sigma_i^2}} e^{-\frac{(x_t-\mu_i)^2}{2\sigma_i^2}}$$

for $t \in \{1, 2, .., T\}$ and $i \in \{1, 2, .., J\}$.

Where the parameters μ_i and σ_i of the distribution function depends on the state $Y_t = \{1, 2, .., J\}$.

The probability of observing a certain value x at t for a HMM $\{Y_t, X_t\}$ is, obviously, largely related to the probability of observing a Markov chain $\{Y_t\}$ in state i at time t. The additional calculation is to add the probabilities of Function 2.1 and summarizing over all states. To sum up over all spaces is necessary since the value x at t could be a realization of all the different state-dependent random variables X_i.

$$P(X_t = x) = \sum_{i=1}^{J} P(Y_t = i) P(X_t = x \mid Y_t = i)$$

for all $t \in \{1, 2, .., T\}$

An important part of the theory of HMM's is the likelihood of the models parameters. Actually, this is one of the most crucial concepts of fitting a HMM to some time series and will be needed for the parameter estimation later in this chapter. Fortunately the likelihood function of a HMM is can be expressed explicitly and computable in a closed formula.

First of, the definition of the likelihood of the observations is the joint probability density function of $X_t = x_t$ for $t \in \{1, 2, .., T\}$.

$$L_T = P(X_1 = x_1, X_2 = x_2, .., X_T = x_T)$$

We remember that we have to sum over all possible states to cover all possible random variables that could result in the realization x_t. Hence, it holds that

$$L_T = \sum_{y_1, y_2, .., y_T = 1}^{J} P(X_1 = x_1, .., X_T = x_T, Y_1 = y_1, .., Y_T = y_T)$$

From equation (2.5) in MacDonald (2009) we know that

$$P(X_1 = x_1, X_2 = x_2, .., X_T = x_T) = P(Y_1) \prod_{k=2}^{T} P(Y_k \mid Y_{k-1}) \prod_{k=1}^{T} P(X_k \mid Y_k)$$

Further, let the $\boldsymbol{P}(x_t)$ be the conditional probabilities defined above as

$$\boldsymbol{P}(x_t) = \begin{pmatrix} f_1(x_t) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & f_J(x_t) \end{pmatrix}$$

Also, we remember that the rows of the TPM $\sum_{j=1}^{J} p_{ij}$ are the probabilities of jumping out of state i for some $i \in \{1, 2..J\}$ on the time step from $t-1$ to t. Along with the initial probability vector $\pi = (\pi_1, \pi_2, .., \pi_J)$ the likelihood function can be expressed by matrix notations. Since this is a function of the parameters of the HMM we will let θ be the set of all model parameters of the HMM. That is, in the case of a Normal HMM, the set of all elements of the transition probability matrix p_{ij}, the initial state distribution π_i and the parameters of the state-dependent distributions of $X_t \sim N(\mu_i, \sigma_i^2)$. Thus, we get the expression of the likelihood function of the observations as

$$L_T(\theta) = P(X_1 = x_1, .., X_T = x_T) = \boldsymbol{\pi} \boldsymbol{P}(x_1) \boldsymbol{\Gamma} \boldsymbol{P}(x_2) ... \boldsymbol{\Gamma} \boldsymbol{P}(x_T) \mathbf{1}^t$$

Although the HMM concept is quite a theoretical one, the states of the hidden Markov chain can often be interpreted easily. In such an interpretation the states is often thought of as different regimes. In the case of this thesis we look at financial time series and more specifically on the possibility of data coming from distributions with different standard deviation. This can be interpreted as evaluating different volatility regimes in the financial world.

Chapter 3

Parameter estimation

This chapter is doveted to focus on the Maximum-Likelihood parameter estimation of HMM's. That is, both the expectation maximization algorithm and direct numerical maximization.

The main usage of HMM's in finance and the essence of this thesis is to fit a HMM to a time series of observations. To do this there are several parameters of the HMM that has to be estimated in some way. The most common parameter estimation is done by maximizing the likelihood function for the HMM. This is done by either using direct numerical maximization (DNM) or the expectation maximization algorithm (EM). None of these methods is superior to the other but it is very common that only one of the algorithms is used in specific studies. In Bulla (2006) a thorough comparison of the two approaches are made which will, along with the mathematical depth in Cappé et al. (2005), build up much of this section. The more mathematical experienced reader may see these two sources, especially the latter, for a greater depth on maximum likelihood parameter estimation.

3.1 Direct numerical maximization

Direct numerical maximization comes from the family of gradient-based methods. These methods constitutes of directly calculating the parameters for maximum likelihood by zero equating the derivative of the likelihood function in respect to the parameters in $L(\theta)$.

Remember that we in the previous section ended up in the function

$$L_T(\theta) = \boldsymbol{\pi}\boldsymbol{P}(x_1)\boldsymbol{\Gamma}\boldsymbol{P}(x_2)...\boldsymbol{\Gamma}\boldsymbol{P}(x_T)\mathbf{1}^t \tag{3.1}$$

Through this, it is easy to see that the likelihood for $L_1(\theta)$ can be retrieved by taking the product of only the initial distribution π and the matrix of state-dependent probabilities $\boldsymbol{P}(x_1)$ for observations at $t = 1$. Recursively we can build up a likelihood function for all observations up to T, which will end up in Function 2.1. We introduce α_t as the forward probabilities, the usage of the name forward probabilities will become evident in later sections. The recursion equations is as follows

$$\boldsymbol{\alpha}_1 = \boldsymbol{\pi}\boldsymbol{P}(x_1)$$
$$\boldsymbol{\alpha}_t = \boldsymbol{\alpha}_{t-1}\boldsymbol{\Gamma}\boldsymbol{P}(x_t), t = \{2, 3, .., T\}$$

If we define $\boldsymbol{B}_t$ as $\boldsymbol{T}\boldsymbol{P}(x_t)$ we have that $L_T(\theta) = \boldsymbol{\pi}\boldsymbol{B}_1...\boldsymbol{B}_T\mathbf{1}^t$. By this we also have that $L_T(\theta) = \boldsymbol{\alpha}_T$.

Further we introduce a scalar weight w_t of the forward probability α at t. That is, $w_t = \boldsymbol{\alpha}_t\mathbf{1}^t$. Using these weights we rescale the forward probabilities to ϕ_t by

$$\boldsymbol{\phi}_t = \boldsymbol{\alpha}_t\mathbf{1}^t$$

For $t = \{0, 1, ..., T\}$.

Using this scaling we can evaluate the likelihood function L_T by using the starting equation

$$\boldsymbol{\phi}_0 = \frac{\boldsymbol{\alpha}_0}{w_0} = \frac{\boldsymbol{\pi}}{\boldsymbol{\pi}\mathbf{1}^t} = \boldsymbol{\pi} \tag{3.2}$$

We easily see that the $\boldsymbol{\phi}_t$ for $t = \{1, 2, .., T\}$ are given through using $\boldsymbol{\alpha}_t$ by

$$\boldsymbol{\phi}_t = \frac{\boldsymbol{\alpha}_t}{w_t} = \frac{\boldsymbol{\alpha}_{t-1}\boldsymbol{B}_t}{w_t} = \frac{w_{t-1}}{w_t}\boldsymbol{\phi}_{t-1}\boldsymbol{B}_t \tag{3.3}$$

Hence we have a reccursive evaluation of the likelihood function L_T by using 3.3 with starting equation given by 3.2. We can then express the likelihood function by using the weights w_t.

$$L_T = \prod_{t-1}^{T} \frac{w_t}{w_{t-1}}$$

Further, by taking the logarithm of the likelihood it can be expressed as a sum of the weights just defined.

$$\ln L(\theta) = \sum_{t=1}^{T} \ln \left(\frac{w_t}{w_{t-1}} \right)$$

The ratio used in each sumation is given as $w_t/w_{t-1} = \boldsymbol{\phi}_{t-1}\boldsymbol{B}_t\mathbf{1}^t$. The logarithmic likelihood can thus be evaluated by using a reccursive algorithm. We start of by using the equations $\ln L_T = 0$ and $\boldsymbol{\phi}_0 = \boldsymbol{\pi}_0$ as starting values for the reccursive algorithm below.

$$v_t = \boldsymbol{\phi}_{t-1}\boldsymbol{B}_t$$

$$u_t = \boldsymbol{v}_t\mathbf{1}^t$$

$$\ln L_t = \ln L_{t-1} + \ln u_t$$

$$\boldsymbol{\phi}_t = \boldsymbol{v}_t/u_t$$

This is repeted for $t = \{1, 2, ..., T\}$. This recursive algorithm will express the logarithmic likelihood function. To maximize it, some numerical maximization procedure is needed. For the gradient-based approach there are a few methods available. The easiest is the steepest ascent algorithm that only takes the first derivative of the log likelihood into account. The model parameters are updated by adding a multiple of the gradient to the existing set of parameters. This way is called walking in the search direction, the update formula is given by

$$\theta^{i+1} = \theta^i + \gamma_i \nabla_\theta \ell(\theta^i)$$

In each step the multiplier gamma needs to be updated to satisfy the fact that the sequence is increasing. Luenberger (1984) provides an account of how to update gamma for the multiplier to be globally convergent. The updating formula is done by

$$\gamma_i = \arg\max_{\gamma \geq 0} \ell[\theta^i + \gamma \nabla_\theta \ell(\theta^i)]$$

There are though several setbacks in using the steepest ascent algorithm, especially for large models with several parameters. The main disadvantage is that the algorithm is only linearly convergent for the set of parameters theta. A better approach is to use some kind of second-order method, like the Newton algorithm. Here we use the Hessian H that is defined as the second derivative $H(\theta^i) = \nabla_\theta^2 \ell(\theta^i)$. The updating algorithm of the parameters theta is defined as

$$\theta^{i+1} = \theta^i + H^{-1}(\theta^i)\nabla_\theta \ell(\theta^i)$$

which comes from second order Taylor approximation

$$\ell(\theta) \approx \ell(\theta') + \nabla \ell(\theta')(\theta - \theta') + \frac{1}{2}(\theta - \theta')' H(\theta')(\theta - \theta')$$

Implementation of this maximization can be done by using the optimization toolbox in MatLab.

3.2 Expectation Maximization

Probably the most used algorithm for parameter estimations of HMM's is the Expectation Maximization algorithm (EM). This algorithm constitutes of two steps that are iterated until the parameters converge to a maximized likelihood. For HMM's the algorithm is also known as the Baum-Welch algorithm. We will start of by introducing the forward-backward algorithm that is crucial for the EM algorithm. We follow the setup used in MacDonald and Zucchini (2009).

3.2.1 Forward-Backward algorithm

The forward-backward algorithm is in fact two separate algorithms for calculating forward and backward probabilities. The forward probability $\alpha_t(i)$ are defined as the joint probability of being in state i at time t and that the observations x_n for $n \in \{1, 2, ..T\}$ are retrieved. Actually, we've already defined the forward probability above. Although, we didn't fully state the connection to the joint probability of $P(X_t = x_t, Y_t = i)$. We state the probabilities again below as a product of matrices.

$$\boldsymbol{\alpha} = \boldsymbol{\pi}\boldsymbol{P}(x_1)\boldsymbol{\Gamma}\boldsymbol{P}(x_2)...\boldsymbol{\Gamma}\boldsymbol{P}(x_T) = \boldsymbol{\pi}\boldsymbol{P}(x_1)\prod_{s=2}^{T}\boldsymbol{\Gamma}\boldsymbol{P}(x_s)$$

This is given by the usage of the recursion

$$\boldsymbol{\alpha}_{t+1} = \boldsymbol{\alpha}_t\boldsymbol{\Gamma}\boldsymbol{P}(x_{t+1})$$

which is, in scalar form,

$$\alpha_{t+1}(j) = \left(\sum_{i=1}^{J} \alpha_t(i) p_{ij} \right) p_j(x_{t+1})$$

Where we remember p_{ij} and f_j, respectively, as the elements of the transition probability matrix and the conditional probability of an observation given a certain state j. That the forward probability is in fact $\alpha_t(i) = P(X_1 = x_1, X_2 = x_2, .., X_t = x_t, Y_t = i)$ can be shown by realizing that the

$$\boldsymbol{\alpha}_1 = \boldsymbol{\pi}\boldsymbol{\Gamma}$$

And in scalar form

$$\alpha_1(i) = \pi_i f_i(x_1) = P(Y_1 = i)P(X_1 = x_1 \mid Y_1 = i)$$

Thus, α_1 is in fact $P(X_1 = x_1, Y_1 = i)$. Recursively we can now see that if $\alpha_t = P(X_1 = x_1, X_2 = x_2, .., X_t = x_t, Y_t = i)$ holds for some t it also holds for $t + 1$ by

$$\begin{aligned} \alpha_{t+1}(j) = \sum_{i=1}^{J} \alpha_t(i) p_{ij} f_j(x_{t+1}) = \sum_{i} & P(X_1 = x_1, .., X_t = x_t, Y_t = i) \\ & \times P(Y_{t+1} = j \mid Y_t = i) P(X_{t+1} = x_{t+1} \mid Y_{t+1} = j) \\ = \sum_{i} & P(X_1 = x_1, .., X_{t+1} = x_{t+1}, Y_t = i, Y_{t+1} = j) \\ & = P(X_1 = x_1, .., X_{t+1} = x_{t+1}, Y_{t+1} = i) \end{aligned}$$

The backward probabilities is the conditional probabilities that we observe the observations $\{x_k\}$ for $k = (t+1, t+2, ..)$ given that the underlying Markov chain is in state i at time t, that is

$$\beta_t(i) = P(X_{t+1} = x_{t+1}, X_{t+2} = x_{t+2}, .., X_T = x_T \mid Y_t = i)$$

In matrix notation it is given as $\boldsymbol{\beta}_t = \boldsymbol{\Gamma}\boldsymbol{P}(x_{t+1})\boldsymbol{\beta}_{t+1}$. The proof can be found in chapter 4 of MacDonald and Zucchini (2009). These probabilities can be used to calculate the likelihood function given as $L_t = P(X_1 = x_1, X_2 = x_2, .., X_t = x_t, Y_t = i)$. Since we have the definitions of the forward and backward probability the product of these to results in the following equation in scalar form, where we have for ease of notations defined $P(\boldsymbol{X}_{(1)}^{(t)} = \boldsymbol{x}_{(1)}^{(t)}) = P(X_1 = x_1, .., X_t = x_t)$

$$\begin{aligned}\alpha_t(i)\beta_t(i) &= P(Y_t = i)P(\boldsymbol{X}_{(1)}^{(t)} = \boldsymbol{x}_{(1)}^{(t)} \mid Y_t = i)P(\boldsymbol{X}_{(t+1)}^{(T)} = \boldsymbol{x}_{(t+1)}^{(T)} \mid Y_t = i) \\ &= P(X_1 = x_1, X_2 = x_2, .., X_t = x_t, Y_t = i)\end{aligned}$$

Further, we have that the conditional probability of being in state i given all available observations as

$$P(Y_t = i \mid \boldsymbol{X}_{(1)}^{(T)} = \boldsymbol{x}_{(1)}^{(T)}) = \alpha_t(i)\beta_t(i)/L_T$$

Finally we need the joint conditional probability of $Y_{t-1} = i$ and $Y_t = j$ given all available observations. Using the forward and backward probabilities, the transition probabilities p_{ij} and the state-dependent probabilities g_i we have

$$\begin{aligned}\alpha_{t-1}(i)p_{ij}f_j(x_t)\beta_t(j)/L_T &= \alpha_{t-1}(i)p_{ij}\Big(P(X_t = x_t \mid Y_t = j)P(\boldsymbol{X}_{(t+1)}^{(T)} = \boldsymbol{x}_{(t+1)}^{(T)} \mid Y_t = j)\Big)/L_T \\ &= P(\boldsymbol{X}_{(1)}^{(t-1)} = \boldsymbol{x}_{(1)}^{(t-1)}, Y_{t-1} = i)P(Y_t = j \mid Y_{t-1} = i)P(\boldsymbol{X}_{(t)}^{(T)} = \boldsymbol{x}_{(t)}^{(T)}, Y_t = j)/L_T \\ & P(Y_{t-1} = i, Y_t = j \mid \boldsymbol{X}_{(1)}^{(T)} = \boldsymbol{x}_{(1)}^{(T)})\end{aligned}$$

3.2.2 Expectation-Maximization Algorithm

We now posses the right tools to define the EM algorithm. The EM algorithm is an iterative way of estimating and maximizing the likelihood for a process where some data is missing. Since the Markov chain is hidden for HMM's it is quite reasonable to use the EM algorithm and treat all states as missing data. The EM uses the fact that it is possible to calculate the full data log likelihood even though the likelihood of only the observed data isn't retrievable. As mentioned before the algorithm constitutes of two steps. First of, the expectation (E) step calculates the expectation of the missing data conditional of the observations and the estimate of the parameter vector, $\boldsymbol{\theta}$. After this, the maximization (M) step maximizes the log likelihood for the complete model with respect to $\boldsymbol{\theta}$. Repetitions of these two steps are done until the convergence criterion is reached, resulting in a maximum likelihood vector $\boldsymbol{\theta}$.

We can express the log likelihood of the HMM as

$$\begin{aligned}\ln\left(P(X_{(1)}^{(T)} = x_{(1)}^{(T)}, Y_{(1)}^{(T)} = y_{(1)}^{(T)})\right) &= \ln\left(\pi_{y_1}\prod_{t=2}^{T} p_{y_{t-1},y_t} + \prod_{t=1}^{T} f_{y_t}(x_t)\right)\\ &\ln\sigma_{y_1} + \sum_{t=2}^{T}\ln p_{y_{t-1},y_t} + \sum_{t=1}^{T}\ln f_{y_t}(x_t)\end{aligned}$$

We can simplify this expression by using variables u_i and v_{ij}, defined as

$$\begin{aligned}u_j(t) = 1 \quad \text{iff.} \quad y_t = j \quad (t = 2, 3, .., T)\\ v_{ij}(t) = 1 \quad \text{iff.} \quad y_{t-1} = i \quad \text{and} \quad y_t = j \quad (t = 2, 3, .., T)\end{aligned}$$

We then get the log likelihood expression as

$$\begin{aligned}&\ln\left(P(\boldsymbol{X}_{(1)}^{(T)} = \boldsymbol{x}_{(1)}^{(T)}, \boldsymbol{Y}_{(1)}^{(T)} = \boldsymbol{y}_{(1)}^{(T)})\right)\\ &\qquad = \sum_{i=1}^{J} u_i(1)\ln\pi_i + \sum_{i=1}^{J}\sum_{j=1}^{J}\left(\sum_{t=2}^{T} v_{ij}(t)\right)\ln p_{ij}\\ &\qquad\quad + \sum_{i=1}^{J}\sum_{t=1}^{T} u_i(t)\ln f_i(x_t)\\ &\qquad = \text{term } 1 + \text{term } 2 + \text{term } 3\end{aligned} \tag{3.4}$$

The expression in Function 3.4 is then used for the E step by replacing $u_i(t)$ and $v_{ij}(t)$ by their conditional probabilities if the observations $\boldsymbol{x}_{(1)}^{(T)}$ are retrieved, $\hat{u}_i$ and $\hat{v}_{ij}$ defined as

$$\begin{aligned}\hat{u}_i(t) = P(Y_t = i \mid \boldsymbol{X}_{(1)}^{(T)} = \boldsymbol{x}_{(1)}^{(T)}) = \alpha_t(i)\beta_t(i)/L_T\\ \hat{v}_{ij}(t) = P(Y_{t-1} = i, Y_t = j \mid \boldsymbol{X}_{(1)}^{(T)} = \boldsymbol{x}_{(1)}^{(T)}) = \alpha_{t-1}(i)p_{ij}f_j(x_t)\beta_t(j)/L_T\end{aligned}$$

The full expression of the log likelihood can then be maximized and conveniently this is only a matter of maximizing the three independent terms in Function 3.4. That is the following equations need to be maximized.

1. $\sum_{i=1}^{J} u_i(1) \log \pi_i$ with respect to $\boldsymbol{\pi}$

2. $\sum_{i=1}^{J} \sum_{j=1}^{J} \left(\sum_{t=2}^{T} v_{ij}(t) \right) \log p_{ij}$ with respect to $\mathbf{\Gamma}$

3. $\sum_{i=1}^{J} \sum_{t=1}^{T} u_i(t) \log f_i(x_t)$ with respect to the parameters of f

Solving these three equations will yield the new parameters as

1. $\pi_i = \hat{u}_i(1)/ \sum_{i=1}^{J} \hat{u}_i(1) = \hat{u}_i(1)$

2. $p_{ij} = \sum_{t=2}^{T} \hat{v}_{ij}(t)/ \sum_{j=1}^{J} \left(\sum_{t=2}^{T} \hat{v}_{ij}(t) \right)$

The solution to the last term depends on the nature of the state dependent distribution g. For a normal HMM the parameters of the state dependent distribution are given by solving the third term above with the distributions g_i as $\mathrm{N}(\mu_i, \sigma_i^2)$. This gives the new parameters as

$$\hat{\mu}_i = \sum_{t=1}^{T} \hat{u}_i(t) x_t \Big/ \sum_{t=1}^{T} \hat{u}_i(t)$$
$$\hat{\sigma}_j^2 = \sum_{t=1}^{T} \hat{u}_i(t)(x_t - \hat{\mu}_i)^2 \Big/ \sum_{t=1}^{T} \hat{u}_i(t)$$

3.3 Comparison

Bulla (2006) states a few differences regarding the two algorithmic approaches for parameter estimations of HMM's. The main setback of the EM algorithm is that it, like the steepest ascent algorithm, it only has linear convergence when evaluating the parameters. Also for an EM algorithm both forward and backward probabilities are needed, although it is enough with only forward probabilities to evaluate and maximize the likelihood by DNM. Cappé et al. (2005) states a few other pros and cons of the different approaches.

Compared to the gradient-based methods the EM algorithm is easy to implement, since it doesn't needs to evaluate the hessian or gradient of the likelihood. With this said one can often use prebuilt generic optimization algorithms for the gradient-based algorithm implementation, which simplifies the procedure significantly. Another setback of the DNM is that it doesn't deal with parameter constraints in the same way that the EM algorithm does.

In the DNM these constraints have to be dealt with explicitly by reparameterization. Further, the most major setback is the fact that the EM algorithm is a lot more stable according to Bulla (2006). That is, the solution does lead to global maximum more frequently for the EM than for DNM algorithms.

3.4 Model validation

The speed of the algorithms and the number of calculations depends strongly on the number of states in the Markov chain of the HMM. Thus, choosing a good amount of states are crucial for a effective algorithm. Two important model valuation methods are Akaike (AIC) and Bayesian information criterion (BIC). These two evaluate the goodness of fit for the selected model. More specifically the two models punish the likelihood function by the number of states, locating the trade of between the increasing likelihood of adding more states and the increasing punish term. The two criterions are given below.

$$AIC = 2k - \ln(L)$$
$$BIC = k \ln(T) - 2\ln(L)$$

Where L is the likelihood, T the number of observations and k is the number of states. We know that $\boldsymbol{\theta} = (\boldsymbol{\Gamma}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\sigma})$. The transition probability matrix has $k^2 - k$ parameters since the last row is retrieved by the rest of the elements. Further we have one standard deviation, one mean value and a starting probability in each state. Hence, for a normal HMM we have that the number of parameters is $k = J^2 + 2J$.

Its important to notice that this only evaluates the model with respect to a change in the number of states and doesn't take into account of the full model in fact has a poor fit to the time series.

Chapter 4

Method

This chapter will cover the method and approach used when working on this thesis. Also, a section regarding the limitations of the study will be included.

4.1 Literature studies

The nature of the problem in this thesis demanded a solid base of knowledge regarding HMM's and Markov processes in general. Fully understanding the setup is crucial to be able to create any algorithm of value for the purpose of modeling any financial time series at all. The choice of literature regarding HMM's was somewhat straightforward after a few searches, at least when it comes to HMM's in general. Although the theories of HMM's are quite young there are quite a few outstanding authors. The books MacDonald and Zucchini (1997) and (2009) and Cappe et al. (2005) has been the three major sources of research. These have naturally led in to a few important and more specific research papers regarding HMM's.

The main disadvantage of these sources has been that many are quite general and aren't applied to finance specifically. Most examples and explanations regards the more evolved field of using HMM's in different type of pattern recognitions. Although, there are a few exceptions like Bulla (2006) and Zhang (2001). In addition, a few sources on the specific algorithms have been used e.g. Dempster et al. (1977) and McLachlan and Krishnan (1997) for the EM algorithm and Turner (2001) and Collings and Rydén (1998) for DNM.

4.2 Data retrieval and analysis

A lot of the work done has been through programming the algorithms of estimating the parameters for the HMM to fit the desired data. All programming has been done through MatLab since this was thought as convenient by being simple. This was not a suggestion by Nordea that probably would have preferred another tool to easy implement and use the study if possible. Though it's important to remember that the point of the study was to evaluate the ML parametric estimations and not to build a final algorithm and for this end MatLab proved to be a good choice.

The financial time series used are extracted from Bloomberg and Nordea's internal database. The series are of a few different currency pairs, OMXS30 index and generic first futures of OMXS30, Gold, S&P 500 index, Eurex Euro bond and 10Y US t-note. For the generic futures indices a 2 day early roll was used to circumvent the inclusion of iliquid contracts. The choice of data was done to get a good variety of assets. All data have been modified by looking at the daily return, this to make the normal-HMM a reasonable model. The drift was set to be independent of states whilst the volatility was state dependent.

For the programming part a few different algorithms has been created some of which can be found in Appendix A. The most part of evaluating the algorithms have been through alternating some of the parts of the algorithm, e.g. starting parameters and time series lengths, to see how the algorithms behave. Also, the information criterions AIC and BIC have been used to evaluate the optimal amount of states.

4.3 Limitations

Regarding some of the time series an approach of comparing different kinds of HMM's could've been made and not only normal HMM's. Also, the way of only varying the standard deviation could be seen as somewhat naïve since different trends in financial time series can often be subject to change in regime. By doing this delimitation we only take volatility regimes into account and disrregard of the trend regimes.

Further, the most focus was directed to the EM algorithm due to its superior stability properties and the fact that only daily data was in mind. Thus, the speed of convergence wasn't a major issue and the DNM didn't get as much attention as it would've got if the speed of the calculations were of greater importance.

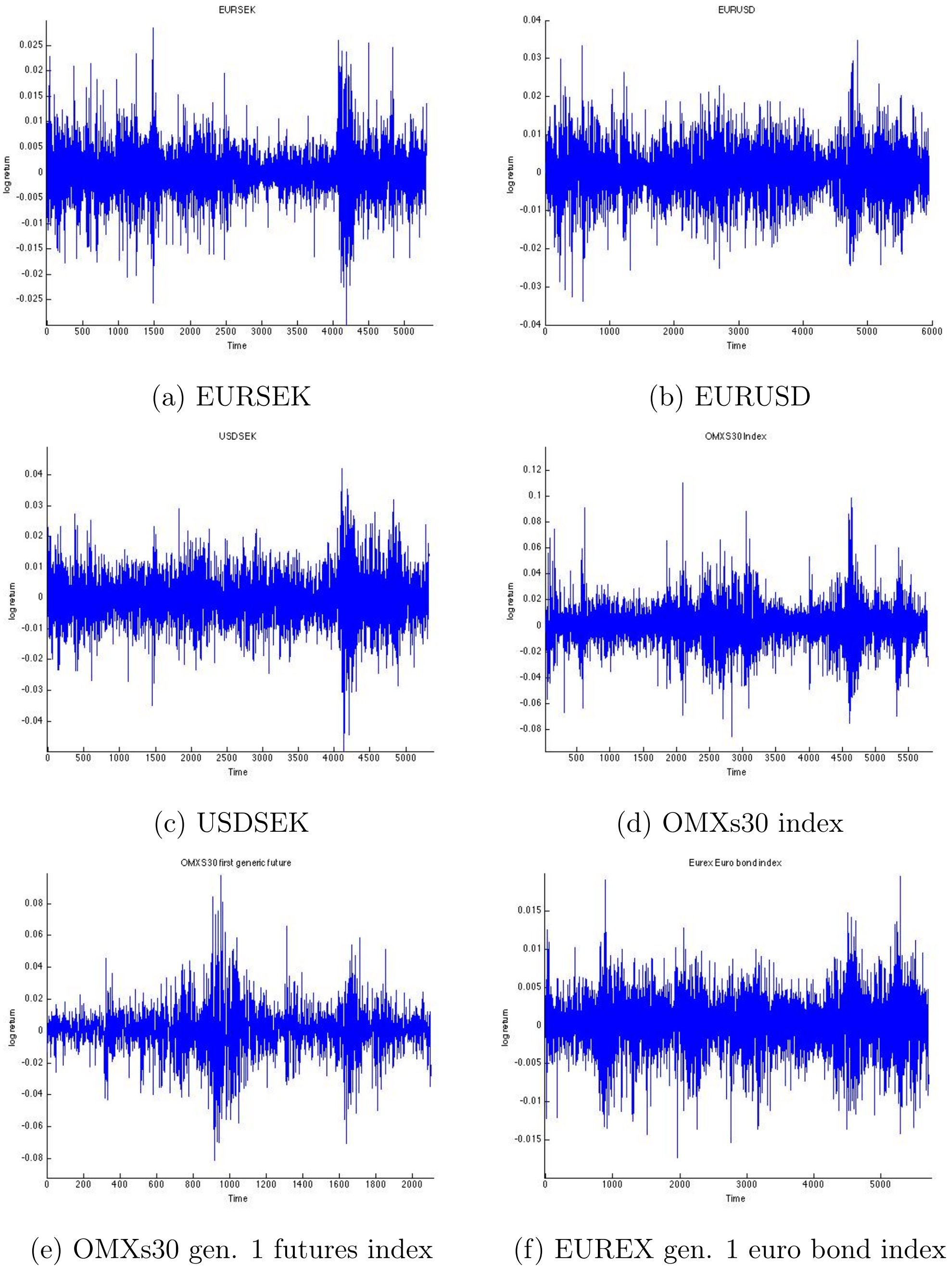
It is also important to remember that the focus in the thesis have been to evaluate the probability of the current state and the transition probability of the HMM. A further approach would've been to look at prediction and forecasting of the HMM.

Chapter 5

Findings

In this chapter the results and findings of the thesis are presented. It constitutes of chielfy visualizations of the perametric behaviour retrieved by the programs found in Appendix A. Only brief information of each result will be included in this chapter, a more extensive analysis follows in the next.

5.1 Results



(a) EURSEK

(b) EURUSD

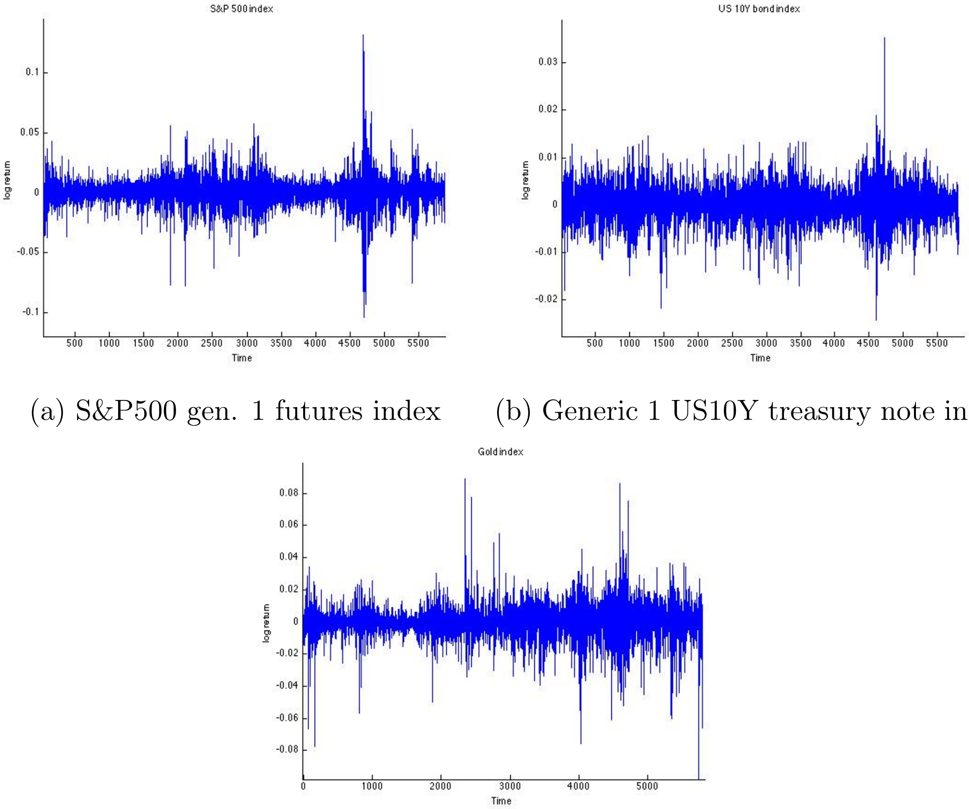
(c) USDSEK

(d) OMXs30 index

(e) OMXs30 gen. 1 futures index

(f) EUREX gen. 1 euro bond index

Figure 5.1: Daily returns for retrieved data.

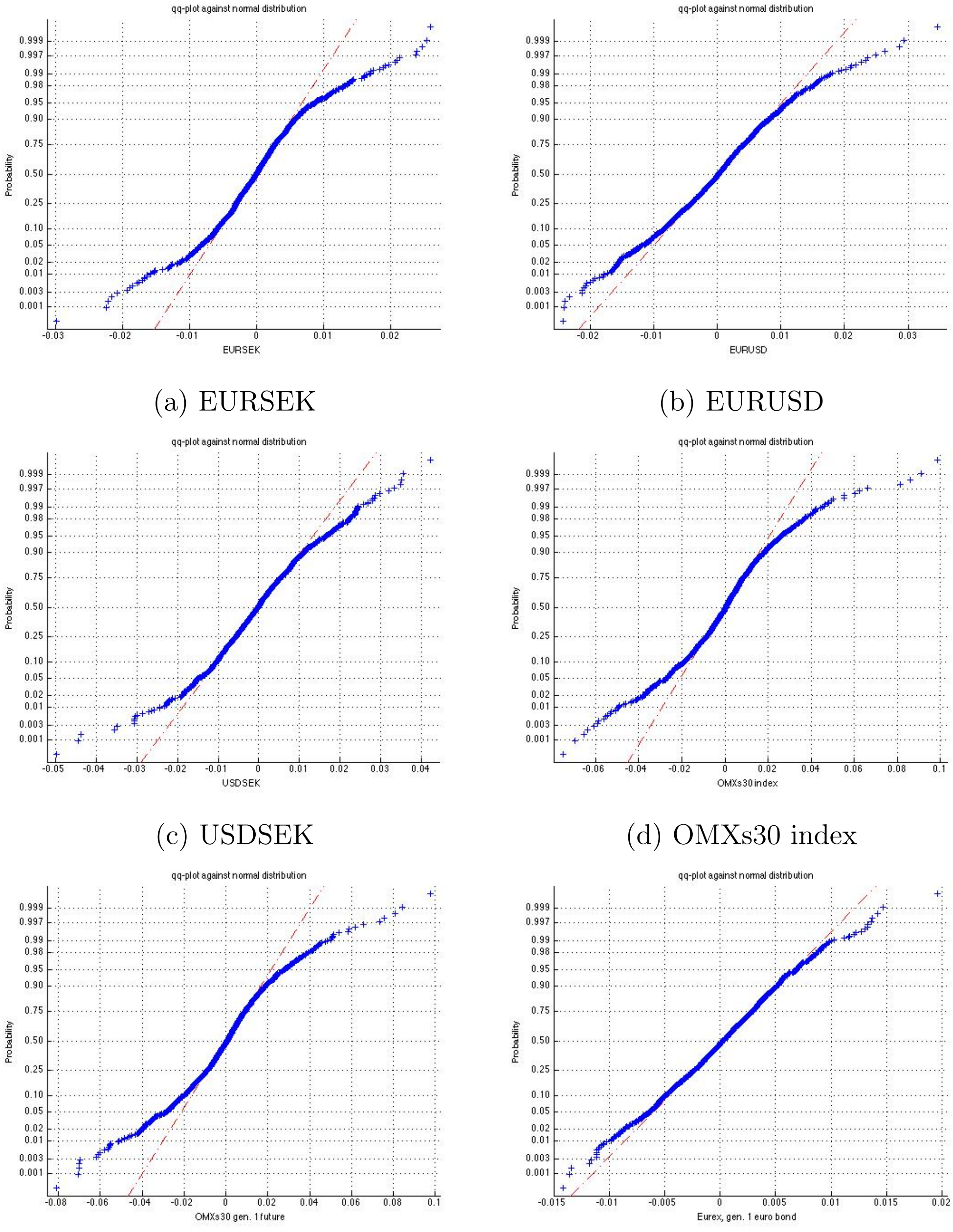


(a) S&P500 gen. 1 futures index

(b) Generic 1 US10Y treasury note index

(c) Generic 1 gold 100 oz index

Figure 5.2: Daily returns for retrieved data.



(a) EURSEK

(b) EURUSD

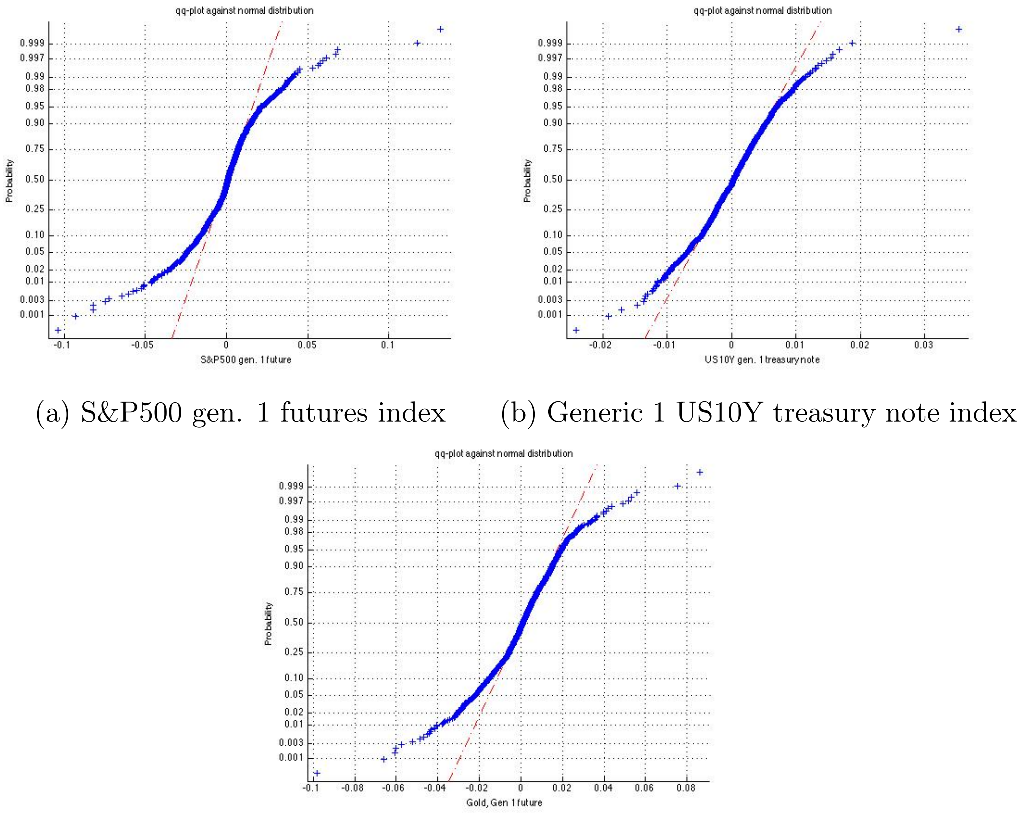
(c) USDSEK

(d) OMXs30 index

(e) OMXs30 gen. 1 futures index

(f) EUREX gen. 1 euro bond index

Figure 5.3: Normal Quantile-Quantile plot of the data.

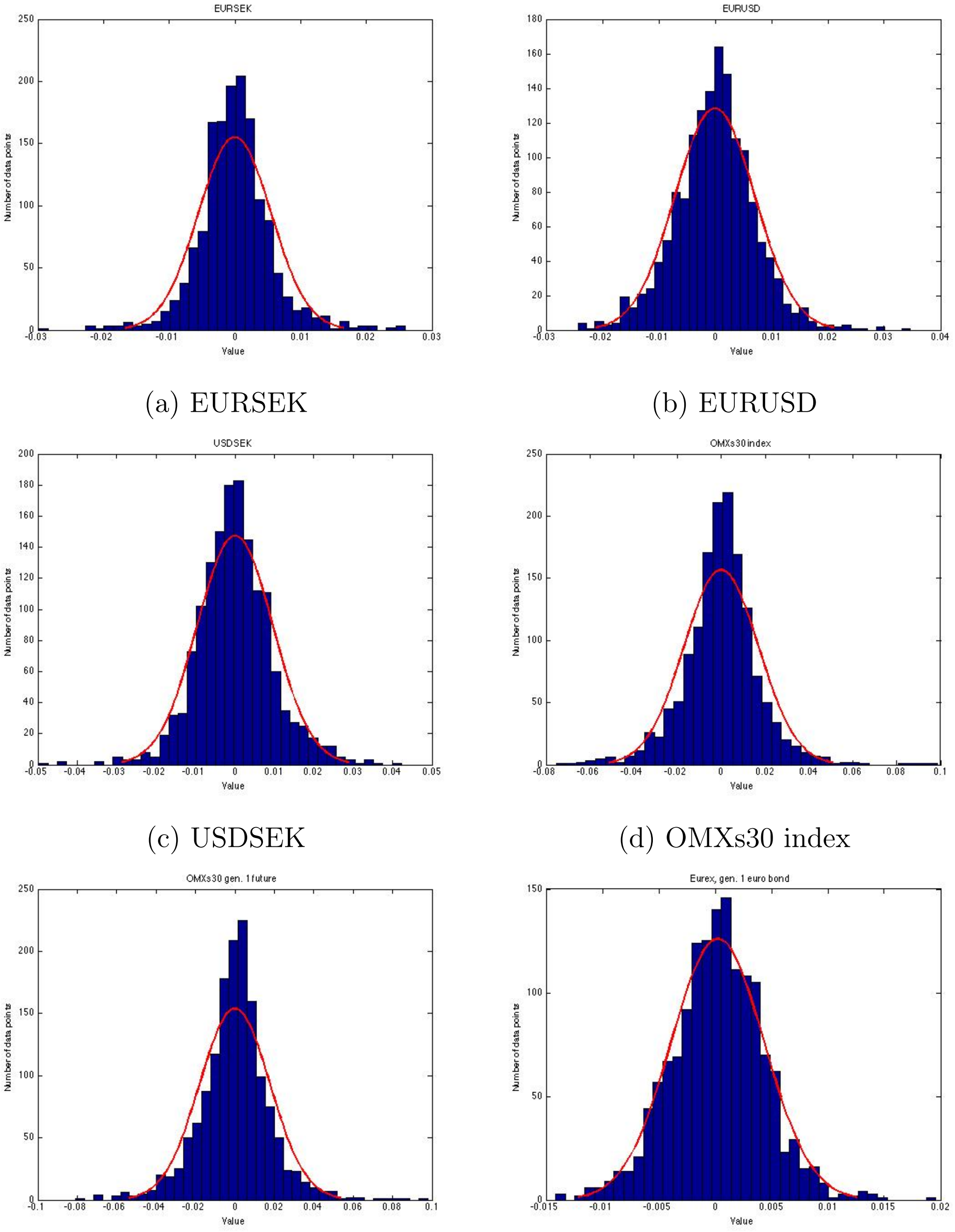


(a) S&P500 gen. 1 futures index

(b) Generic 1 US10Y treasury note index

(c) Generic 1 gold 100 oz index

Figure 5.4: Normal Quantile-Quantile plot of the data.



(a) EURSEK

(b) EURUSD

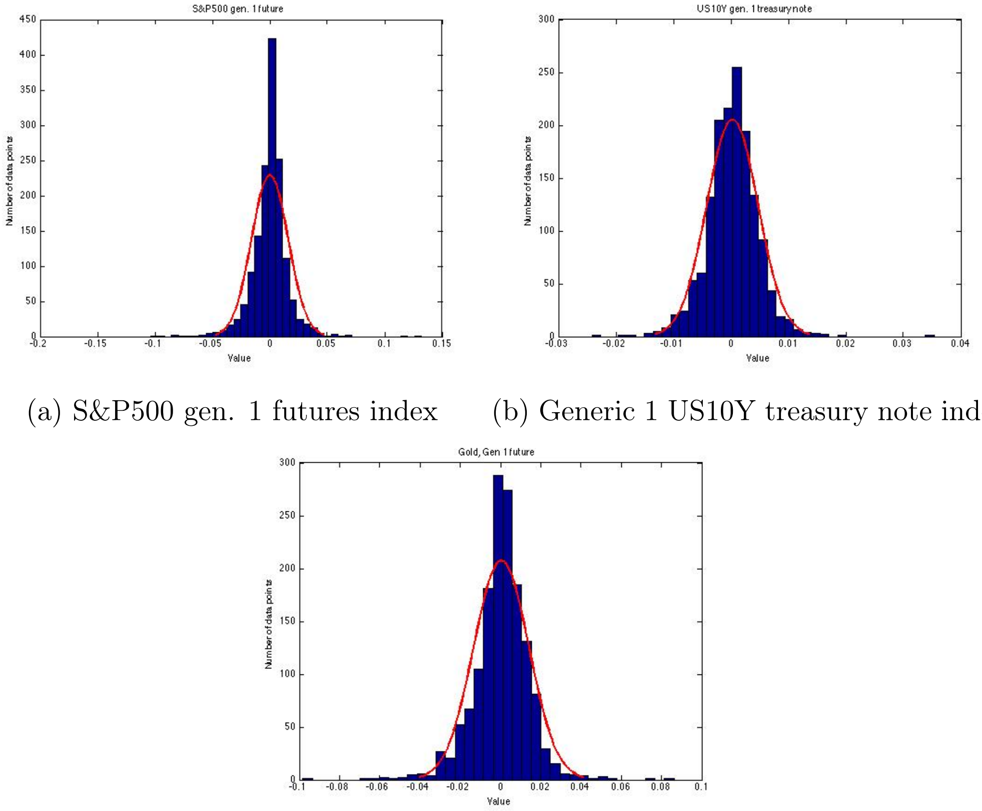
(c) USDSEK

(d) OMXs30 index

(e) OMXs30 gen. 1 futures index

(f) EUREX gen. 1 euro bond index

Figure 5.5: Histograms of data against a Normal distribution.



(a) S&P500 gen. 1 futures index

(b) Generic 1 US10Y treasury note index

(c) Generic 1 gold 100 oz index

Figure 5.6: Histograms of data against a Normal distribution.

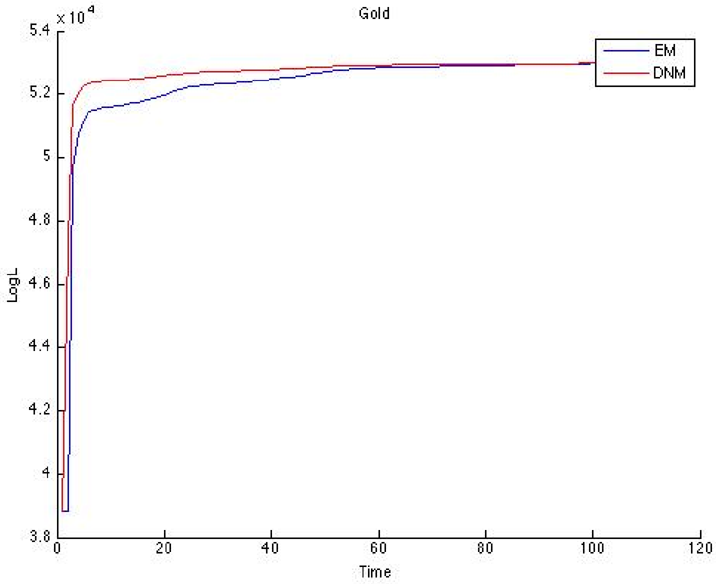


Figure 5.7: Comparison of the convergence of the logarithmic likelihood for DNM and EM with 4 states on Gen. 1 gold index.

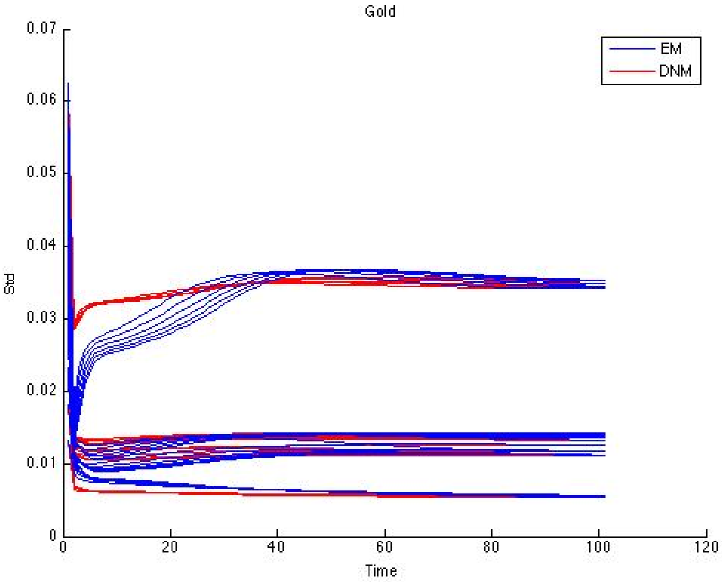


Figure 5.8: Comparison of the convergence of the state dependent standard deviations for DNM and EM with 4 states on gen. 1 gold index.

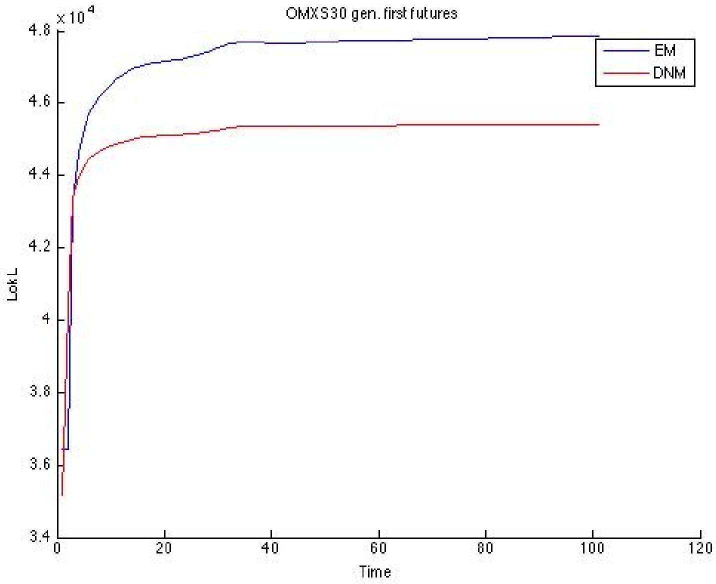


Figure 5.9: Comparison of the convergence of the logarithmic likelihood for DNM and EM with 4 states on OMX gen. 1 futures index.

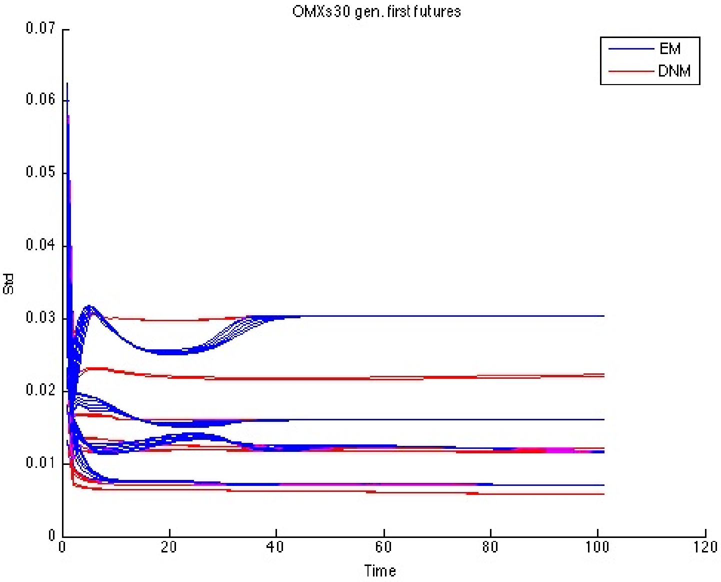


Figure 5.10: Comparison of the convergence of the state dependent standard deviations for DNM and EM with 4 states on OMX gen. 1 futures index.

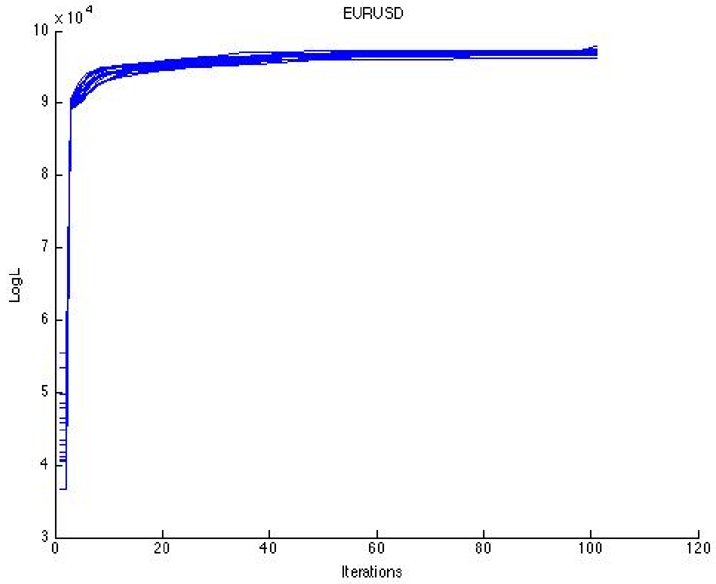


Figure 5.11: Convergence of the logarithmic likelihood for several runs of the EM algorithm with very small deviations in starting parameters.

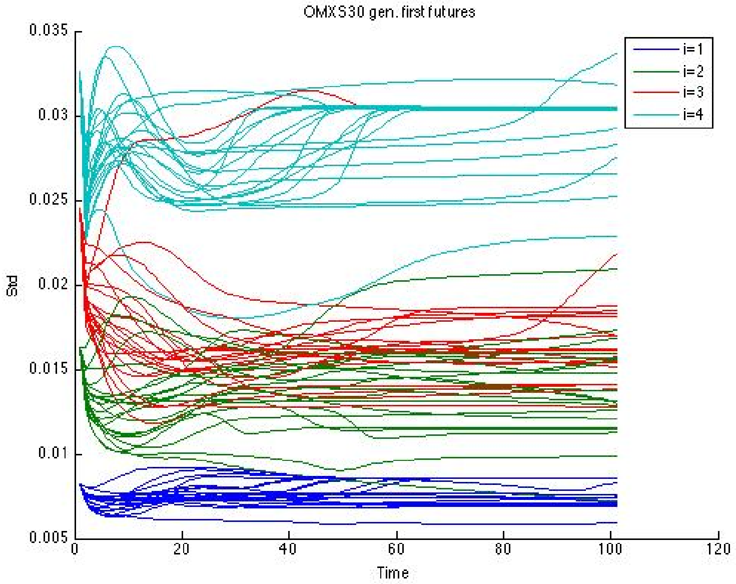
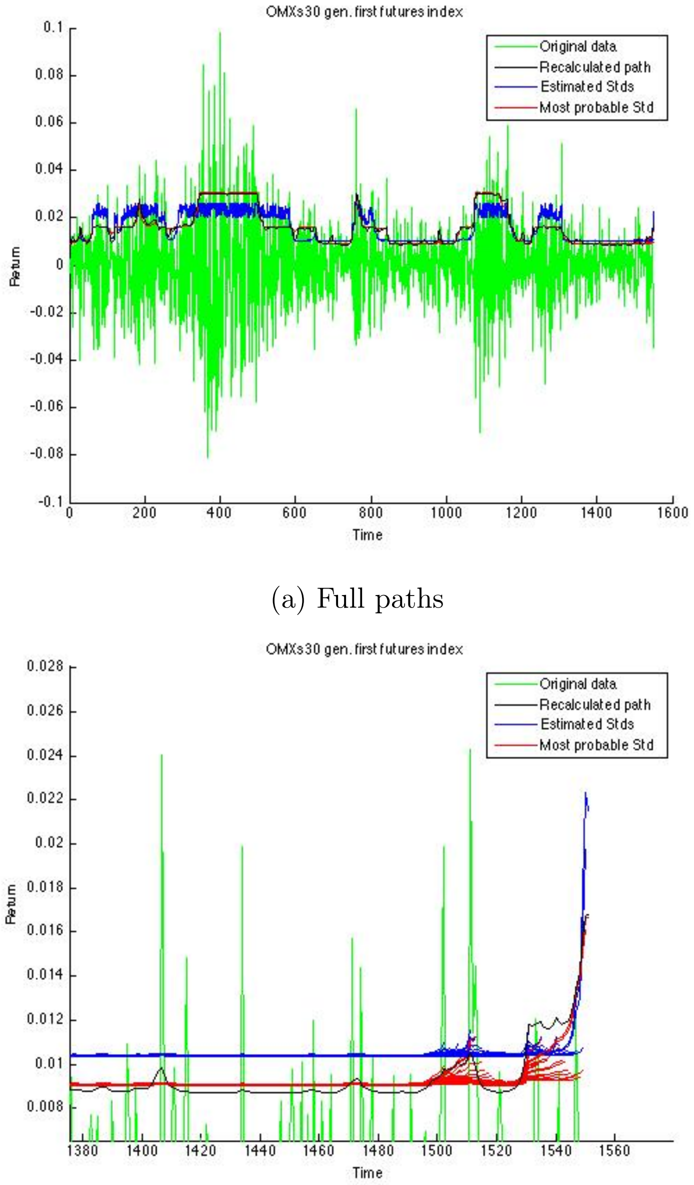


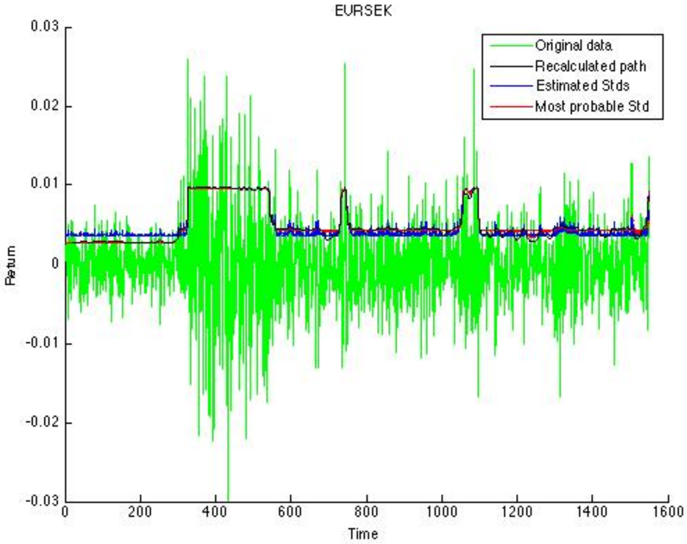
Figure 5.12: Convergence of the state dependent standard deviations for several runs of the EM algorithm with large deviations in starting parameters.



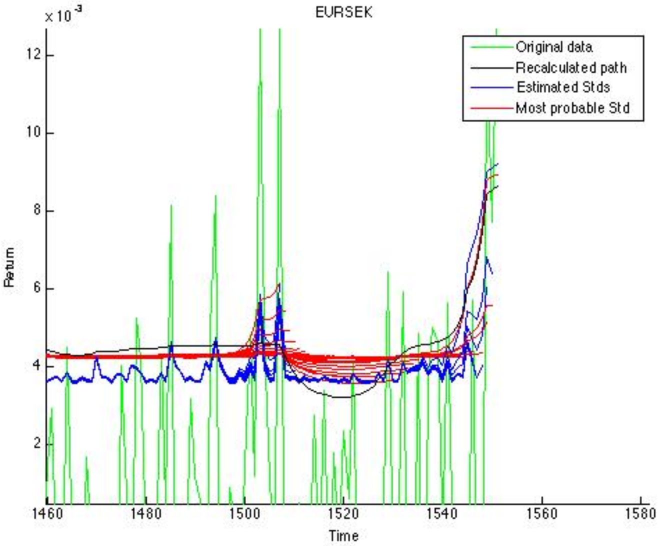
(a) Full paths

(b) Zoomed in paths

Figure 5.13: Paths of standard deviations recalculated with the θ at t as starting parameters. Most probable path uses the θ with the highest likelihood whilst the blue lines uses the θ with lower likelihood at t. The recalculated path uses the more extensive calculation of starting parameters. OMXs30 generic 1 futures index with 4 states.

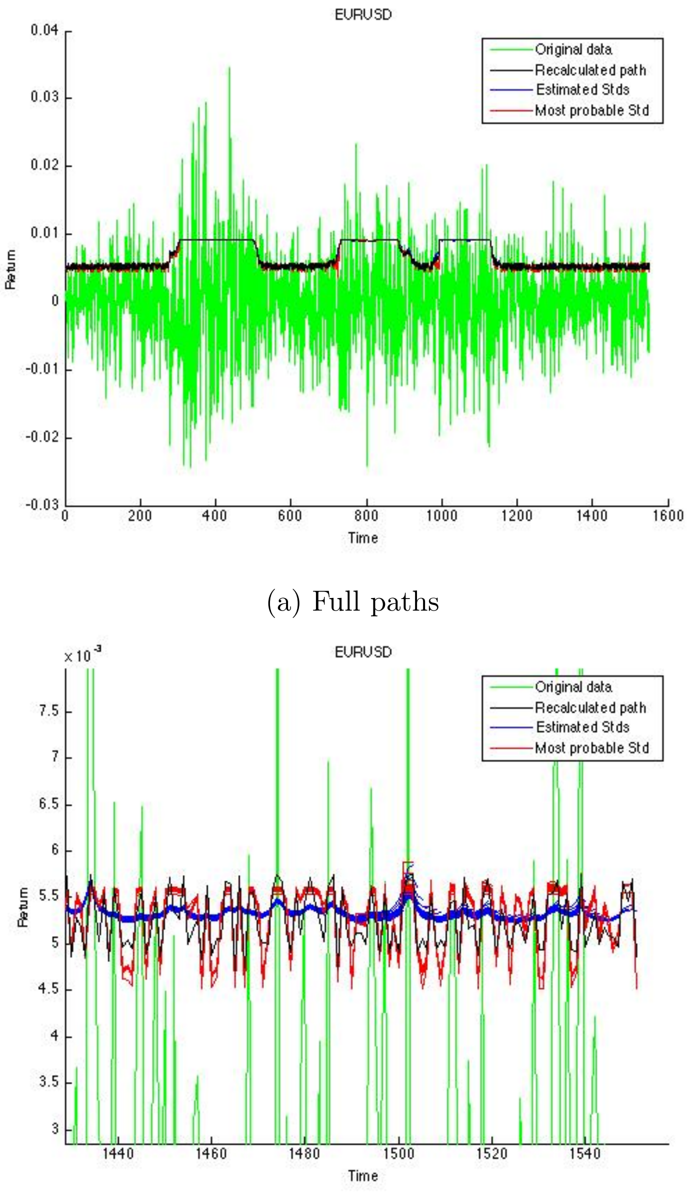


(a) Full paths



(b) Zoomed in paths

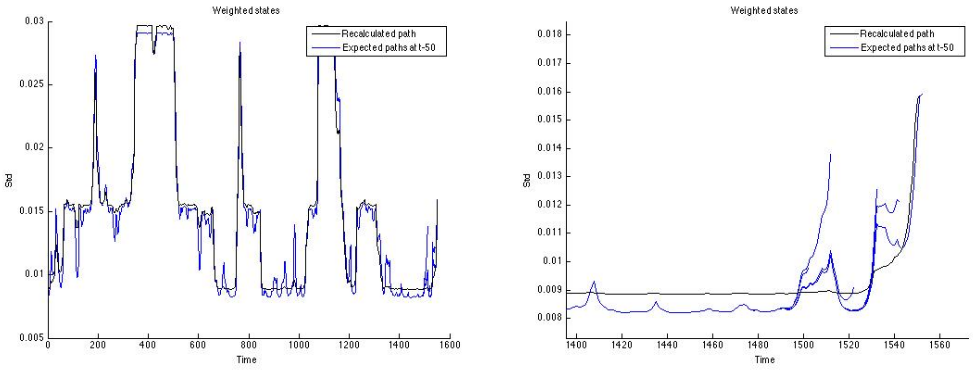
Figure 5.14: Paths of standard deviations recalculated with the θ at t as starting parameters. Most probable path uses the θ with the highest likelihood whilst the blue lines uses the θ with lower likelihood at t. The recalculated path uses the more extensive calculation of starting parameters. EURSEK with 4 states.



(a) Full paths

(b) Zoomed in paths

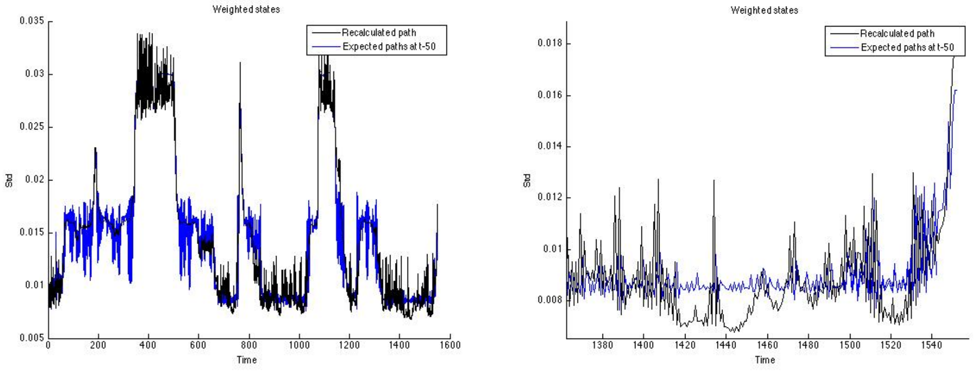
Figure 5.15: Paths of standard deviations recalculated with the θ at t as starting parameters. Most probable path uses the θ with the highest likelihood whilst the blue lines uses the θ with lower likelihood at t. The recalculated path uses the more extensive calculation of starting parameters. EURUSD with 4 states.



(a) Full paths

(b) Zoomed in paths

Figure 5.16: Path of standard deviations at t compared to a recalculated path 50 data points forward. OMXs30 index with 3 states.



(a) Full paths

(b) Zoomed in paths

Figure 5.17: Path of standard deviations at t compared to a recalculated path 50 data points forward. OMXs30 index with 8 states.

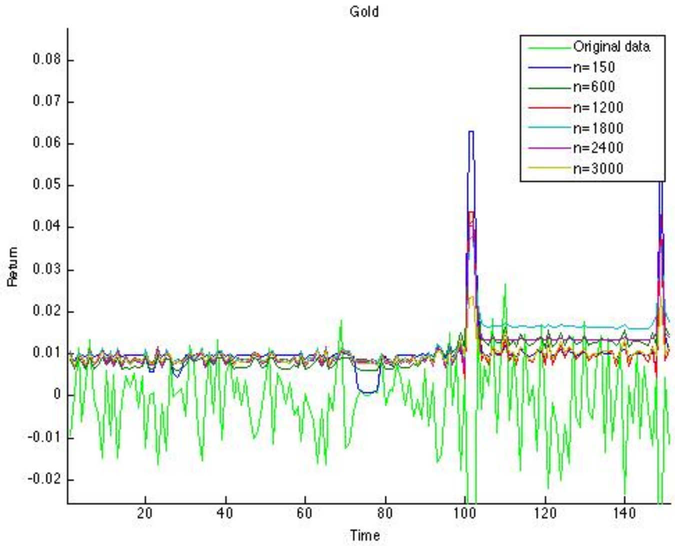


Figure 5.18: Estimated path of standard deviations for different numbers of data points of generic 1 gold index in the parameter estimation. The model used includes 8 states. The graph shows the last 150 points of all paths to be comparable.

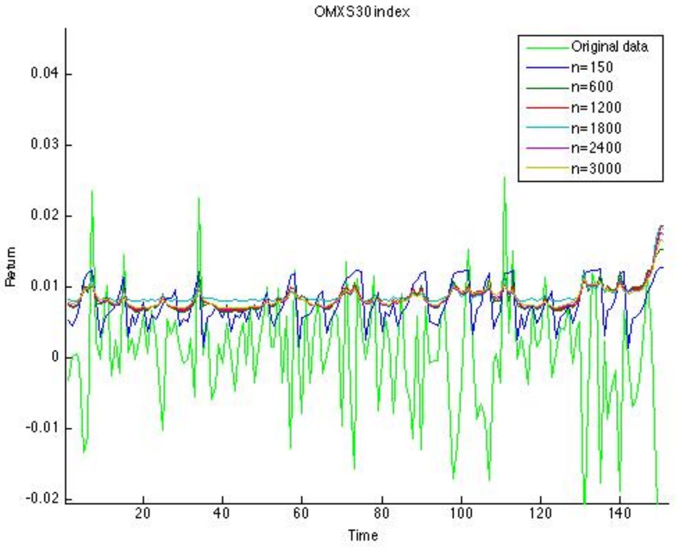
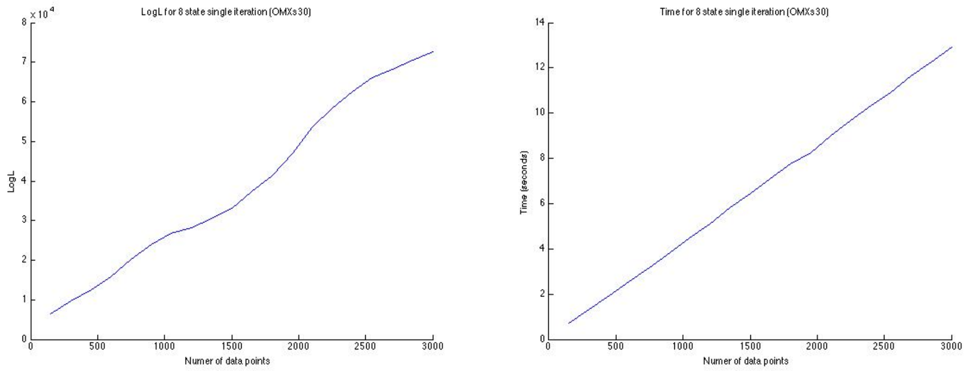


Figure 5.19: Estimated path of standard deviations for different numbers of data points of OMXs30 index in the parameter estimation. The model used includes 8 states. The graph shows the last 150 points of all paths to be comparable.



(a) Log Likelihood

(b) Time

Figure 5.20: The logarithmic likelihood and time of execution as functions of analysed data points of OMXs30 index, modeled with an 8 state model.

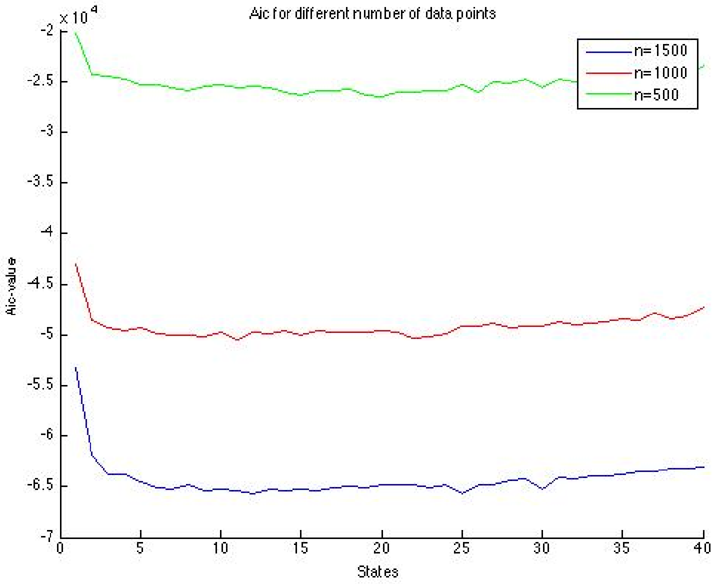


Figure 5.21: Comparison of AIC function for different lengths of the OMXs30 index.

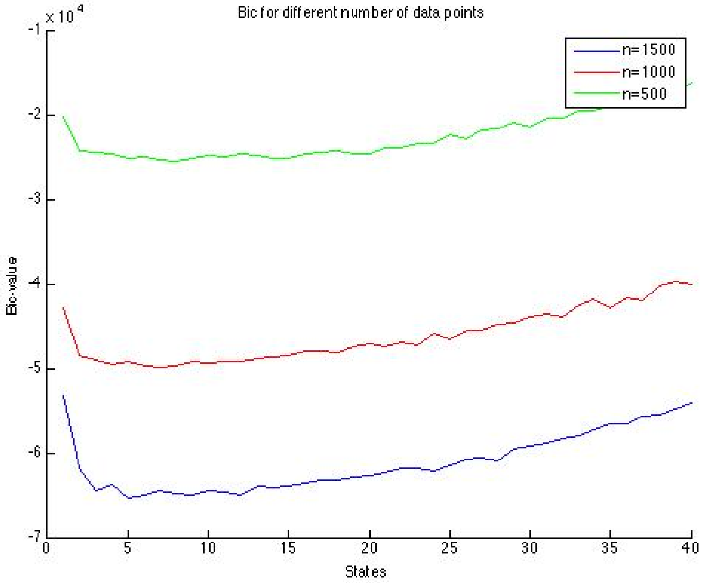


Figure 5.22: Comparison of BIC function for different lengths of the OMXs30 index.

States	AIC	BIC
n=500	20	7
n=1000	11	6
n=1500	12	5

Figure 5.23: Table of AIC and BIC values for different lengths of OMXs30 index.

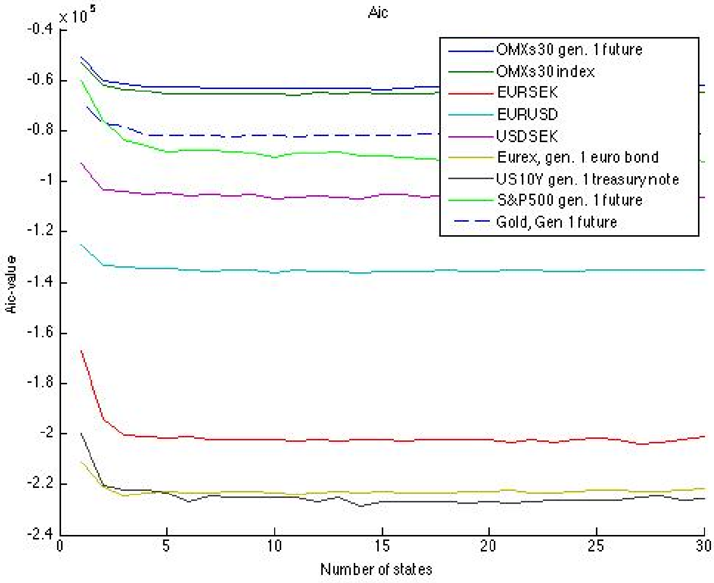


Figure 5.24: Comparison of AIC function for all data series with 1500 data points.

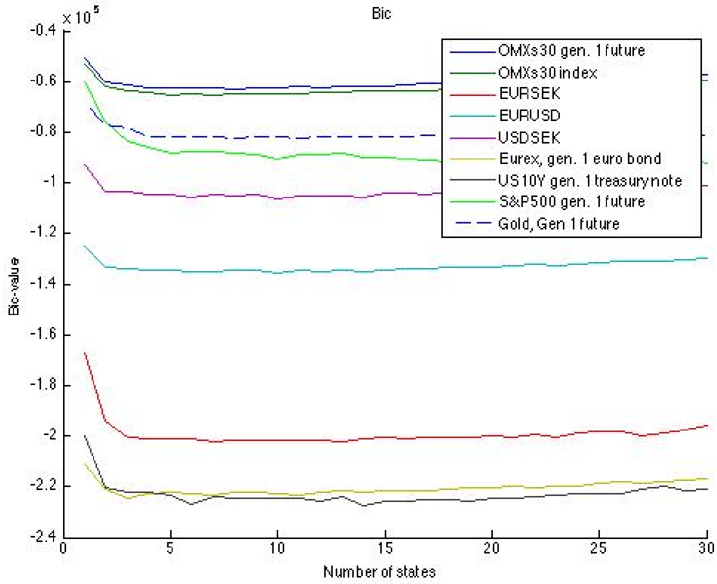


Figure 5.25: Comparison of BIC function for all data series with 1500 data points.

States	AIC	BIC
OMXs30	8	15
OMXs30 gen. 1 fut	5	25
EURSEK	7	27
EURUSD	10	14
USDSEK	10	29
Eurex gen. 1 fut	3	3
US10Y gen. 1 fut	14	14
S&P 500 gen. 1 fut	18	30
Gold gen. 1 fut	8	8

Figure 5.26: Table of AIC and BIC values for different data series with all length 1500.

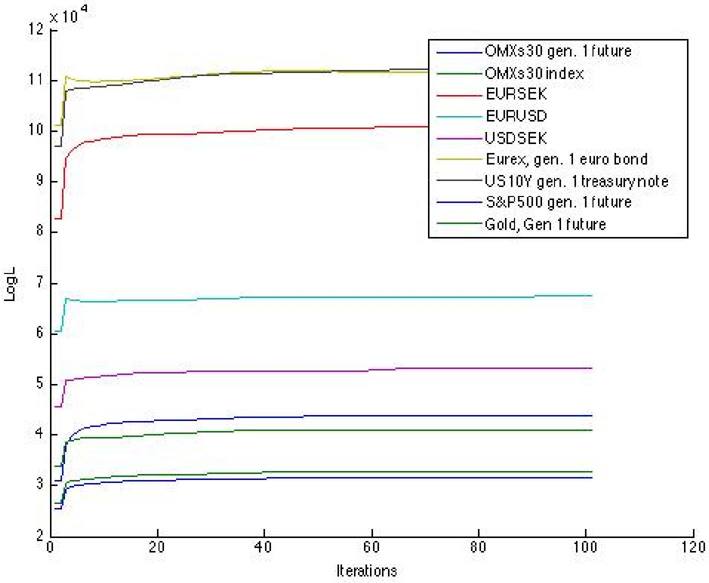
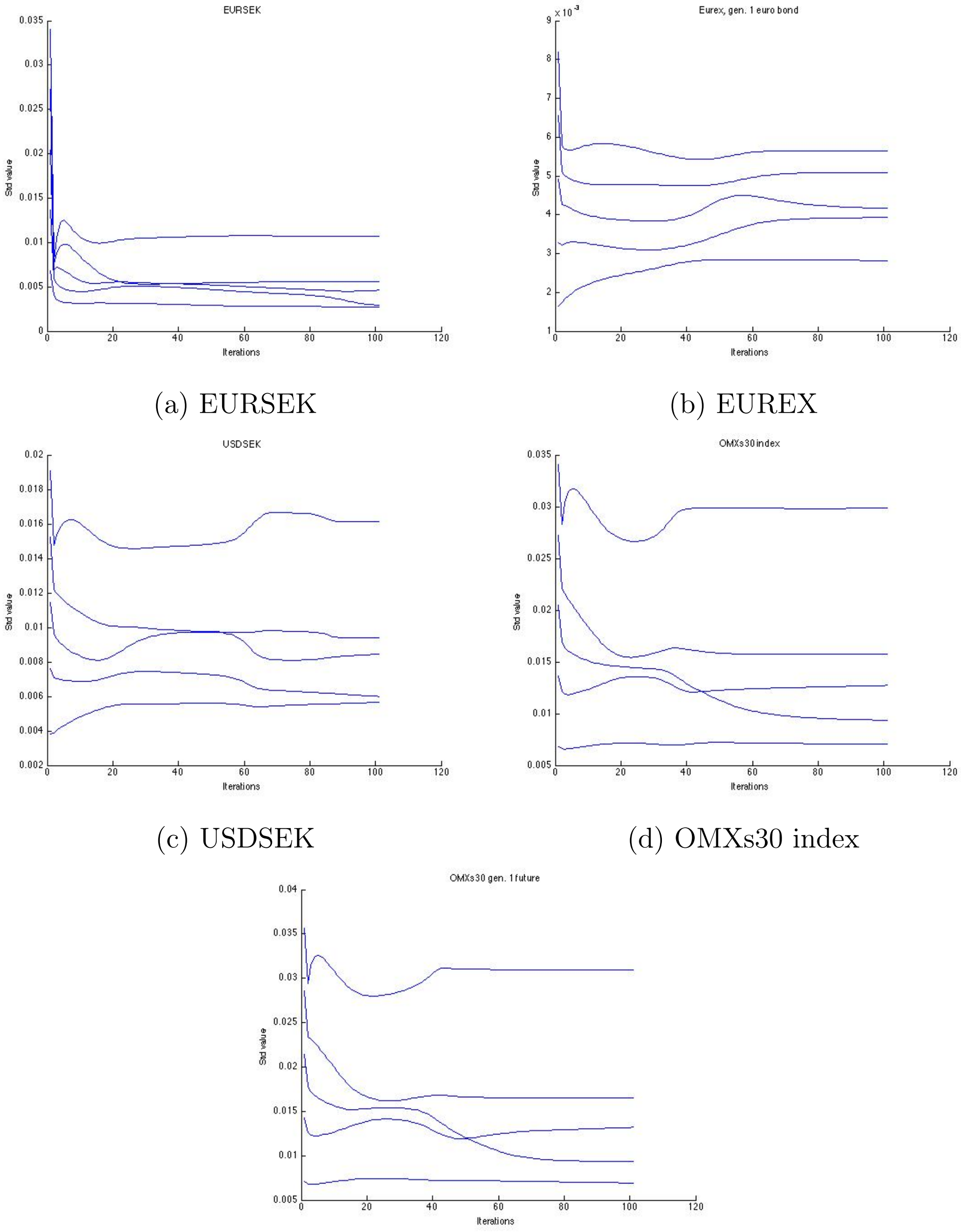


Figure 5.27: Best estimates of logarithmic likelihood with a 5-state model of 1500 data points.



(a) EURSEK

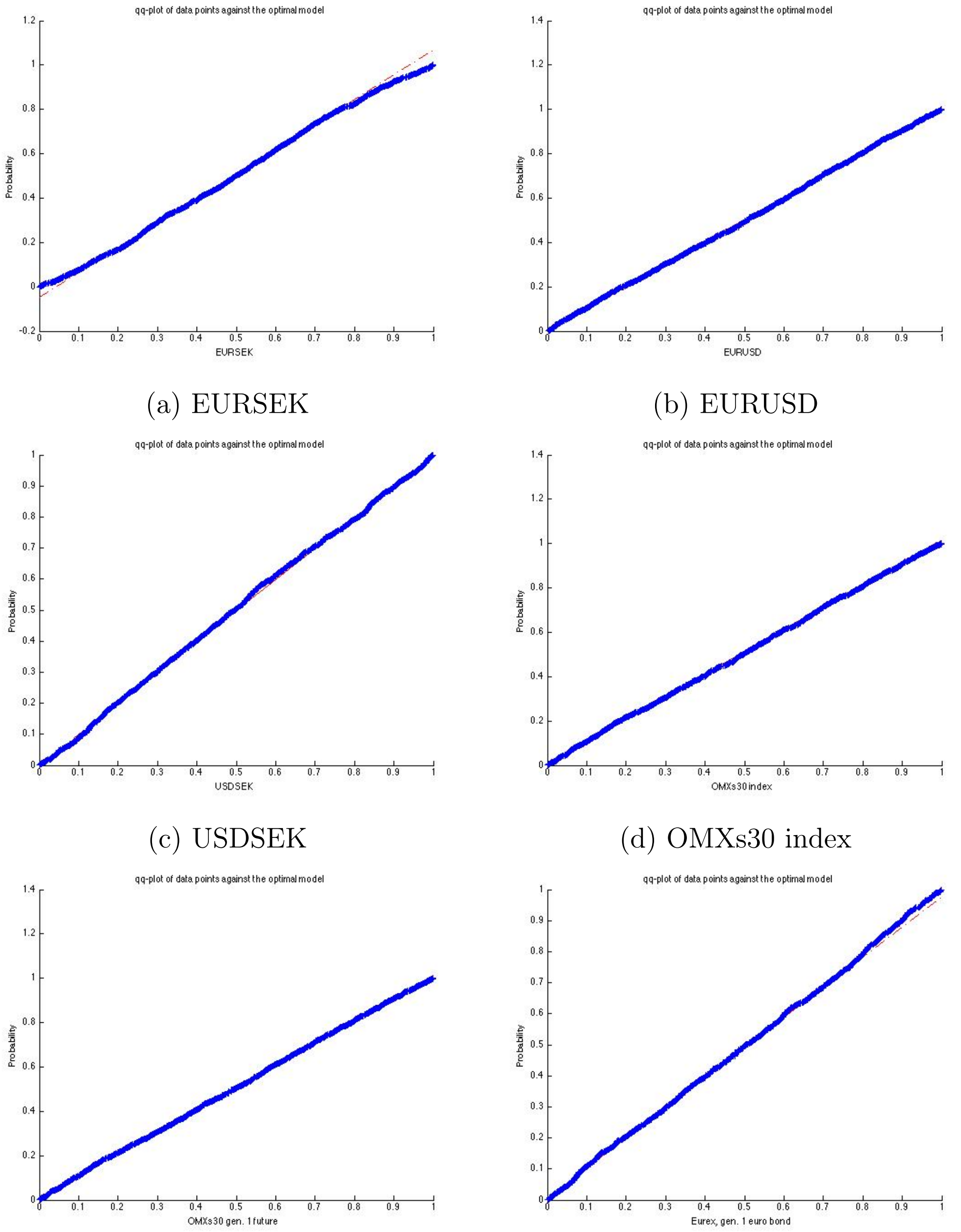
(b) EUREX

(c) USDSEK

(d) OMXs30 index

(e) OMXs30 gen. 1 futures index

Figure 5.28: Converged iterations of state dependent standard deviations for a 5-state model on 1500 data points.



(a) EURSEK

(b) EURUSD

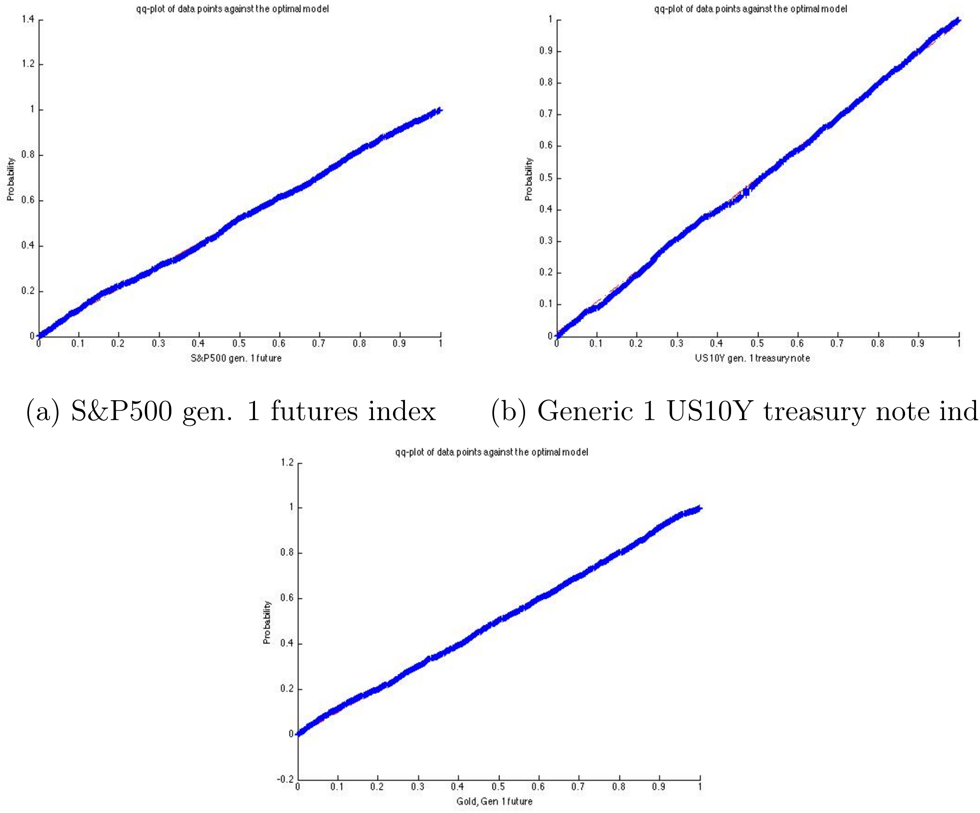
(c) USDSEK

(d) OMXs30 index

(e) OMXs30 gen. 1 futures index

(f) EUREX gen. 1 euro bond index

Figure 5.29: Quantile-Quantile plot of data series against the estimated HMM for that series.

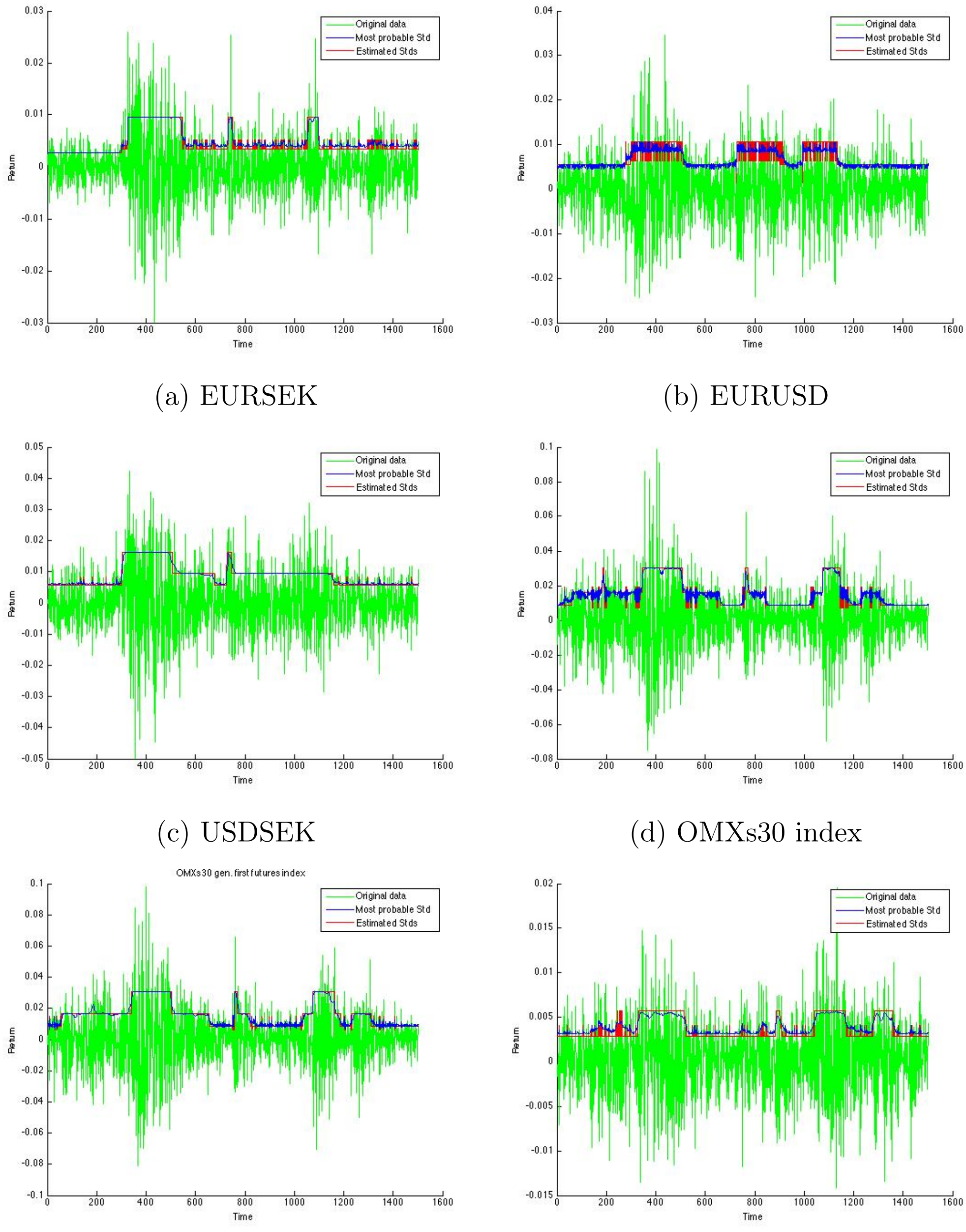


(a) S&P500 gen. 1 futures index

(b) Generic 1 US10Y treasury note index

(c) Generic 1 gold 100 oz index

Figure 5.30: Quantile-Quantile plot of data series against the estimated HMM for that series.



(a) EURSEK

(b) EURUSD

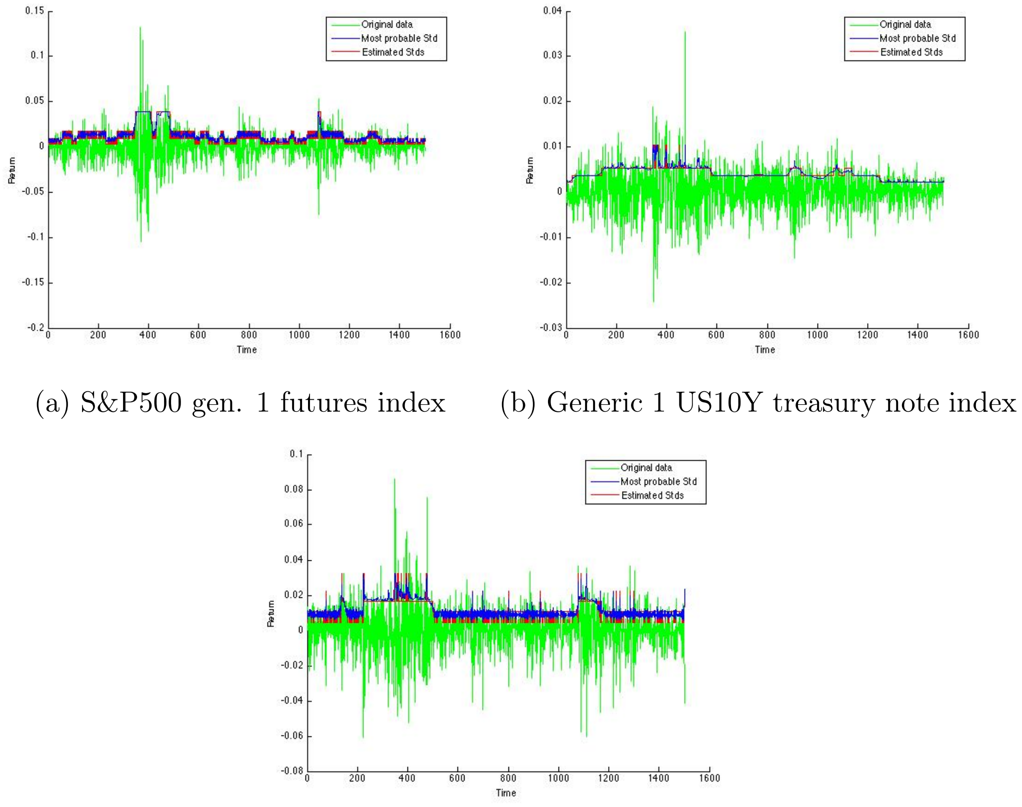
(c) USDSEK

(d) OMXs30 index

(e) OMXs30 gen. 1 futures index

(f) EUREX gen. 1euro bond index

Figure 5.31: Estimated paths of standard deviations. Most probable path are weighted mean of standard deviations at a certain time. The red line is the Standard deviation in the state with highest probability.



(a) S&P500 gen. 1 futures index

(b) Generic 1 US10Y treasury note index

(c) Generic 1 gold 100 oz index

Figure 5.32: Estimated paths of standard deviations. Most probable path are weighted mean of standard deviations at a certain time. The red line is the Standard deviation in the state with highest probability.

Chapter 6

Analysis

This chapter includes first a section in which the findings in the previous chapter is discussed in respect to the theories. The last section aims to sumarize all findings done in this report.

6.1 Discussion

The first few graphs in the result section are visualizations of the plain data that has been used in this thesis. Both Figures 5.1 and 5.2 are simply the return of the different assets. As noticed, the graphs don't match the number of data points for between series. This is explained by the fact that the time series wasn't equally available at our source. For most of the study only the latter 1500 data points will be used. In fact, if any other length is used this will be noted.

As seen by Figures 5.1 and 5.2 some similarities within the time series can be seen. This is of obviously nothing surprisingly since there often are significant patterns within financial data. For example, we see that the 2009 financial crisis stands out quite significantly in almost all of the time series by an increased volatility about 1000 data points from the end. Also interesting is that we can see some volatility spikes in some of the series at 2012, about 400 data points from the end, pointing out the euro zone crisis.

Looking at the next section of figures we have gone more into the cause of why a hidden Markov model is a good approach of modeling these time series. Figures 5.3 and 5.4 shows a quantile-quantile plot of the time series fitted as a normal distribution. As seen, the tails of all the time series seems to be significantly heavier than what the model of a single normal distribution can handle. This is also inferred by the figures 5.5 and 5.6 that shows histograms for all data against a normal distribution. We see that the tails are heavier than the normal model and also more concentrated around the mean. Hence, the usage of an independent mixture model could be a good substitute of the normal model since we would be able to concentrate much of our density around the mean but still capture the heavy tails. Though, the patterns in Figures 5.1 and 5.2 suggests that maybe an independent approach isn't the best approach. Remember the fact that a dependent model needs to be used for time series that seems to cluster. Naturally this leads us into the family of dependent mixture models, of which HMM's are a part of.

By the theory, and also mentioned in the method section, we know that the two different approaches of parametric estimation differs somewhat. None of which is superior to the other in all aspects, e.g. the Expectation-Maximization algorithm seems to be more stable whereas the approach of direct numerical maximization is faster to converge. This can be seen in figures 5.7, 5.8, 5.9 and 5.10 that visualize the convergence process of the algorithms on the time series of OMXs30 generic first futures index and generic first gold index. In all graphs a 4-state model are used. In Figures 5.8 and 5.10 the starting parameters for the standard deviations have been bumped to visualize how the algorithms behave and how the σ_i's converge. As seen in especially figure 5.7 the theory that estimation by DNM should be faster is supported. DNM seemed to reach the stopping criterion quite a lot faster than the EM algorithm did. The figure shows only the best run with the starting parameters resulting in the highest logarithmic likelihood for the model. In Figure 5.8 only 6 blue lines can be distinguished even though the bump is done 10 times. Hence, some instability in the EM is spotted in this graph. This goes also for the DNM since we can distinguish 5 clusters of red lines in the graph. This is also true for figure 5.10 where 6 different clusters of red lines can be seen and since it is a 4-state model this must also be an effect of some instability. Regarding the blue lines on the other hand all the starting values seems to converge to the same set of parameters, suggesting stability.

In figure 5.9 the difference in convergence of logarithmic likelihood is compared for one of the starting values where the estimation by DNM seems to converge to a different value. As seen by the figure the logarithmic likelihood converges but to a different, and lower, value. This is actually an interesting result since it shows that if the starting parameters are not chosen wisely the model can come to converge to a local maximum, and not the wanted global maxima. As was mentioned in the method part we choose in respect to these results to look deeper into the EM algorithm since we don't need to take such a big account of the time issue for our, relatively, short time series and the fact that we're only looking at daily data. With this said though, the instability of the two algorithms isn't that significant for small deviations in some of the starting parameters.

Even though the result isn't that dependent on small differences in the starting parameters, see figure 5.11 that is a graph of 40 different starting parameters of the standard deviations, major changes in all starting parameters can yield vastly different results. Some quite big alterations was done in figure 5.12 which really shows the instability if starting parameters aren't chosen wisely. Because of this the model that has been used in the latter part of the results runs several tries of the EM algorithm with different starting parameters. The algorithm for this is a little bit naïve since it doesn't evaluate the different starting parameters by the resulting likelihood but only loops over a few different pre-set parameters and some randomized ones.

The figures in 5.13, 5.14 and 5.15 are all the same but on different time series. The approach here have been to first find a high likelihood from a big amount of starting parameters, after which a second iteration has been done with smaller changes in the parameters. We've then rolled the estimation forward to model a larger time series and used the final parameters θ as starting parameters. The extensive procedure of alternating starting parameters has been done at $t + 50$ to model benchmark against. The lower pictures in all figures are zooms of the figures above. As seen by the graphs the red line models the recalculation better than the blue. Hence it still seems to have a higher likelihood than the estimates done earlier. Still, it's not a perfect fit with the black line which means that some other starting parameters yield a higher likelihood. Looking at the end of each red line we see that those lines model the black line quite good, especially for the currency graphs. In Figure 5.13 we see that before the cluster of high volatility in the end some difference with respect to the black line is evident. Suggestible would therefore be to recalculate this model earlier than in 50 time steps. For all figures the deviations seems to grow when further away from the calibration data, hence recalibrating for all series at $t + 30$ would be suggestible.

Similar to figures in 5.13, 5.14 and 5.15 are the figures 5.16, 5.17 and 5.17. The difference is that here we don't run the full EM algorithm at each new calculation, instead only the transition matrix and the path are used to calculate the path of the Markov chain. It can be seen as more of a semi-prediction of the future Markov chain since we use the available data at $t + k$ but only the parameters retrieved at t without any new itearations. The patterns are similar to the ones in the previous figures since the improvement by iteration the algorithm from the parameters θ at t isn't that big when the number of values hasn't increased significantly. For 5.16 we have used a model of 3 states whilst a 8-state model was used in 5.17. For both models it's quite clear that if a new iteration of the EM algorithm isn't done the fit decreases much faster than if we run the algorithm. Also, the time of running the algorithm from good starting parameters is quite low.

As seen in the figure the 8-state model is more volatile since it got more states to jump between. This is a setback when increasing the amount of states, the likelihood is increased but the simplicity of the model is simultaneously decreased. If we for example have the number of states in the same order as data points we will have a perfect fit, but a very uninteresting model. This lies under the subject of validation the model which has been done, not only through AIC and BIC but also through changing the number of data points used in the model. Figures 5.18 and 5.19 shows such a comparison of the estimated Markov chain. Here, data points from $t-x$ to t have been used to create the estimates. We see that by using a quite short learning window the model becomes more volatile, which is reasonable considering it was the same by increasing the number of states. Figure 5.20 shows that when we uses more data the likelihood increases, but so does the execution time of the model. Here the information criterions come into place to model the trade of for an as effective model as possible.

Firstly the AIC and BIC in Figures 5.21 and 5.22 compares the behavior of the information criterion in respect to the number of data points. It is evident that at least 1000 data points should be used since the difference isn't linear compared to the difference between the 1000 and 1500 data point models. Table 5.26 shows the optimal amount of states for each model. The BIC and AIC differs quite significantly since the BIC also take the number of elements into account. Thus, BIC is punishing the likelihood harder than AIC for higher amounts of data points. Because of this our suggestion is to use a 5-6 state model for a higher amount of values. Supporting this we also have the fact that the execution time in the graph (b) of figure 5.20 isn't that large for these series. Figures 5.24 and 5.25 shows the different AIC and BIC values for all time series. We also have table 5.26 that shows that the optimal number of states is highly dependent on the type of time series. Thus, a general HMM is quite hard to reach. Reasonably one could choose the number of states equal to the model with the highest BIC-value. But then, again, we're looking at an inefficient model and even uninteresting in some cases. The suggestion would therefore be to have a dynamic amount of states. The best model possible would be to first use any of the two information criterions, preferably the BIC, and then use the calibration algorithm.

In contrast to the reasoning in the previous paragraph the last part of the results uses the same model for all time series to get some kind of comparison feature. A model of 1500 data points was used on a 5-state model. Figure 5.27 shows the different logarithmic likelihoods for this model on all the time series. Looking at the size of logarithmic likelihood it is evident that some of the series isn't modeled optimally by only 5 states. Not too surprisingly the Eurex bond index have the best fit, since that was the model with the least value of BIC. This comparison is a little bit confusing though, since the logarithmic likelihood can differ significantly with respect to the time series. The figure 5.27 could in fact only tell us that Eurex and US10Y are series easier to model by a HMM. Figure 5.28 shows the convergence of the models standard deviations. Not to surprisingly the two OMX connected time series have similar states of standard deviations. Most interesting is the EURSEK since it seems to converge to a 4-state model. This is also the case for the Eurex and USDSEK graphs, but not as significant as for the EURSEK. For the Eurex series it could be caused by the fact that the optimal model for this time series is only 3, hence a reduction of states is supported by the BIC-value. For the other two series the 5-state model is far from the optimal model of 27 and 29 the EURSEK and the USDSEK respectively. Thus, a 5-state model really misses some information such that a reduction of states might not be negative for the likelihood.

The most interesting result in the thesis is the figures 5.29 and 5.30 that plot the quantile-quantile of the data against the suggested models. These figures show that if the estimation is done extensively quite good models can be generated. For almost all of the data the fit seems to be perfect. The last figures of the result section show the realization of the estimated Markov chains. The state with highest probability, denoted by red, seems to be a bit more volatile than that of the weighted mean of states, blue. This is in fact not too surprising since the weighted realization models equally probable states better. It is also evident that for some of the time series an increase in the amount of states is needed. In some parts large clusters seems to be disregarded.

6.2 Conclusions

The approach of modeling financial time series with Hidden Markov Models proved to yield a more accurate model than of only using a fitted normal distribution. As the choice of algorithm for parameter estimation the EM algorithm proved to be the best one for to this end. Mostly due to the fact that the main advantage of DNM lies in the speed of convergence whilst we where more interested in the most stable solution. We've also seen that the most important factor of making an accurate HMM model is to choose the optimal number of states. Using too few models will result in a bad fit whilst too many will yield uninteresting models.

Regarding an implementation of a model, the best way would be to use an extensive algorithm of estimating starting parameters monthly and use these final parameters as the starting parameters for the days between recalibration dates. If possible though, the calibration could be done more often to provide an even better fit.

6.2.1 Future research

The most interesting expansion of this thesis would be to further look into the prediction possibility of using HMM's. Another natural expansion would be to include a wider variety of distributions over the states. For example, trend regimes could be introduced or other types of distributions and not only normal HMM's.

A Monte Carlo simulation approach could also be used to further evaluate the prediction possibility by HMM's. This could also be a way a more extensive comparison of the impact different time series lengths has on the model.

But the most interesting take to move forward would be to look into the possibility of using implied volatility to help the model calibrate and estimate.

Appendix A

Source Code

A.1 EM-algorithm

```
function [ vSP, mTP, mEP, dMV, vStd, dLL, vLLplot, vMVplot, mStdplot,
    %defining variables & allocationg space
    iLen=numel(vData);
    iK=numel(vSP0);
    i=2;
    cStTime=clock;
    dMV=dMV0;
    vStd=vStd0;
    mTP=mTP0;
    vSP=vSP0;
    vLLplot=zeros(iIter+1,1);
    vMVplot=zeros(iIter+1,1);
    mStdplot=zeros(iIter+1,iK);
    mSP0plot=zeros(iIter+1,iK);
    vMVplot(1)=dMV;
    mStdplot(1,:)=vStd;
    mSP0plot(1,:)=vSP;
    mG=fNcdf(vData, iLen, vStd, dMV);
    [~,dLL]=fFwd( mG, mTP, vSP, iLen, iK);
    vLLplot(1)=dLL;
    dDeltaLL=100;
    dDeltaPar=100;
    while and(i<=iIter+1, or(dEpsLL<dDeltaLL, dEpsPar<dDeltaPar))
        mTPold=mTP;
        mG=fNcdf(vData, iLen, vStd, dMV);
```

```
[mFwd,dLL]=fFwd( mG, mTP, vSP, iLen, iK);
mBwd=fBwd(mG, mTP, iLen, iK);
%Emission matrix updated
mEP=mFwd.*mBwd;
mEPsum=sum(mEP,2);
mEP=mEP./repmat(mEPsum,1,iK);
%Fix matrixes Fwd, Bwd and G for calculation of TPM
mFwd=mFwd(1:iLen,:)';
mFwd=repmat(mFwd(:),1,iK);
mBwd=mBwd(2:iLen+1,:);
mBwd=repmat(mBwd(:),1,iK)';
mBwd=reshape(mBwd,iLen*iK,iK);
mG=repmat(mG(:),1,iK)';
mG=reshape(mG,iLen*iK,iK);
%Transition matrices defined
mTPM=repmat(mTP,iLen,1);
mT=mFwd.*mTPM.*mBwd.*mG;
mTSum=sum(reshape(sum(mT,2),iK,iLen))';
mTSum=repmat(mTSum,1,iK)';
mTSum=repmat(mTSum(:),1,iK);
mT=mT./mTSum;
mT=mT';
mT=reshape(sum(reshape(mT(:),iK*iK,iLen),2),iK,iK)';
mOutT=repmat(sum(mT,2),1,iK);
%New parameters defined
vSP=mEP(1,:);
mTP=mT./mOutT;
%Looping iLoop-times to calculate new parameters for mean & st
mEP=mEP(2:iLen+1,:);
dMS=1000;
j=1;
dEpsMS=10^-10;
iLoop=100;
while and(iLoop>j,dEpsMS<dMS)
     vIStd=1./vStd.^2;
     dMVlast=dMV;
     vStdlast=vStd;
     dMV=sum(sum(repmat(vData,1,iK).*mEP.*repmat(vIStd,iLen,1))
     vStd=sqrt(sum(repmat((vData-dMV).^2,1,iK).*mEP)./sum(mEP))
     dMS=sum((vStd-vStdlast).^2)+(dMV-dMVlast)^2;
     j=j+1;
```

```
        end
        vMVplot(i)=dMV;
        mStdplot(i,:)=vStd;
        mSP0plot(i,:)=vSP;
        vLLplot(i)=dLL;
        dDeltaLL=(vLLplot(i)-vLLplot(i-1))^2;
        dDeltaPar=sum((mStdplot(i,:) - mStdplot(i-1,:)).^2,2)+(vMVplot(i
        q=i;
        i=1+i;
        Elapsedtime=etime(clock, cStTime);
    end
    vLLplot=vLLplot(1:q,1);
    vMVplot=vMVplot(1:q,1);
    mStdplot=mStdplot(1:q,:);
    mSP0plot=mSP0plot(1:q,:);
%     Likelihood=dLL
%     Elapsedtime=etime(clock, cStTime)
%     Loops=q
%     dDeltaLL
%     dDeltaPar
end
```

A.2 Forward-Backward algorithm

A.2.1 Forward-algorithm

```
function [mFwd, dLL]=fFwd( vG, mTP, vSP, iLen, iK)
    %Define return variables
    mFwd=zeros(iLen+1,iK);
    vA=vSP;
    dLL=0;
    mFwd(1,:)=vA;
    %Recursive forward calculations
    for j=2:iLen+1
        %Calculate next Alpha
        vA=vA * mTP * diag(vG(j-1,:));
        %Normalize
        vAsum=sum(vA);
        vA=vA/vAsum;
        %Update return values
        dLL=dLL+vAsum;
        mFwd(j,:)=vA;
    end
end
```

A.2.2 Backward-algorithm

```
function mBwd=fBwd( vG, mTP, iLen, iK)
    %Define return variables
    mBwd=zeros(iLen+1,iK);
    vB=ones(iK,1)/iK;
    mBwd(iLen+1,:)=vB';
    %Recursive backward calculations
    for j=1:iLen
        %Calculate next Beta
        vB=mTP*diag(vG(iLen-j+1,:))*vB;
        %Normalize
        vBsum=sum(vB);
        vB=vB/vBsum;
        %Update return values
        mBwd(iLen-j+1,:)=vB;
    end
end
```

A.2.3 State-dependent distributions

```
function mNcdf=fNcdf(vData, iLen, vStd, dMV)
    mNcdf=(ones(iLen,1)*(1./(vStd*sqrt(2*pi)))).*(exp(-((vData-dMV).^2
    return
end
```

A.3 DNM

```
function [dLL,vT]=DNM(iK,iLen,iSeries, iIter)
    [vData,mTP0,vSP0,dMV0,vStd0]=fBase(iK,iLen,iSeries);
    iK=numel(vSP0);
    iLen=numel(vData);
    mTP=mTP0;
    vSP=vSP0;
    dMV=dMV0;
    vStd=vStd0;
    vT=fT(mTP,vSP,dMV,vStd);
    while k < iIter+1
        mG=(ones(iLen,1)*(1./(vStd*sqrt(2*pi)))).*(exp(-((vData-dMV).^
        vAs=zeros(iLen,iK);
        vASum=sum(vSP,2);
        vAs(1,:)=vSP./repmat(vASum,1,iK);
        dLL=vASum;
        for i=2:iLen
            vAp=mG(i-1,:).*(vAs(i-1,:)*mTP);
            vASum=sum(vAp,2);
            vAs(i,:)=vAp./repmat(vASum,1,iK);
            dLL=dLL*vASum;
        end
        vAt=fAt(mTP,vSP,dMV,vStd);
        vG=1/vASum*sum(vAt,2);
        mB=mBt(mTP,vSP,dMV,vStd);
        mH=1/vASum*sum(mBT,2)+1/vASum^2*sum(mBT,2)*sum(mBT',2);
        vT=vT-inv(mH)*vG;
        [mTP,vSP,dMV,vStd]=finvT(vT);
        k=k+1;
    end
end
```

A.4 AIC and BIC

```
%iK=3;
iLoop=50;
%iIter=10;
dEpsLL=10^-5;
dEpsPar=10^-5;
iKtop=30;
iIter=100;
vBic=ones(iKtop,9);
vAic=ones(iKtop,9);
for t=1:9
    vData=fData(t,1500,50);
    iLen=numel(vData);
    vLLplotRC=zeros(iIter+1,1,iKtop);
    vMVplotRC=zeros(iIter+1,1,iKtop);
    mStdplotRC=zeros(iIter+1,iKtop,iKtop);
    mSP0plotRC=zeros(iIter+1,iKtop,iKtop);
    for iK=1:iKtop
```

```
        mTP0=rand(iK);
        mTP0=mTP0./repmat(sum(mTP0,2),1,iK);
        vSP0=ones(1,iK)/iK;
        dMV0=mean(vData);
        vStd0=(1:iK)/iK*2*std(vData);

        [~,~,~,~,~,dLL,vLLplot,vMVplot,mStdplot,mSP0plot]=EM(vDat
        vBic(iK,t)=log(iLen)*(1+2*iK+iK^2)-2*dLL;
        vAic(iK,t)=2*(1+2*iK+iK^2)-2*dLL;
        iK
    end
    series=t
end
```

Bibliography

[1] Bulla, J. (2006) *Application of Hidden Markov Models and Hidden Semi-Markov Models to Financial Time Series*: Munich Personal RePEc Archive.

[2] Böhning, J. et al. (1999) *Application of Hidden Markov Models and Hidden Semi-Markov Models to Financial Time Series*: Munich Personal RePEc Archive.

[3] Cappé, O., Moulines, E. and Rydén T. (2005) *Inference in Hidden Markov Models*: Springer Science and Business Media, Inc.

[4] Churchill, G. (1992) *Hidden Markov chains and the analysis of genome structure*: Computers & Chemistry 16 107115.

[5] Collings, I.B. and Rydén, T. (1998) *A new maximum likelihood gradient algorithm for on-line hidden markov model identification.*

[6] Dempster, A.P., Laird, N.M. and Rubin, D.B. (1977) *Maximum Likelihood from Incomplete Data via the EM Algorithm*: Journal of the Royal Statistical Society. Vol 39, pp. 1-38.

[7] Elliott, R. J., Aggoun, L. and Moore, J. B. (1995) *Hidden Markov Models*: Springe-Verlag New York, Inc.

[8] van Handel, R. (2008) *Hidden Markov Models, Lecture Notes.*

[9] Häggström, O. (2002) *Finite Markov Chains and Algorithmic Applications*: Cambridge University Press.

[10] Lee, S.W. and Park, H.S. (1998) *Hidden Markov Model For Off-Line Handwritten Character Recognition*: Pattern Recognition, 31 12 18491864.

[11] Luenberger (1984) *Linear and Nonlinear Programming*: Addison-Wesley, 2nd edition.

[12] Lystig, T.C. and Hughes, J.P. (2002) *Exact Computation of the Observed Information Matrix for Hidden Markov Models*: Journal of Computational and Graphical Statistics, Vol. 11, No. 3 (Sep., 2002), pp. 678-689.

[13] Norris, J.R. (1998) *Markov Chains*: Cambridge Series in Statistical and Probabilistic Mathematics

[14] MacDonald, I and Zucchini, W (1997) *Hidden Markov and Other Models for Discrete-Valued Time Series*: Chapman & Hall/CRC.

[15] MacDonald, I. and Zucchini, W. (2009) *Hidden Markov Models for Time Series*: Chapman & Hall/CRC.

[16] Mamon R.S. et al. (2007) *Hidden Markov Models in Finance*: Springer Science and Business Media, Inc.

[17] McLachlan, G. J. and Krishnan, T. (1997) *The EM Algorithm and Extensions*: John Wiley & Sons, Inc.

[18] Parzen, E. (1962) *Stochastic processes*: Holden-Day Inc.

[19] Rabiner, L. (2008)*A tutorial on hidden markov models and selecterd applications in speech recognition*: Institute of Electrical and Electronics Engineers.

[20] Turner, R. (2008) *Direct maximization of the likelihood of a Hidden Markov Model*: Elsevier Computational statistics & data analysis.

[21] Zhang, Y. (2004) *Prediction of financial time series with Hidden Markov Models*

TRITA-MAT-E 2014:12
ISRN-KTH/MAT/E—14/12-SE

www.kth.se