Parameter Update Schemes for Hidden Markov Models applied to Financial Returns

Sigfrid Forsberg

June 2022

This thesis was dedicated to investigating the use of different parameter update schemes for Hidden Markov models with time-varying parameters, with an emphasis on developing alternatives to the quasi-Newton step. The focus was on applications to financial returns, using data from the S&P-500 and the Nikkei index, and for comparison, a trial using synthetic data was also performed. Different properties of the parameter update schemes were explored, with Predictor-Corrector and Trust-Region based methods showing promise in comparison to the quasi-Newton methods previously tried. The Trust-Region method proved to be a more stable alternative, whereas the Predictor-Corrector method showed a significant smoothing of parameter adaptation which was not replicable by using the quasi-Newton method. Additionally, manipulating the norm of the Trust-Region method proved to be a versatile tool for e.g. calibrating the persistence of the hidden states without interfering with other parameter updates.

Contents

1 Introduction

1.1 Hidden Markov Models for Financial Returns

There are several stylized facts about financial returns. These range from an absence of autocorrelation and heavy tails to a slow decay in the autocorrelation of absolute returns and the clustering of volatility. These facts can be observed in a wide range of data and can be viewed as constraints to be put on models in order to reproduce financial returns accurately (Cont, 2000).

On the other hand, one of the main driving forces behind volatility in the stock market is the stage of the economic cycle. It has also been found that financial returns follow a similar rhythm to economic cycles, with short periods of high volatility followed by longer periods of low volatility (Hamilton, 1995).

The first published use of Hidden Markov models for modelling financial returns was made 1989 (Hamilton, 1989). In it, the author attempted to model the different economic regimes using a latent Markov chain. This was later expanded on by Rydén (1998) who showed that a Hidden Markov model with zero mean normal distributions could reproduce most of the stylized facts presented by Hamilton (1995). The one stylized fact that could not be reproduced was the slow decay of the autocorrelation function of absolute and squared returns.

Later on, Bulla (2011) showed that using conditional t-distributions showed some improvements over conditional Gaussian distributions. The main results were that the stylized facts presented by Hamilton (1995) were easier to recreate and the resulting model was more resilient to outliers.

Bulla (2011) also pioneered the use of Hidden Markov models for dynamic asset allocation purposes, using a two-state Hidden Markov model to decode the economic regimes and basing a trading strategy around the results. Many authors have done similar studies since, see e.g. Nystrup (2014) and Abrahamsen et al (2021).

1.2 Hidden Markov model with time-varying parameters

Holst, U. and Lindgren, G. (1991) were amongst the first to study recursive updating schemes for the Hidden Markov model. The proposed algorithm was a combination of a recursive maximum likelihood method used for independent data and the Expectation-Maximization (EM) algorithm.

Rydén (1997) later introduced the concept of score driven parameter updates for Hidden Markov models, using an algorithm with sequentially decaying update increments. This method was then applied to synthetic data and showed great improvements in asymptotic variance of the estimators when compared to the EM-style algorithm by Holst and Lindgren (1991).

The first recursively updated Hidden Markov model applied to financial data was made by Nystrup et al. (2017), in which the information matrix was approximated sequentially and a quasi-Newton optimization step was used in each update step and the step size was not decreased over time. The model managed to successfully reproduce the long memory of the squared daily returns, which is the stylized fact that has proven most difficult to reproduce using Hidden Markov models (Rydén et al. 1998). They also found that sequentially updating the parameters lead to a better density forecast as opposed to using a static model.

1.3 Generalized Autoregressive Score (GAS) and Trust-Region Optimization

Trust-region methods originate from work on numerical optimization for nonlinear least squares methods in the middle of the twentieth century, see Levenberg (1944) and Marquard (1963). Since then, Trust-region methods have become a staple in unconstrained numerical optimization theory and are covered in many textbooks (Sun W, Yuan Y. 2006).

The methods generally consist of restricting the step size of optimization algorithms based on their local performance, with a subproblem solved in each iteration step. There are numerous ways of solving the Trust-Region subproblem, ranging from the simple Cauchy-point calculation to the more complex conjugate-gradient method (Sun W, Yuan Y. 2006).

On the other hand, generalized autoregressive score (GAS) models were first defined by Creal et al. (2008, 2013). The 2013 publication started a fervor, with over 250 papers studying models that fall under the nowaday rather generous GAS classification to date. The framework generalizes score-driven parameter update steps using a non-parametric driving mechanism for scaling the score. They showed how a wide range of successful financial models such as the generalized autoregressive conditional heteroskedasticity (GARCH) models and the Beta-t-(E)GARCH can be found as special cases of the GAS framework. Notably though, only one paper has touched on the use of the GAS framework for Hidden Markov models (Nystrup et al, 2017).

1.4 Thesis Statement

Further building upon the foundation set by Nystrup et al. (2017), there is a case to be made for using different optimization schemes for parameter updating. Specifically, exploring more stable schemes could be very beneficial, since the simple quasi-newton method is prone to abrupt updates when new data is introduced. Coupled with the fact that the likelihood function generally has multiple local maxima, addressing this instability could increase performance by preventing the algorithm from leaving the global maximum through overshoot. Lastly, a scheme that is less sensitive to outliers could resolve some of the previously mentioned issues either through slower reacting parameter updates or through a possible decrease in moving window length.

The purpose of this thesis is to develop and evaluate different Hessian-based and Hessian-free numerical schemes as alternatives to quasi-Newton recursive estimators. One aspect will be to draw inspiration from solvers for ordinary differential equations such as linear multistep methods and use these methods as update mechanisms similar to the GAS models. Another will be using algorithms for numerical optimization such as trust region and line-search algorithms and apply them to sequentially update the model parameters.

The models will be evaluated on both S&P 500 and Nikkei index data, as well as on synthetic data. Using synthetic data allows models to be compared in different scenarios, such as rapid vs. progressive parameter changes. It also allows for a deeper understanding of how the algorithms behave when the underlying process is a true Hidden Markov model.

Another focus of this thesis will be related to manipulating the persistence of the states of the underlying Markov chain by using alternatives to the quasi-Newton method. Since the persistence of the states is directly linked to the usefulness of the model for e.g. state inference (Nystrup et al. 2020b), the ability to regulate it at no great cost to model fit could in and of itself constitute an improvement over the quasi-Newton method.

2 Theory

2.1 Financial returns

Modelling financial returns presents many challenges. Amongst these is the fact that the distribution of returns can usually be observed as having significantly greater tail risk than e.g. a Gaussian distribution, as well as a somewhat skewed appearance. As a result, a Gaussian model for financial returns is often a poor fit.

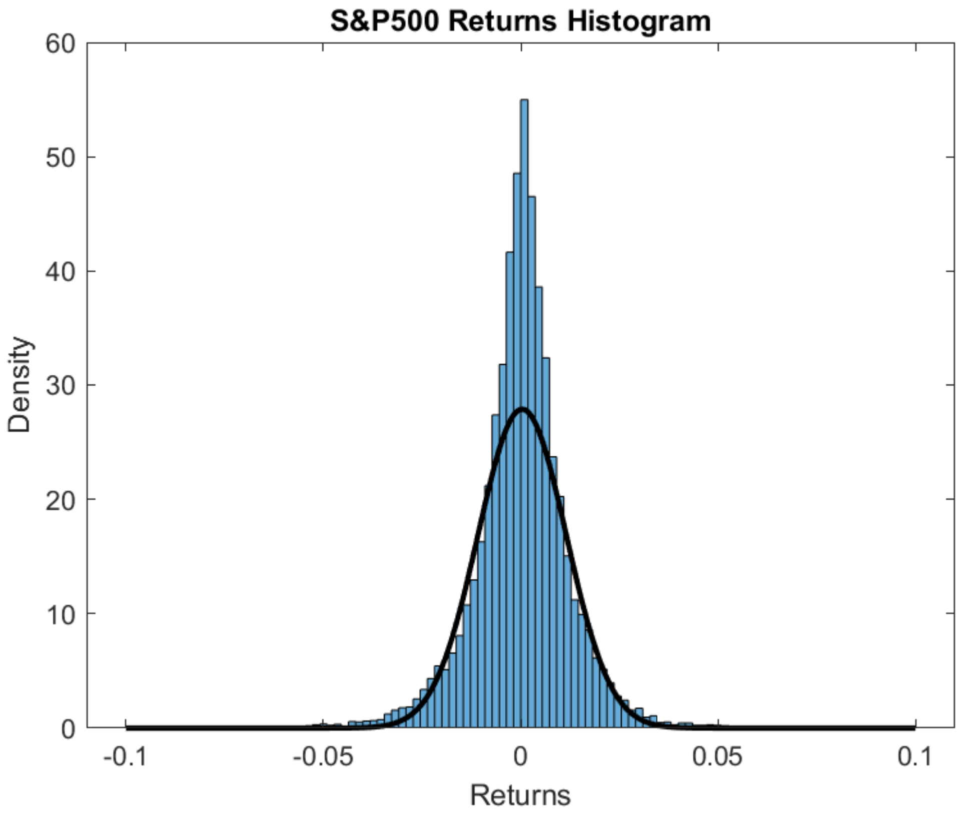

Figure 1: A histogram of the daily returns of the S&P 500 with an estimated Gaussian fit overlaid.

Instead of using one Gaussian distribution to describe financial returns, a mixture of Gaussian distributions can be used. Let e.g. $p(Y_t|Z = z_i) \sim N(\mu_i, \sigma_i)$ with some discrete distribution as a prior for Z be the conditional distribution for each financial return Y_t. In this case the financial returns are assumed to be Gaussian conditional on some stochastic variable, but the unconditional distribution of Y_t is typically not Gaussian. For this reason, issues such as leptokurtosis and skewedness can be partially solved by using Gaussian mixtures as opposed to a purely Gaussian model.

It is not necessary for Z in the above example to have identical distribution for each observation Y_t. The model can be extended to allow for a Markov chain to determine the conditional distribution for each Y_t at different times t. The result is a discrete time Hidden Markov model with Gaussian distributions conditional on the underlying Markov chain.

2.2 Hidden Markov Models

2.2.1 The Markov chain

The Markov chain is the foundation upon which the Hidden Markov Model is built. Suppose $\{X_1, X_2, ...\}$ is a discrete time stochastic process. Then the process $\{X_t\}$ is a discrete-time Markov chain if and only if it satisfies the Markov Property

$$P(X_t = x_t | X_{t-1} = x_{t-1}, X_{t-2} = x_{t-2}, ...) = P(X_t = x_t | X_{t-1} = x_{t-1}). \tag{1}$$

This definition implies that for each X_t, the distribution of X_{t+1} depends only on the value of X_t and t. Additionally, the Markov chain is said to be homogeneous if the distribution is independent of t. The transition probabilities $P(X_t = k | X_{t-1} = j) = \lambda^t_{j,k}$ form a state transition matrix

$$\Gamma_t = \begin{pmatrix} \lambda^t_{1,1} & \lambda^t_{2,1} & ... & \lambda^t_{s,1} \\ \lambda^t_{1,2} & ... & ... & \lambda^t_{s,2} \\ ... & ... & ... & ... \\ \lambda^t_{1,s} & \lambda^t_{2,s} & ... & \lambda^t_{s,s} \end{pmatrix} \tag{2}$$

and a distribution δ is said to be stationary if $\delta\Gamma_t = \delta$. Additionally, the Markov chain is transient if

$$P(X_{t+k} \neq i | X_t = i) > 0 \tag{3}$$

for all $k > 0$ and is said to be recurrent otherwise. In practice, a recurrent Markov chain regularly visits all states whereas the states of a transient Markov chain contains a subset of states that, as $t_0 \to \infty$, will never be visited again for all $t > t_0$ with probability 1.

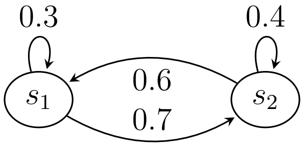

Figure 2: A 2-state homogeneous recurrent Markov chain with $\lambda_{1,1} = 0.3, \lambda_{2,2} = 0.4$

For a homogeneous Markov chain, the sojourn times for staying in one state is geometrically distributed with expected value

$$E[T_i] = \frac{1}{1 - \lambda_{i,i}} \tag{4}$$

where each T_i corresponds to the sojourn time of state i. For the inhomogeneous case, the distributions of the sojourn times do not have a general form. Consequently, a homogeneous Markov chain can be fully characterized by a state transition matrix Γ and an initial distribution δ_0 such that the ith element of δ_0 corresponds to $P(X_0 = x_i)$, whereas the characterization for inhomogeneous Markov chains includes one state transition matrix Γ_t for each t.

2.2.2 Hidden Markov Model

The discrete-time Hidden Markov model (HMM) is characterized by two main restrictions to two discrete-time stochastic processes. Firstly, suppose there exists an observable discrete-time stochastic process $\{Y_t\}$ and an unobservable discrete-time stochastic process $\{X_t\}$ taking values in $\{1, 2, ..., s\}$ such that

$$P(X_t | X_{t-1}, X_{t-2}, ...) = P(X_t | X_{t-1}), \tag{5}$$

that is, $\{X_t\}$ is assumed to be a Markov chain. Then the conditional distribution of Y_t given X_t is assumed to be independent of all previous observations $\{Y_{t-1}\}$. Explicitly,

$$P(Y_t | X_t, X_{t-1}, ..., Y_{t-1}, Y_{t-2}, ...) = P(Y_t | X_t). \tag{6}$$

Since the process $\{X_t\}$ is a (not necessarily homogeneous) Markov chain, it can be fully described by an initial distribution δ and a transition matrix Γ_t for each t. Additionally, due to (6) the distribution for each Y_t is known conditional on the state of the Markov chain. Hence Y_t can be characterized by one density $\pi_t(j)$ for each time point t and current state j. Summing up, an arbitrary discrete-time Hidden Markov model can be completely characterized by:

1. An s by s state transition matrix for each time point t: $\Gamma_t = \begin{pmatrix} \lambda^t_{1,1} & ... & \lambda^t_{1,s} \\ ... & ... & ... \\ \lambda^t_{s,1} & ... & \lambda^t_{s,s} \end{pmatrix}$
2. The collection of observational densities at each time step t, given $X_t = j$: $\pi_t(j) = p(Y_t | X_t = j)$
3. An initial distribution of the hidden state. Explicitly, the probability that the Markov chain initializes in state j: $\delta(j) = p(X_1 = j)$

2.2.3 Example: The 2-state Gaussian HMM

A relevant example of an HMM of this kind is a 2-state model with 2 different Gaussian conditional distributions,

$$Y_t \sim \begin{cases} p_{N,\mu_1,\sigma_1}, X_t = 1 \\ p_{N,\mu_2,\sigma_2}, X_t = 2 \end{cases} \tag{7}$$

where the latent Markov chain X_t is homogeneous with state transition matrix

$$\begin{pmatrix} \lambda_{1,1} & 1-\lambda_{1,1} \\ 1-\lambda_{2,2} & \lambda_{2,2} \end{pmatrix}. \tag{8}$$

with $(\lambda_{1,1}, \lambda_{2,2}) \in [0,1)^2$ as to avoid transience. In this case, the model is characterized by only 8 parameters: Distribution and transition probabilities $\theta = (\mu_1, \mu_2, \sigma_1, \sigma_2, \lambda_1, \lambda_2)$ and initial distribution $\delta_0 = (\delta_0^1, \delta_0^2)$. Additionally, if the Markov chain has initial distribution $\delta_0 = \delta$ equal to the stationary distribution of the Markov chain, the first two moments can be computed as

$$\begin{aligned} E[Y_t] &= \delta^1\mu_1 + \delta^2\mu_2 \\ &= \delta^1\mu_1 + (1-\delta^1)\mu_2 \end{aligned} \tag{9}$$

$$V[Y_t] = \delta^1\sigma_1^2 + \delta^2\sigma_2^2 + \delta^1(1-\delta^1)(\mu_1-\mu_2)^2. \tag{10}$$

Since the underlying Markov chain is homogeneous, the moments do not depend on t for the unconditional process. Conditional on the state of the Markov chain at different time points, it is further possible to compute 1-step predictions for expectation and variance of the observable process based on

$$E[Y_{t+1}|Y_t, Y_{t-1}...] = P(X_t = 1)(\lambda_1\mu_1 + (1-\lambda_1)\mu_2) + P(X_t = 2)(\lambda_2\mu_2 + (1-\lambda_2)\mu_1) \tag{11}$$

$$\begin{aligned} E[Y_{t+1}^2|Y_t, Y_{t-1}...] =& P(X_t = 1)(\lambda_1(\sigma_1^2 + \mu_1^2) + (1-\lambda_1)(\sigma_2^2 + \mu_2^2)) \\ &+ P(X_t = 2)(\lambda_2(\sigma_2^2 + \mu_2^2) + (1-\lambda_2)(\sigma_1^2 + \mu_1^2)). \end{aligned} \tag{12}$$

This follows from the fact that $E[Z^2] = \mu^2 + \sigma^2$ for $Z \sim N(\mu, \sigma^2)$.

2.3 Maximum Likelihood Estimation

Maximum likelihood estimation consists of finding the set of parameters θ such that the probability of obtaining the observed data given the parameters is maximized. Explicitly,

$$\hat{\theta} = \text{argmax}_{\theta} \ell_t(Y|\theta). \tag{13}$$

with

$$\ell_t(Y, \theta) = \ln(P(Y_1, Y_2, ...Y_t|\theta)) \tag{14}$$

is the log-likelihood of the data, given the model parameters θ. Thus, finding the maximum likelihood estimate of the model parameters is equivalent to maximizing the log-likelihood function with respect to these parameters.

When Y is the realization of a Hidden Markov model, this probability is not trivial to compute since although the underlying process has the Markov Property (5), the observable process Y typically does not. Instead, the distribution of Y_t depends on the entire history $Y_{t-1}, Y_{t-2}, ...$ which means that $\ln(P(Y_1, Y_2, ...Y_t|\theta))$ cannot be deconstructed into transition probabilities between observations similar to a Markov chain. Due to this fact, new observations cannot easily be added or removed without the need to recalculate the log-likelihood.

2.4 The modified forward algorithm

There are two main methods for estimating the parameters of a Hidden Markov model. The first is the expectation-maximization algorithm (EM), a robust algorithm that aims to estimate the parameters by maximizing a pseudo log-likelihood function where the true likelihood has been replaced with estimated expectations. The algorithm alternates between an expectation estimation step, where the expectation of the log-likelihood is computed and a maximization step, where the pseudo log-likelihood function is maximized.

An alternative to the EM algorithm is direct maximization of the log-likelihood function without the use of a pseudo log-likelihood function. Both the log-likelihood function, the score function and the information matrix can be computed in tandem using a modified forward algorithm similar to the one used in the EM algorithm. Once computed, the log-likelihood can be maximized using any number of numerical optimization algorithms.

Other less explored alternatives to the EM or direct maximization methods are primarily Bayesian Markov chain Monte Carlo methods (Rydén, 2008) and Jump Models (Nystrup et al., 2020a), although they will not be considered in this thesis.

2.4.1 The log-likelihood

The modified forward algorithm presented by Lystig & Hughes (2002) is recursive in nature, and is based on the traditional forward algorithm. In the forward algorithm, the collection of probabilities $\alpha_t(j) = p(Y_1, Y_2, ..., Y_t, X_t = j)$ are computed sequentially and summed over the possible states to acquire the log-likelihood. The issue with this approach is that $\alpha_t(j) \rightarrow 0$ exponentially, commonly causing numerical underflow issues. The revised algorithm instead computes the conditional probabilities

$$\bar{\alpha}_t(j) = p(Y_t, X_t = j | Y_1, Y_2, ..., Y_{t-1}). \tag{15}$$

Initializing $\bar{\alpha}_1(j) = p(Y_1, X_1 = j) = \pi_1(j)\delta(j)$, the conditional probabilities are computed recursively via

$$\bar{\alpha}_t(j) = \sum_{i=1}^{s} \frac{\bar{\alpha}_{t-1}(i)\pi_t(j)\lambda_{i,j}^t}{\sum\limits_{k=1}^{s} \bar{\alpha}_{t-1}(k)}. \tag{16}$$

The log-likelihood can then be computed as

$$l_T = \sum_{t=1}^{T} \Lambda_t \tag{17}$$

with $\Lambda_t = \sum\limits_{k=1}^{s} \bar{\alpha}_t(k)$.

An important tool in manipulating the log-likelihood function is the score, defined as the gradient of the log-likelihood function with respect to the parameters θ:

$$\nabla l(\theta) = (\frac{\partial l}{\partial \theta_1}, \frac{\partial l}{\partial \theta_2}, ..., \frac{\partial l}{\partial \theta_s}). \tag{18}$$

In the modified forward algorithm, the score can be computed parallel to the log-likelihood function using a similar approach as for the log-likelihood. For each time point t and parameter θ_k indexed from 1 to s, let

$$\psi_t(j, \theta_k) = \frac{\frac{\partial}{\partial \theta_k} p(Y_1, Y_2, ..., Y_{t-1}, Y_t, X_t = j)}{p(Y_1, Y_2, ..., Y_{t-1})} \tag{19}$$

with initialization at time $t = 0$

$$\psi_1(j,\theta_k) = \frac{\partial}{\partial\theta_k} p(Y_1, X_1 = j) = [\frac{\partial}{\partial\theta_k}\pi_1(j)]\delta(j) + \pi_1(j)[\frac{\partial}{\partial\theta_k}\delta(j)] \tag{20}$$

The components $\psi_t(j,\theta)$ can then be updated recursively alongside the $\bar{\alpha}_t(j)$ counterparts using the relation

$$\begin{aligned}\psi_t(j,\theta_k) = &\sum_{i=1}^{s}(\psi_{t-1}(i,\theta_k)\pi_t(j)\lambda_{i,j}^t + \bar{\alpha}_{t-1}(i)[\frac{\partial}{\partial\theta_k}\pi_t(j)]\lambda_{i,j}^t \\ &+ \bar{\alpha}_{t-1}(i)\pi_t(j)[\frac{\partial}{\partial\theta_k}\lambda_{i,j}^t])/(\Lambda_{t-1})\end{aligned} \tag{21}$$

with Λ_{t-1} as before. The derivation of the algorithm is essentially a repeated use of the chain rule. After all Ψ have been computed, the components of the score are computed from

$$\frac{\partial}{\partial\theta_k} l_T(\theta) = \frac{\sum_{i=1}^{s}\psi_T(i,\theta_k)}{\Lambda_T}. \tag{22}$$

The main drawback of using this algorithm for computing the score is that parameter transformations require analytical calculations of the derivatives with respect to the transition probabilities and conditional densities. This is especially true if different transformations are to be tried or different data sets are used which require multiple sets of parameter transformations. For this reason, a finite difference approach may be used as a practical alternative.

The last part of the modified forward algorithm consists of computing the information matrix:

$$I(\theta) = -E\left(\frac{\partial^2}{\partial\theta^2} l(\theta)\right) \tag{23}$$

Since this thesis is dedicated to methods where the Hessian is approximated rather than analytically computed, this is not used in any of the models presented and will not be explored further. The interested reader can find the explicit calculations in the works of Lystig & Hughes (2002).

2.4.2 Sequential updating of parameters

Since the dynamics of the economy evolves over time, it is unlikely that a model with fixed parameters is able to accurately describe an index stretching several decades.

Therefore, methods for sequentially updating the parameters of the Hidden Markov model when new data is introduced could be an improvement over e.g. updating parameters after a set amount of time.

One method is to use a rolling window where each data point contributes equally to the log-likelihood and new data replaces old data over time. Although this method has some artificial properties caused by not weighting the datapoints, it has some advantages in ease of implementation since the log-likelihood does not need to be weighted.

Another method is to use an exponential weighting, where the most recent data is assigned a higher weight than older data. One of the advantages is that using an exponential weighting allows for a longer window length while still being able to quickly adapt to new data points. On the other hand, quick adaptation based on a few data points is a double edged sword in that it makes the model more vulnerable to outliers. Nevertheless there is some evidence to support that an exponential weighting is superior to a rolling window in some circumstances (Nystrup et al., 2017).

As for methods used in updating the parameters for each time step, there is no consensus on the best way to go about. The only numerical scheme that has been tried is a Quasi-Newton method with approximated information matrix (Nystrup et al., 2017) and while the results are encouraging, there is possibly room for improvement.

2.5 Hessian based numerical optimization

The school of numerical analysis has a plethora of algorithms for maximizing functions. One of the simplest and most well known such algorithm is the Newton-Rhapson method, which utilizes both the gradient and hessian matrix of the target function.

Suppose $\nabla f(x)$ and $H(x)$ the gradient and hessian of a twice differentiable function $f : R^n \to R$ and suppose one is interested in finding the global minima of the target function $f(x)$. One way of accomplishing this is finding x such that $\nabla f(x) = 0$, which reduces the problem to finding roots for the gradient of $f(x)$.

Let x_0 be an initial guess at a minima for $f(x)$. Then, by Taylor

$$f(x_0 + h) \approx f(x_0) + \nabla f(x_0)h + \frac{1}{2}h^T H(x_0)h \tag{24}$$

for a small increment h. Differentiation with respect to h then yields

$$\nabla f(x_0 + h) = \nabla f(x_0) + Hh. \tag{25}$$

Then, setting $\nabla f(x_0 + h) = 0$ and solving for h yields the optimal improvement

$$h_0 = -H^{-1}\nabla f(x_0). \tag{26}$$

The Newton-Rhapson algorithm iterates this method for $t \to t+1$ by $x_{t+1} = x_t + h_t$. The algorithm achieves quadratic convergence, but is unable to differentiate between local and global extremes. In addition, the algorithm is not guaranteed to converge if the initial guess is too far from the global minima.

One of the great drawbacks of Newton-Rhapson is that it requires the Hessian to be explicitly computed in each step of the iteration process. Since the Hessian is often difficult to compute analytically, an approximation is often used instead of the true Hessian. Methods using Hessian approximations rather than true Hessians are called Quasi-Newton methods. Although it is convenient not to have to compute the Hessian, the approximation often comes at the cost of quadratic convergence.

There are a few different methods for approximating and sequentially updating the Hessian during optimization. The one approach that will be covered in this thesis is the symmetric rank-1 (SR(1)) method (See e.g. Sun W, Yuan Y. (2006)). The idea is to additively update the Hessian by some symmetric rank-1 matrix $E = uv^T$, where u and v are 2 column vectors. Let $y_t = \nabla f(x_{t+1}) - \nabla f(x)$, $h_t = x_{t+1} - x_t$ and $B_t = H_t^{-1}$ be the inverse Hessian. Then define

$$B_{t+1} = B_t + uv^T. \tag{27}$$

Using result (25), it follows that

$$h_t = (B_t + uv^T)y_t \tag{28}$$

and consequently

$$v^T y_t u = h_t - B_t y_t. \tag{29}$$

If B_t does not satisfy $h_t = B_t y_t$, (27) can be rewritten to

$$B_{t+1} = B_t + (v^T y_t)^{-1}(h_t - B_t y_t)v^T. \tag{30}$$

Further, setting $v = h_t - B_t y_t$ ensures symmetry of the inverse Hessian which is required, and the resulting expression is

$$B_{t+1} = B_t + \frac{(h_t - B_t y_t)(h_t - B_t y_t)^T}{(h_t - B_t y_t)^T y_t} \tag{31}$$

which is the formula for the Symmetric Rank-1 update. An important note is that while the update preserves symmetry, it does not guarantee that the resulting approximation is positive definite.

2.5.1 Stability issues

Whilst the Hidden Markov model is a very powerful tool, it has some limitations especially when considering parameter estimation. When the parameters are assumed to be static these limitations are not too bothersome but when the parameters are sequentially updated, especially based on data which heavily deviates from the previous observations, issues can arise.

The first issue is that the log-likelihood generally has several local maxima, which means that each parameter update runs the risk of moving away from the global maxima towards a local one. This is especially true when considering the impact outliers can have on financial data, causing amongst other things large overshoots in parameter updates using a simple quasi-newton update (Nystrup et al 2017).

The second issue is that constrained parameters may require transformation as to avoid issues with convergence. For a Hidden Markov model with conditional normal distributions, this issue arises for both the variances and transition probabilities of the states. The result is that the choice of transformation heavily impacts the behavior of the algorithm. This is both a blessing and a curse, seeing as there is both an addition of freedom and a difficulty in direct model comparison since there are more factors to consider for each model.

It should also be noted that even with appropriate transformations (e.g. a probit transform for the transition probabilities), severe problems can still arise if e.g. one transition probability is set very close to 1 or 0 in some time step since this effectively makes the underlying Markov chain transient.

2.6 Possible improvements over conventional quasi-newton correction

2.7 Trust Region Methods

One of the aims of this paper is to explore possible improvements to the quasi-newton step, especially when sequentially updating model parameters in dynamic systems where the parameters may be time-varying. This could lead to time points where the quadratic approximation used in the quasi-newton method is rather poor. Since the traditional quasi-newton method does not evaluate the fit of the quadratic approximation, one possible improvement is the implementation of a Trust-Region method.

The idea behind a trust-region method is simple: When the quadratic approximation $f(x+h) \approx f(x)+h\nabla f(x)+hH(x)h^T$ is good for a certain stepsize $||h||$ and time step t, this can be exploited by increasing the allowed step size for the next time step, allowing for the possibility of a larger improvement. On the other hand, if the approximation is poor, the allowed stepsize $||h||$ can be decreased in order to better the predictability of the optimization since the smaller the stepsize, the more accurate the quadratic approximation is. This evaluation is usually done by comparing the improvement in function value to the *expected* improvement, using the quadratic approximation. Explicitly, a trust region subproblem with given maximum allowed stepsize Δ at time t can be expressed as

$$\min m_t(h) = \min_h f(x_t) + \nabla f(x_t)h + \frac{1}{2}h^T H(x_t)h \tag{32}$$

subject to $||h|| \leq \Delta$. After this subproblem is solved, the fit of the quadratic approximation can be evaluated using the quotient

$$\rho_t = \frac{f(x_t + h) - f(x_t)}{m_t(h) - m_t(0)} \tag{33}$$

The measurement ρ_t is then compared to several thresholds and the size of the trust region is adjusted accordingly. An example of how such an algorithm can be constructed is the following: Select starting value x_0 for $t = 0$, multipliers $k_1 = 0.25, k_2 = 2.0$ and thresholds $\eta_1 = 0.1, \eta_2 = 0.25, \eta_3 = 0.75$. Additionally, select some initial trust region Δ_0 and a maximal allowed trust region Δ_M. Then iterate using the following scheme for each time step t:

```
for $t = t_0, t_0 + 1, t_0 + 2...$ do
    Solve the trust region subproblem.
    Obtain and evaluate $h_t$ and $\rho_t$
    if $\rho_t > \eta_3$ and $||h|| \approx \Delta_t$ then
        The quadratic approximation is good and the size of the trust region
          can be increased:
        $\Delta_{t+1} = k_2 \Delta_t$ and $x_{t+1} = x_t + h_t$.
    else
        if $\rho > \eta_2$ then
            The quadratic approximation is reasonable, but not stellar. the size
              of the trust region is not increased or decreased:
            $x_{t+1} = x_t + h$.
        else
            if $\rho > \eta_1$ then
                The quadratic approximation is poor and the size of the trust
                  region should be reduced:
                $\Delta_{t+1} = k_1 \Delta_t$ and $x_{t+1} = x_t + h_t$.
            else
                if $\rho < \eta_1$ then
                    The quadratic approximation is too poor for a parameter
                      update to be performed. The size of the trust region should
                      be reduced:
                    $\Delta_{t+1} = t1 \Delta_t$.
                else

                end
            end
        end
    end
end
```

2.7.1 Solving the Trust-Region Subproblem. Steihaugh CG.

There are several ways to go about in solving the trust-region subproblem. In this thesis, the focus will be on one of the most commonly used: The Steihaug-Taut Conjugate-Gradient method. The method aims to solve the trust-region subproblem (25) using an iterated Hessian-free approach, but utilizing both the gradient and approximated Hessian of the previous time point. For a general reference to Steihaug-Taut and general CG-methods, see e.g. Sun W, Yuan Y (2006).

The method divides the subproblem into three distinct cases. Let $d = -f$ be the negative gradient of f and H be the current hessian approximation. If $d^T H d > 0$, the solution is found using the direction of steepest descent and is put on the boundary if the solution lays outside the trust region. If instead $d^T H d \leq 0$, the direction has a negative curvature and the solution is taken as the point on the boundary obtained by following the direction of d.

Put into an algorithm, the method finds a solution s to the subproblem in the following way:

```
Initialization:
A tolerance $\epsilon > 0$ is selected as well as an $r_0 = f(x_t)$ and $d_0 = -r_0$ and $h_0 = 0$.
if $||r_0|| < \epsilon$ then
    The algorithm is terminated and the solution is taken as $s = h_0 = 0$.
else
    Begin iterating to find a good solution:
    while $||h_{j+1}|| > \Delta$ do
        if $d^T Hd \leq 0$ then
            A $\tau > 0$ is found such that $h = h_0 + \tau j$ is on the boundary of the
              trust region, i.e. $||h|| = \Delta$.
            The algorithm is terminated with $s = h$.
        else
            Set $h_{j+1} = h_j + \frac{r_j^T r_j}{d_j^T H d_j} d_j$.
            if $||h_{j+1}|| \geq \Delta$ then
                Transform $h_{j+1}$ back onto the border of the trust region in the
                  negative gradient direction:
                Set $s = h_{j+1} - \tau d_j$ such that $||s|| = \Delta$.
                Terminate the algorithm.
            else
                Set $r_{j+1} = \frac{r_j^T r_j}{d_j^T H d_j} H d_j$.
                if $||r_{j+1}|| < \epsilon$ then
                    Set $s = h_{j+1}$.
                    Terminate the algorithm.
                else
                    Update $d_j$ by $d_{j+1} = -r_{j+1} \frac{r_{j+1}^T r_{j+1}}{r_j^T r_j} d_j$.
                end
            end
        end
    end
end
```

An important thing to note is that if a Hessian approximation which lacks positive definiteness is used, Steihaugs CG method still works. For this reason, a SR(1) update can be used without fear of the algorithm breaking due to this issue.

2.8 Generalized Autoregressive Score Models (GAS)

2.8.1 Definitions

An alternative to using a Hessian approximation is to simply not use a Hessian at all, and instead let the model parameters be updated by functions of the likelihood and score. This is essentially what the broad class of models called Generalized Autoregressive Score (GAS) models attempt to do, although some Hessian based models also fall under this umbrella term. In fact, many models from different areas of optimization theory and especially in econometrics can be found as special cases of the GAS framework.

Suppose $p(Y_t, Y_{t-1}, ...|\theta_t)$ is the probability of the observations $Y_t, Y_{t-1}, ...$ given the parameters θ_t at time t. Then the GAS(p, q) model is characterized by the relation

$$\theta_{t+1} = \omega + \sum_{j=1}^{q} \beta_j \theta_{t+1-j} + \sum_{i=1}^{p} \alpha s_{t-i+1} \tag{34}$$

with ω, α_i and β_j as dimension appropriate vector and matrices respectively. Additionally the "driving mechanism" s_t is updated according to

$$s_t = S_t \dot{\nabla} l(\theta_t) \tag{35}$$

with S_t as some matrix function of the time varying parameters θ and all available information at time t $S(t, \theta_t, F_t)$.

It can be noted that there are some similarities between this method and the previously discussed quasi-Newton methods. In fact, Newton's method can be found as a special case of the GAS(1,1) model with $S_t = -H_t$, as can many other popular parameter update schemes. Due to the flexibility in selecting s_t, orders p, q and coefficient matrices, there are numerous schemes with the potential to outperform the quasi-Newton method.

2.8.2 Parallel between ODE-solvers and optimization

Methods for solving ordinary differential equations numerically and numerical optimization are closely related concepts. For this reason, methods used in one setting can have applications in the other.

Suppose that the $\frac{\mathrm{d}y}{\mathrm{d}t} = f(y, t)$ of y is some function of y and t can be evaluated and suppose that some initial value $y_0 = y(t_0)$ is given. The value $y(t)$ of some arbitrary t can then be approximated using numerical integration by discretizing the interval (t_0, t) and using numerical integration methods in each discretized time step. In numerical analysis, this kind of initial value problem is greatly important (see e.g. Griffiths, Higham 2010 as a general reference for numerical integration methods) and as a result, numerous numerical schemes exist and can be used to find an approximation of $y(t)$.

It is clear that there are parallels between this numerical integration problem and GAS models. In fact, so long as $q \geq 1$ and $\beta_1 = 1$ in (34), approximations of $y(t+1)$ with $t_0 = t$ can be interpreted as a parameter update step in the GAS model if the time descritization is one whole time step.

Take e.g. $\theta_{t+1} = y(t+1)$, $t_0 = t$. A numerical solution to the initial value problem then finds an approximation of θ_{t+1} given θ_t and the relation

$$\frac{\partial \theta}{\partial t} = f(\theta, t). \tag{36}$$

If e.g. the numerical method taken is Euler's method

$$\theta_{t+1} = \theta_t + f(\theta_t, t), \tag{37}$$

and $f(\theta, t) = -H(\theta_t)^{-1}\ell(\theta_t)$, the resulting model is identical to the previously covered Newton-Rhapson method for updating parameters. In fact, substituting θ_{t+1} in the GAS equation and setting $f(\theta, t) = \theta_{t+1} - \theta_t$ yields

$$\begin{aligned} f(\theta, t) &= \theta_{t+1} - \theta_t \\ &= \omega + \sum_{j=2}^{q} \beta_j \theta_{t+1-j} + \sum_{i=1}^{p} \alpha s_{t-i+1} \end{aligned} \tag{38}$$

and using Euler's method recreates the corresponding GAS method as the solution to the initial value problem. Thus, substituting Euler's method for an alternative opens up the door to potential improvements over the conventional GAS models in the update step.

It should be noted that there is a key difference between traditional ODE-solvers and numerical optimization. Specifically, in the traditional initial value problem, the target function can only be computed in the initial value whereas in optimization it can be computed for any value. In practice this means that progress can be monitored by evaluating the target function, ensuring that each step brings an increase in function value. For this reason, some ODE-algorithm which do not utilize this resource when performing optimization will be suboptimal when compared to corresponding modified versions.

It should also be noted that for some models, the log-likelihood is computed parallel to the score and hence there is no computational cost associated with the evaluation.

Since this is the case for the modified forward algorithm, this method can be favorably implemented for the Hidden Markov model.

2.8.3 Linear multistep methods

Linear multistep methods is a class of numerical schemes for use in solving ordinary differential equations. As opposed to one-step methods where previous evaluations of the gradient and target function are discarded, this class of methods aims to increase performance by storing and utilizing the previous evaluations. Linear multistep methods impose a linear restriction to the update, resulting in schemes of the form

$$y_t = \sum_{i=1}^{p} \alpha_i y_{t-i} + \sum_{j=0}^{q} \beta_j f(y_{t-j}, t-j) \tag{39}$$

with $f(t, y_t) = \frac{\mathrm{d}y}{\mathrm{d}t}$ as before.

A linear multistep method can be either explicit or implicit. An explicit method utilizes only computations up to $t-1$ in the computation of y_t whereas an implicit method can use computations up to and including time t. Hence the method is called explicit if $\beta_0 = 0$ and implicit otherwise. For the implicit case, methods must be employed to solve the equation for y_t since it then appears in both the right hand side and the left hand side. Because $f(y_t, t)$ seldom is a simple function, this is often done using numerical approximation.

For the purpose of maximizing the log-likelihood, there are advantages to employing implicit methods over explicit ones. Since stability issues have been observed in the past when employing explicit methods it is possible that an implicit scheme could improve stability in the same way as implicit numerical schemes have been used to solve stiff differential equations in the past, where alternative methods have proven to be too unstable (Nystrup et al., 2017).

2.8.4 Predictor-Corrector Scheme

One way to extend the family of linear multistep methods is to include functions of predicted values y_{t+1} in an additional step of computations. One such method is Heun's method which contains two steps:

$$\begin{aligned} (1):\quad & \tilde{y}_{t+1} = y_t + f(y_t, t) \\ (2):\quad & y_{t+1} = y_t + \frac{1}{2}(f(y_t, t) + f(\tilde{y}_{t+1}, t+1)) \end{aligned} \tag{40}$$

This corresponds to the special case of the linear multistep family for which $\beta_0 = \beta_1 =$

1/2 and $\alpha_1 = 1$, where the value of y_{t+1} on the right hand side has been approximated using an Euler step.

One interesting aspect of the predictor-corrector scheme is that if the approximation y_{t+1} is found to be inaccurate, the approximation can be improved by iterating the correction step of the algorithm. This is especially useful in an optimization setting where the target function can be monitored since this gives a gauge of the accuracy of the estimation.

2.9 A brief mention: Line-search algorithms

A third approach to the optimization problem is the use of a line-search algorithm, see e.g. Sun and Yuan, (2006) as a general reference. In contrast to a trust-region approach where the objective function is optimized over a domain centered around the current parameter estimates in each time step, the family of line-search algorithms first selects an improvement direction p_t, and then attempts to find an optimal step size α for which $f(\theta_t + \alpha p_t)$ is maximized.

One of the advantages of this kind of algorithm is that it only requires the one initial computation of the score (and if required, Hessian), after which only the log-likelihood needs to be computed in each time step. This contrasts to the Predictor-Corrector scheme, where each correction step requires an additional computation of the score.

Additionally, the algorithm can be used in any case where the step size is not restricted. This can be very beneficial when e.g. stability is an issue and large steps can cause a great decrease in log-likelihood.

With $d_t = -\nabla f(x_t)$, one simple implementation is the backtracking line-search. The method is based on the fact that, for each $\beta \in (0,1)$, $\rho \in (0,1/2)$, $\tau > 0$ and $d_t = -\nabla f(x_t)$, there exists at least one integer $m > 0$ such that

$$f(x_t) - f(x_t + \beta^m \tau d_t) \geq \rho \beta^m \tau ||d_t||^2. \tag{41}$$

Using this relation, the following algorithm can be implemented in just a few steps:

```
Set $\alpha_0 = 1$ and $w \in (0,1)$.
for $t = t_0, t_0 + 1, t_0 + 2...$ do
    Compute $f(x_t + \alpha_t d_t)$.
    if $f(x_t + \alpha_t d_t) \leq f(x_t) - \rho \alpha d_t^T d_t$ then
        Set $h = \alpha d_t$.
        Terminate the algorithm.
    else
        Set $\alpha_{t+1} = w \alpha_t$.
    end
end
```

3 Data Selection

3.1 Synthetic Data

In the evaluation of different models it is important to use synthetic data as well as real data. A few of the more interesting aspects that need to be evaluated are how well the model adapts to sudden as well as progressive changes, especially when it comes to changes in volatility. For this reason, the synthetic data was generated using a homogeneous Markov chain of 9000 data points and with time varying parameters μ_2 and σ_2.

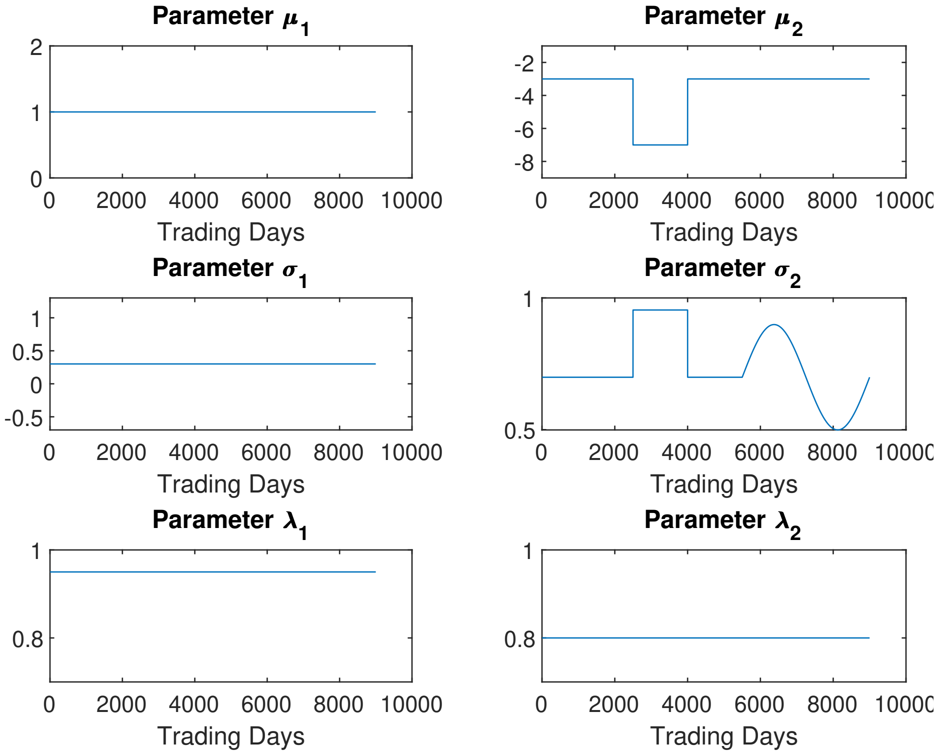

Figure 3: Parameters of the synthetic data used, generated from 9000 data points.

Here, a sudden change in μ_2 is seen at $t = 2500$ going from $\mu_2 = -3$ to $\mu_2 = -7$, after which it returns to the initial value at $t = 4000$. Simultaneously, the σ_2 parameter is adjusted from $\sigma_2 = 5$ to $\sigma_2 = 9$ and back in a similar fashion. After an additional 1500 datapoints, σ_2 follows a gradual adjustment according to a sinusoidal function, with maximum and minimum values obtained as $\sigma_2^{\text{max}} = 8$ and $\sigma_2^{\text{min}} = 2$.

3.2 Real Index data

3.2.1 S&P-500

The first data set chosen consists of daily close data from the S&P 500 taken from its 1978 until 2022 representing 11138 trading days was selected. There are several reasons for including this particular data set. Importantly it is one of the most well studied indices in the world, containing stocks from the 500 largest companies listed on the US stock exchange. The fact that this index is so well studied gives context to acquired results. Secondly, the data set includes the infamous date Black Monday (october 19, 1987) which caused issues in previous similar models. How the models presented above perform when encountering such an event is of great interest, and so an index which contains several years of data both before and after 1987 is a desirable inclusion.

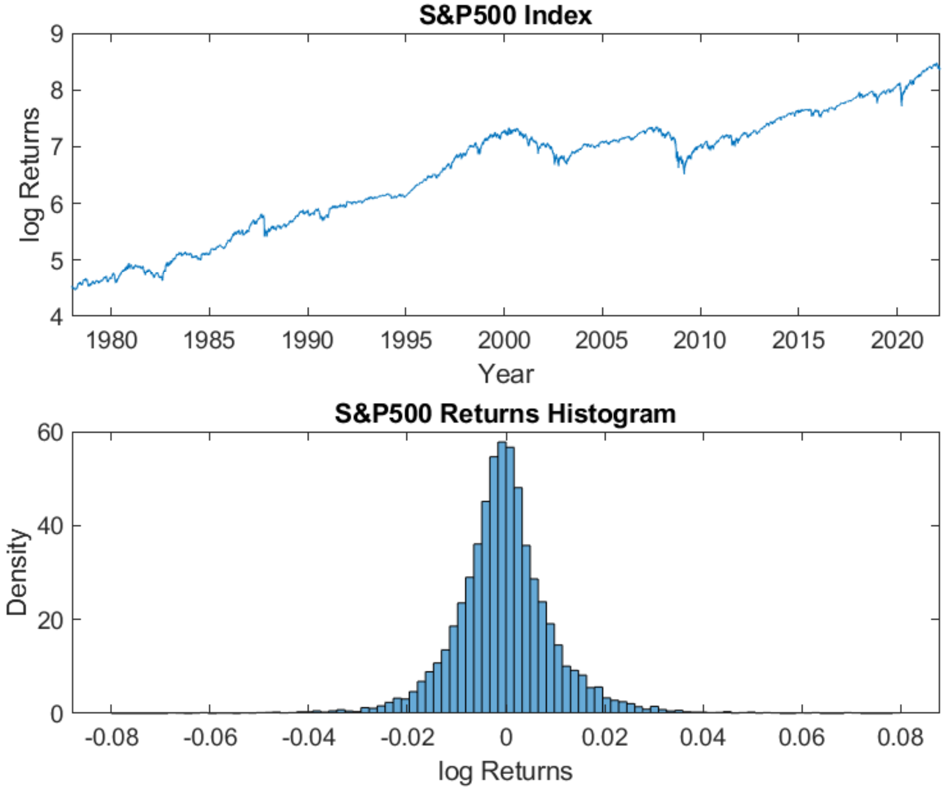

Figure 4: The S&P 500 Index

3.2.2 Japanese Nikkei

In order to contrast the previously mentioned S&P-500 data set, the japanese Nikkei 225 index is taken from the beginning of 1970 until 2022. The importance of including this data set is mainly related to the difference in behavior between it and the S&P-500,

with the chief differences being the increased variance of the Nikkei data set as well as the fact that the japanese crisis in 1991 has resulted in a price high which the index has not reached again to date. This is clearly different from the S&P 500 index, and so it is interesting to see whether there is a difference in model performance between the two data sets or not.

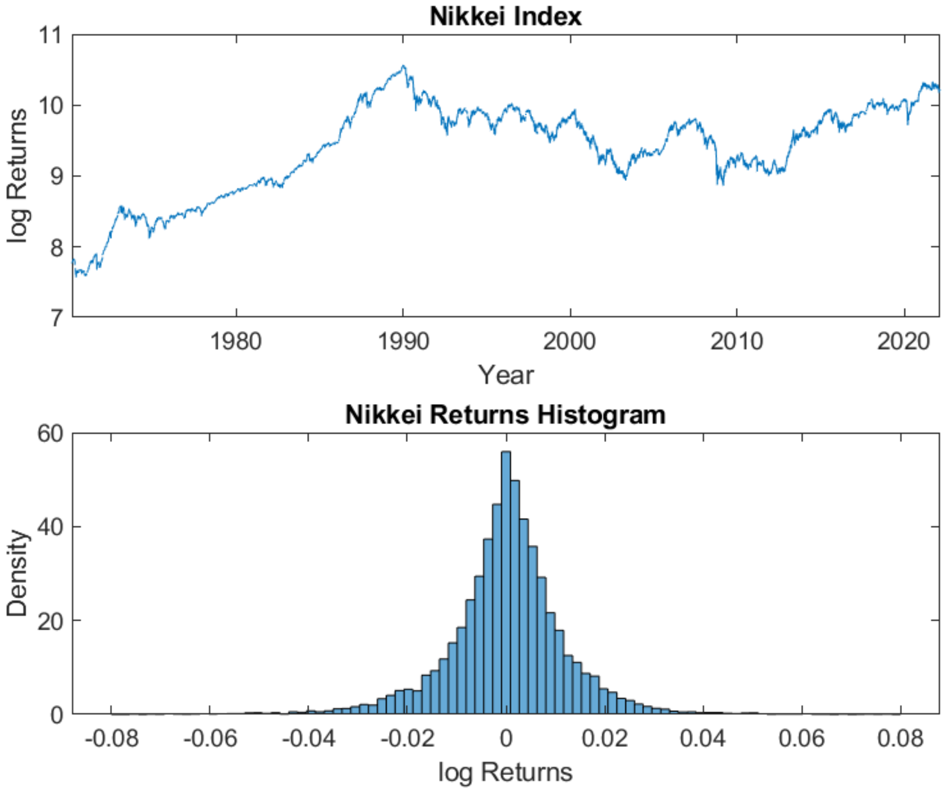

Figure 5: The Nikkei Index

4 Method

4.1 Models

The main objective of this paper is to explore the application of different numerical schemes to the sequential updating of model parameters, given new data. In order to do this, 4 main models were identified as potential candidates:

- Trust-Region method based on allowing the step length to be increased and decreased depending on how well the trust-region subproblem is solved.
- Predictor-Corrector method based on iterating correction steps until the taken step increases the likelihood function. This is a Hessian-free approach.
- Backtracking line-search algorithm based on the quasi-Newton method.
- Score driven backtracking line-search algorithm.

4.1.1 The Trust Region Model

There are a few important reasons for choosing a trust-region method for updating the model parameters when modelling financial data. Firstly, the quadratic approximation that the Newton-Rhapson uses could become poor when extreme points are included into the moving window, which could lead to problems such as overshoots in parameter updates and similar issues. Using a trust-region could mitigate this issue by shrinking the allowed step size when the accuracy of the quadratic approximation becomes poor. Secondly, since some parameters are easier to estimate than others in financial time series, it is interesting to see whether the trust-region method will adapt to this by prioritizing updates in e.g. volatility over mean. Thirdly, the Trust-Region method for optimization is practical for reducing the sensitivity of certain parameters.

Given these motivations, the first model presented in this thesis is a trust-region model with an SR(1) update for the Hessian and using Steihaug CG for solving the trust-region subproblem.

Given a moving window length N and a maximum step size Δ, the algorithm proceeds as follows:

```
Select initial parameters $\theta_0$ using an iterated solver to maximize the
  log-likelihood function provided by the modified forward algorithm, using the
  first $N$ data points. For this thesis, a Nelder-Mead algorithm was used for
  this purpose.
Initialize the hessian using a finite-difference approach to obtain an initial
  estimate of $H_0$. Set an initial trust region size $\Delta_0$.
for $t = t_0, t_0 + 1, t_0 + 2, ...$ do
    Update the parameters $\theta_t \rightarrow \theta_{t+1}$ by solving the Trust-Region subproblem
      by applying the Steihaug CG method, taking into account the last $N$ data
      points.
    Update $H_t \rightarrow H_{t+1}$ using the SR(1) method and allowed step size
      $\Delta_t \rightarrow \Delta_{t+1}$ according the trust-region method, with parameters
      $\eta_1 = 0.1, \eta_2 = 0.25, \eta_3 = 0.75$.
end
```

As for the norm used, the regular L_2 norm was used in addition to different norms constructed by

$$||\theta||_z = ||A_N\theta||_2, \tag{42}$$

with $A_N = \frac{A}{|A|^{1/6}}$ as a diagonal matrix with weights $w_i = A_{i,i}$ corresponding to the parameters $\theta = (\mu_1, \mu_2, \sigma_1, \sigma_2, \lambda_1, \lambda_2)$ and $|A_N| = 1$ by design:

$$A = \begin{pmatrix} w_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & w_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & w_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & w_4 & 0 & 0 \\ 0 & 0 & 0 & 0 & w_5 & 0 \\ 0 & 0 & 0 & 0 & 0 & w_6 \end{pmatrix} \tag{43}$$

.

By adjusting the different weights w_i, the model can allowed to vary some parameters more than others. This could potentially be beneficial when e.g. restricting the transition probabilities λ_1, λ_2 or variance parameters, since these have been shown to vary quite a lot throughout the sample (Nystrup et. al, 2017).

4.1.2 The Predictor-Corrector Model

Another objective of this thesis is to explore Hessian-free numerical schemes for the purpose of updating the parameters of the Hidden Markov model. The reason for this is largely inspired by the success of the GAS models and an interest in whether an application to Hidden Markov models is possible.

For this reason, the second model presented in this thesis is inspired by the traditional Predictor-Corrector scheme. The hope is that a model based on Predictor-Corrector could reduce the overfitting issues, and increase stability without sudden jumps and larger than reasonable variance predictions.

Let $\ell(\theta)$ be shorthand notation for $\ell(Y_t, Y_{t-1}, ...|\theta)$. Then the algorithm uses 1 Euler step for predicting $\tilde{\theta_{t+1}} = h\ell(\theta_t, t)$ using constant step size h and then iterates correction steps until the log-likelihood has increased by evaluating $\ell(\tilde{\theta_{t+1}})$ for each iteration. If the algorithm fails to find a positive increase, the parameter is not updated and $\theta_{t+1} = \theta_t$:

```
Set step size $h$ based on the data provided.
for $t = t_0, t_0 + 1, t_0 + 2, ...$ do
    Compute $\tilde{\theta}_{t+1} = \theta_t + \nabla\ell(\theta_t)$.
    Compute $\hat{\theta}_{t+1} = \theta_t + \frac{1}{2}\Big(\nabla\ell(\theta_t) + \nabla\ell(\tilde{\theta}_{t+1})\Big)$
    while $\ell(\hat{\theta}_{t+1}) < \ell(\theta_t)$ do
        Compute a new $\hat{\theta}^{\text{new}}_{t+1} = \theta_t + \frac{1}{2}\Big(\nabla\ell(\theta_t) + \nabla\ell(\hat{\theta}_{t+1})\Big)$
    end
    Return $\theta_{t+1} = \hat{\theta}_{t+1}$.
end
```

One drawback of this type of algorithm can be that the step size h may need to be exceedingly small, since the explicit Euler step can yield a hefty decrease in the log-likelihood which may not always be corrected enough in the correction step. However, since one central aim of this model is to decrease sensitivity it could be that a small step size is appropriate regardless.

4.1.3 Backtracking line-search: A brief mention

As previously mentioned, one issue with the Predictor-Corrector approach is that it significantly restricts the step size used. Additionally, for even moderate step sizes the number of correction steps required to acquire an increase in the log-likelihood may be excessive.

If this becomes too burdensome, an alternative score-driven approach is to use a backtracking line-search, starting with a large step size in the direction of the negative gradient and scaling it back until an improvement in the log-likelihood is observed.

The backtracking line-search algorithm used in this thesis follows precisely the steps detailed in ”A brief mention: Line-search algorithms”:

```
Set $\alpha_0 = 1$ and $\rho = 0.499$ and $w = 0.7$.
for $t = t_0, t_0 + 1, t_0 + 2...$ do
    Compute $\ell(\theta_t + \alpha_i \nabla \ell(\theta_t))$.
    for $i$ = 1, 2, ..., $i_{max}$ do
        if $\ell(\theta_t + \alpha_i \nabla \ell(\theta_t)) \leq \ell(\theta_t) - \rho\alpha \nabla \ell(\theta_t)^T \nabla \ell(\theta_t)$ then
            Set $\alpha_{i+1} = w\alpha_i$.
        else
            Set $\theta_{t+1} = \theta_t + \alpha_i \nabla \ell(\theta_t)$.
        end
    end
end
```

Additionally, a line-search for Newton's method was also implemented. This line search algorithm however was only applied when the initial step resulted in a negative contribution to the log-likelihood. Hence the only difference between this method and the regular quasi-Newton presented by Nystrup et al (2017) is that the model temporarily reduces step size if there is no improvement in log-likelihood:

```
Set $\alpha_0 = 1$ and $\rho = 0.499$ and $w = 0.7$.
for $t = t_0, t_0 + 1, t_0 + 2...$ do
    Compute $\ell(\theta_t + \alpha_i \nabla \ell(\theta_t))$.
    for $i$ = 1, 2, ..., $i_{max}$ do
        if $\ell(\theta_t + \alpha_i \nabla \ell(\theta_t)) \leq \ell(\theta_t) - \rho\alpha \nabla \ell(\theta_t)^T \nabla \ell(\theta_t)$ then
            Set $\alpha_{i+1} = w\alpha_i$.
        else
            Set $\theta_{t+1} = \theta_t + \alpha_i \nabla \ell(\theta_t)$.
        end
    end
end
```

The reason for including such a model is that the quasi-Newton easily gets stuck in local extreme points, and this simple improvement immediately makes the algorithm more robust. It is also interesting to see whether this simple fix makes a lot of difference, or if more advanced methods are required.

4.2 Performance metrics

One of the most commonly modelled aspects of financial returns is the volatility component. Although the actual returns of financial data is very difficult to forecast, forecasting the squared return is comparatively easier. For this reason, one interesting use of the Hidden Markov Model is volatility forecasting. This prediction is easily performed using the posterior probabilities calculated in the first part of the forward algorithm, which can be derived from equation (15). This can serve as an important metric when validating the model fit and a mean squared error (MSE) can be calculated and compared to e.g. a naive model that assumes constant variance. Suppose Y_t is the tth observation and $\hat{Y_t}$ is the prediction of Y_t at time t. Then the MSE is defined as

$$\text{MSE} = \frac{1}{n}\sum_{t=1}^{n}(Y_t - \hat{Y_t})^2. \tag{44}$$

Since the volatility is the predicted component here, $Y_t = R_t^2$ is used with $\hat{Y_t}$ derived from the forward algorithm. The result can then be compared to the sample variance of the entire sample $\hat{Y_t} = \frac{1}{n-1}\sum_{t=1}^{n}(Y_t - \bar{Y_t})^2$.

Another important metric to compare between models is the predictive log-likelihood. This metric measures the likelihood of the data given the 1-step density forecasts of the models. For this reason, it is only really reliable for comparisons of models where the same data and same window length has been used. The theoretical predictive log-likelihood is defined as

$$\begin{aligned} \ell_p(\theta) &= \int \log p_\theta(x) p_{\theta_0}(x)\mathrm{d}x \\ &\approx \sum_i \log p_\theta(x_i), \end{aligned} \tag{45}$$

The predictive log-likelihood can be computed in a similar fashion to the conditional expectations and variances computed in (11) and (12), applied to the Gaussian conditional distributions. Given data until time t and corresponding parameter estimate θ_t and probabilities $P(X_t = 1)$ and $P(X_t = 2)$ and transition probabilities estimated by the modified forward algorithm, the density function for the next observation is

$$\begin{aligned} f_{Y_{t+1}}(x) =& P(X_t = 1)(\lambda_1^t f_{N_1^t}(x) + (1-\lambda_1^t) f_{N_2^t}(x)) \\ &+ P(X_t = 2)(\lambda_2^t f_{N_2^t}(x) + (1-\lambda_2^t) f_{N_1^t}(x)), \end{aligned} \tag{46}$$

where λ_1^t, λ_2^t are the estimated transition probabilities. The predictive log-likelihood can then be defined as

$$\text{PLL} = \sum_t \ln(f_{Y_{t+1}}(y_{t+1})). \tag{47}$$

and can be used as a metric for how well the model predicts data. This metric contrasts greatly to the MSE used in that the MSE measures how well the model predicts the squared returns, whereas the predictive log-likelihood measures how well the returns themselves are predicted.

5 Results

5.1 Synthetic Data

Method	Variance MSE	Predictive Log-Likelihood
Trust Region	$2.20 \cdot 10^3$	$-2.22 \cdot 10^4$
Quasi-Newton	$2.19 \cdot 10^3$	$-2.21 \cdot 10^4$
Quasi-Newton Line-Search	$2.19 \cdot 10^3$	$-2.23 \cdot 10^4$
Predictor-Corrector	$2.21 \cdot 10^3$	$-2.22 \cdot 10^4$
Line-Search	$2.20 \cdot 10^3$	$-2.22 \cdot 10^4$

Table 1: MSE and Predictive Log-Likelihood for the synthetic data, using a window length of 450.

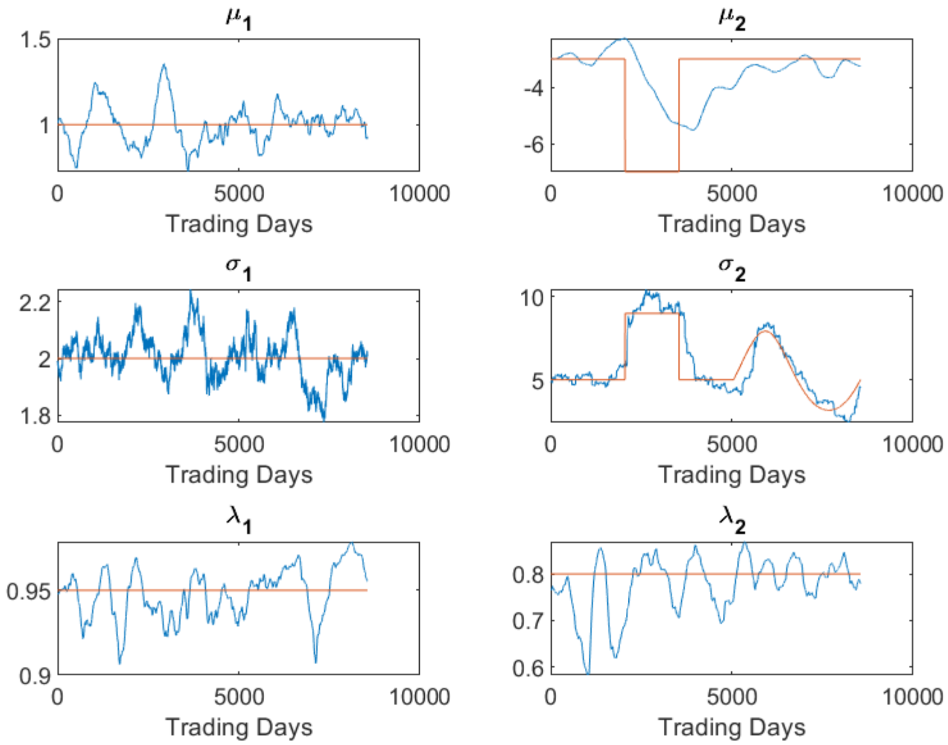

Figure 6: Parameter estimates for the Line Search model applied on synthetic data, using a window length of 450.

Parameter estimation was performed for all 4 models, using appropriate parameter transforms for the window length of 450. No real stability issues arose for any of the models, and so the parameter transforms did not need to be especially restrictive.

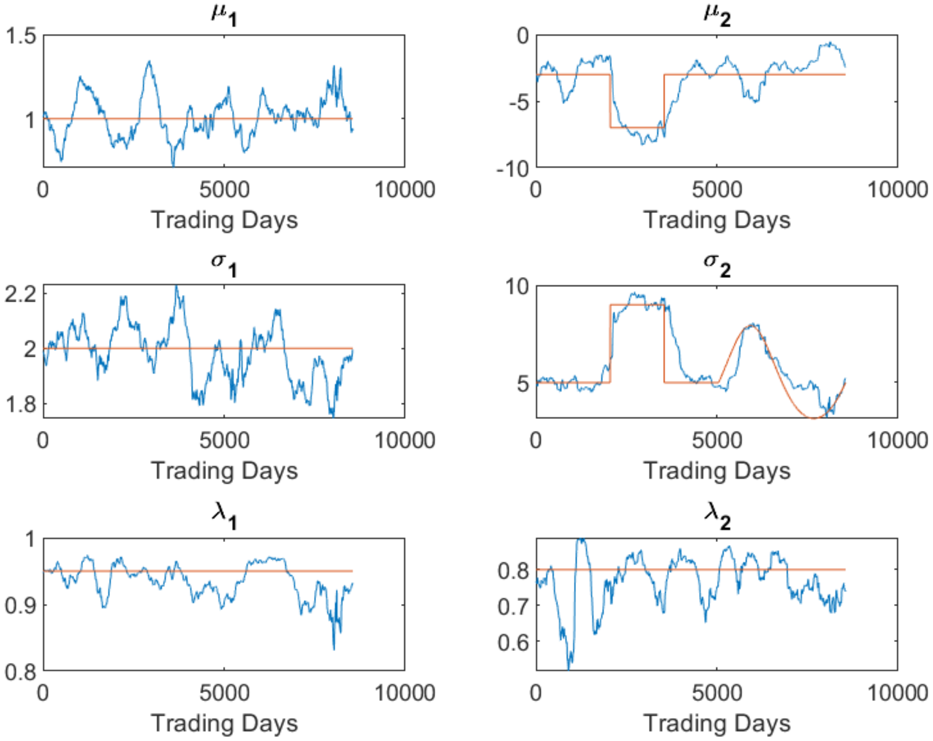

Figure 7: Parameter estimates for the regular quasi-Newton model applied on synthetic data, using a window length of 450.

One of the things that immediately stand out is that the predicted state transition probabilities vary heavily throughout the sample for all models, but especially for the Trust-Region and Quasi-Newton models. This speaks to the fact that although the underlying process might follow a homogeneous Markov chain, the Hidden Markov model can locally vary quite a lot.

Since the transition probabilities were $\lambda_1 = 0.95$ and $\lambda_2 = 0.8$, the corresponding sojourn times are $T_1 = 20$ and $T_2 = 5$ respectively. Consequently, on average, the Markov chain can be expected to switch states from state 1 every 20 steps and from state 2 every 5 steps. This means that the window length of the rolling window is rather large compared to the sojourn times, which gives the process plenty of switches in each window. Since this should in theory lead to a more accurate estimation of parameters, it is especially interesting to see the parameters vary so much throughout the sample.

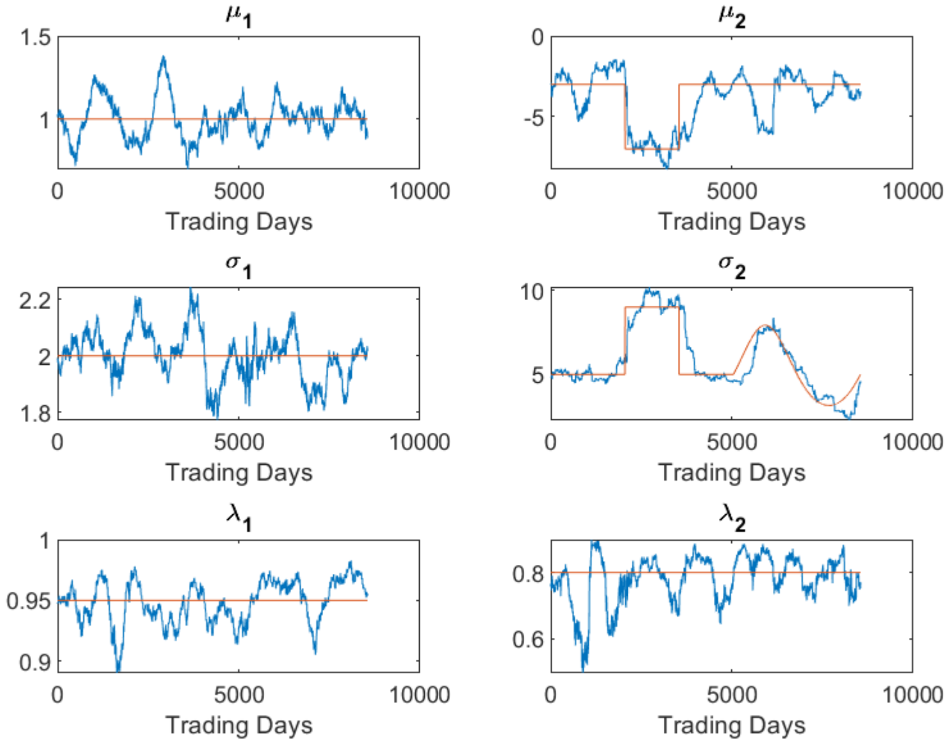

Figure 8: Parameter estimates for the Line Search quasi-Newton model applied on synthetic data, using a window length of 450.

5.1.1 Differences in parameter estimation between models

One of the primary features that stand out is the inability for the Line-Search and Predictor-Corrector models to react to a change in μ_2. This may be related to the fact that these are the only 2 score-driven models, which means that the additional complexity of the Hessian could be the deciding factor. The difference is much more extreme for the Predictor-Corrector model than for the Line-Search model though, see figure 6.

For the parameters $\sigma_1, \sigma_2, \lambda_1, \lambda_2$, there is clearly a larger degree of variation in the parameter estimates for the Quasi-Newton and Trust-Region models than for the score-driven models.

It should be noted however that whereas the line-search method adjusts much slower than the Predictor-Corrector method does for μ, the adjustment is virtually identical for σ_2.

It is not clear whether or not it is a problem from a practical standpoint that the

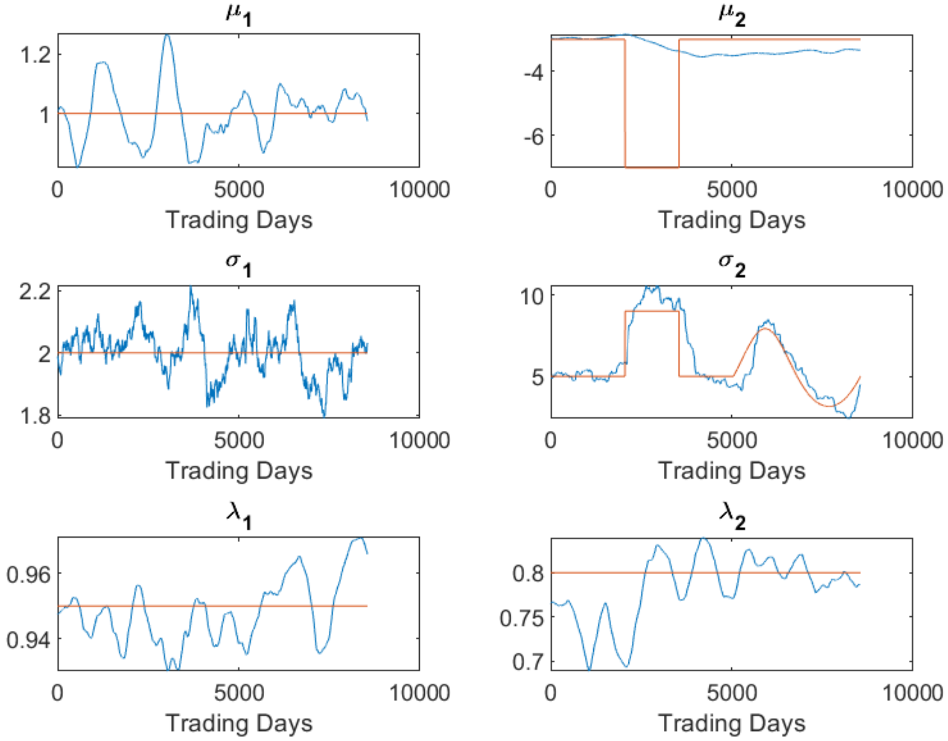

Figure 9: Parameter estimates for the Predictor-Corrector model applied on synthetic data, using a window length of 450.

Predictor-Corrector method does not react well to sudden changes in mean. In fact, this could be an advantage when it comes to real returns since extreme events could otherwise highly impact the mean as it does to the Hessian based models.

The non-reaction is also interesting considering that in terms of MSE and PLL the Predictor-Corrector and Line-search algorithms are not at a disadvantage compared to the other models. This may partly be because the inaccuracy in the parameter switch regions are compensated for by the relatively constant parameter estimates in the other regions and especially the Predictor-Corrector algorithm has a very low rate of change in general.

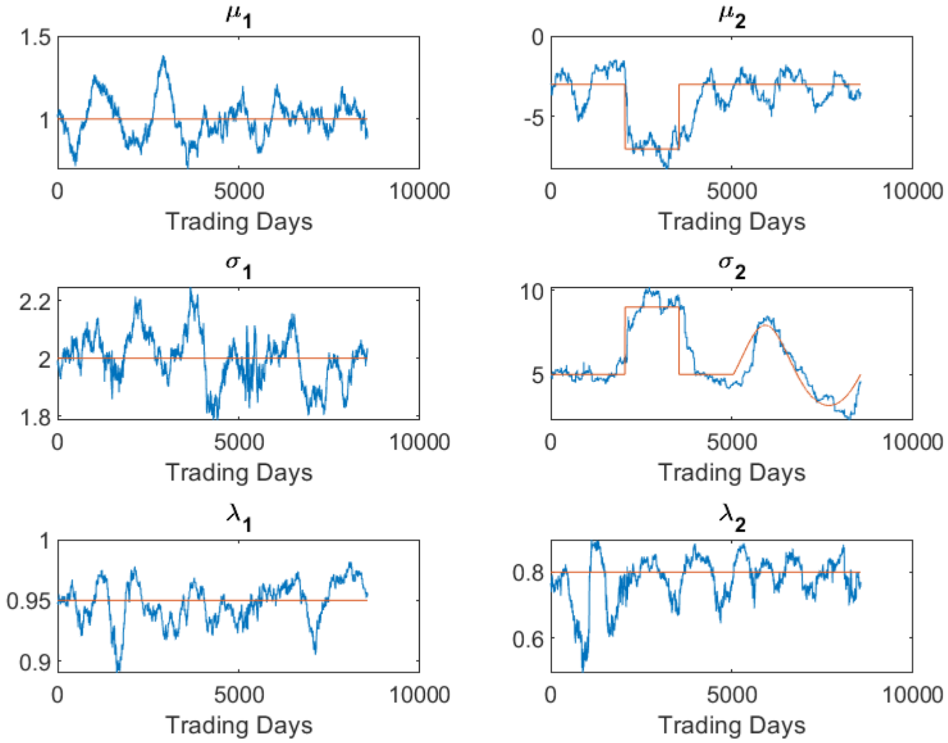

Figure 10: Parameter estimates for the Trust-Region model applied on synthetic data, using a window length of 450.

5.2 Real Returns

Method	Variance MSE	Predictive Likelihood
Trust Region	$4.38 \cdot 10^{-7}$	$3.36 \cdot 10^{4}$
Quasi-Newton	$4.42 \cdot 10^{-7}$	$3.42 \cdot 10^{4}$
Quasi-Newton Line-Search	$4.36 \cdot 10^{-7}$	$3.37 \cdot 10^{4}$
Predictor-Corrector	$4.42 \cdot 10^{-7}$	$3.41 \cdot 10^{4}$
Line-Search	$4.44 \cdot 10^{-7}$	$3.38 \cdot 10^{4}$

Table 2: Results for 450 trading days window length, S&P 500 index.

The results for Real financial returns were split into 2 different sections with different methodology as to both compare the performance between different model types, but also to study how the Trust-Region model performs more in depth.

For the first section, all models were fitted for both window length 450 and 1700 for the S&P 500 index and for length 450 for the Nikkei index. For all 3 cases, a line-search quasi-Newton model was included for comparison due to the instability of regular quasi-

Method	Variance MSE	Predictive Likelihood
Trust Region	$5.00 \cdot 10^{-7}$	$2.94 \cdot 10^{4}$
Quasi-Newton*	$4.93 \cdot 10^{-7}$	$2.91 \cdot 10^{4}$
Quasi-Newton Line-Search	$4.98 \cdot 10^{-7}$	$2.91 \cdot 10^{4}$
Predictor-Corrector	$4.98 \cdot 10^{-7}$	$2.94 \cdot 10^{4}$
Line-Search	$4.97 \cdot 10^{-7}$	$2.93 \cdot 10^{4}$

Table 3: Results for 1700 trading days window length, S&P 500 index. Note the high reduction in MSE for the Quasi-Newton method compared to the other methods.

Method	Variance MSE	Predictive Likelihood
Trust Region	$2.80 \cdot 10^{-7}$	$3.95 \cdot 10^{4}$
Quasi-Newton	$2.90 \cdot 10^{-7}$	$3.92 \cdot 10^{4}$
Line-Search Newton	$2.87 \cdot 10^{-7}$	$3.99 \cdot 10^{4}$
Predictor-Corrector	$2.83 \cdot 10^{-7}$	$4.00 \cdot 10^{4}$
Line-Search	$2.84 \cdot 10^{-7}$	$3.96 \cdot 10^{4}$

Table 4: Results for 450 trading days window length, Nikkei index.

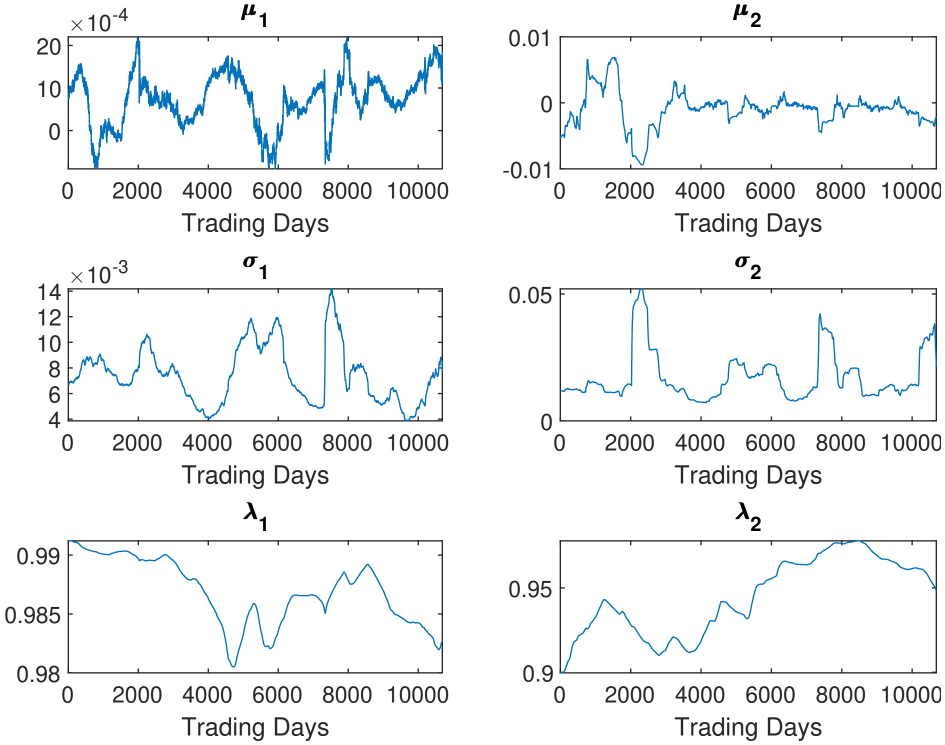

Figure 11: Parameter estimates for the Line Search model applied on the S&P-500, using a window length of 450. Note the spike in σ_2 volatility around the 2000th trading day (Black Monday).

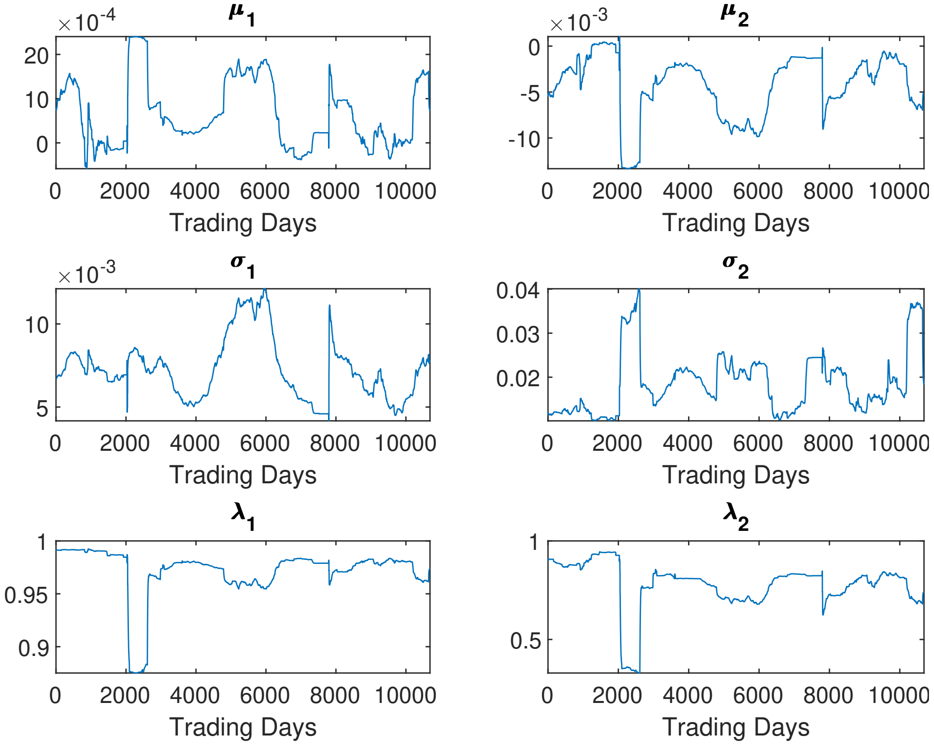

Figure 12: Parameter estimates for the regular quasi-Newton model applied on the S&P-500, using a window length of 450. Note the spike in σ_2 volatility around the 2000th trading day (Black Monday), accompanied by sharp dips in λ_1, λ_2.

Newton. The performance metrics were compiled into tables 2, 3 and 4. All parameter estimates for the 450 window length S&P-500 can be found in figures 11-15, as well as parameter estimates for the regular quasi-Newton model and the Trust-Region model for the 1700 window length S&P-500 index. The parameter estimates for the Nikkei index and the remaining models for the 1700 window length S&P-500 index can be found in the appendix.

For the second section, the Trust Region model was fitted to the S&P 500 index using a window length of 450, but with constant maximum step length and different norms as to study the behavior of the model when the variance is allowed to vary more than the transitional probabilities and vice versa. For this section, certainty measurements in the form of bootstrapped confidence intervals were also simulated and presented in table 5. Parameter estimates for the different norms can be found in figures 18-20.

5.2.1 Overall Performance

All in all, the tests were inconclusive in determining which model performs the best purely based on the performance metrics. There does not seem to be much correlation between the MSE and predictive log-likelihood measurements of the different models,

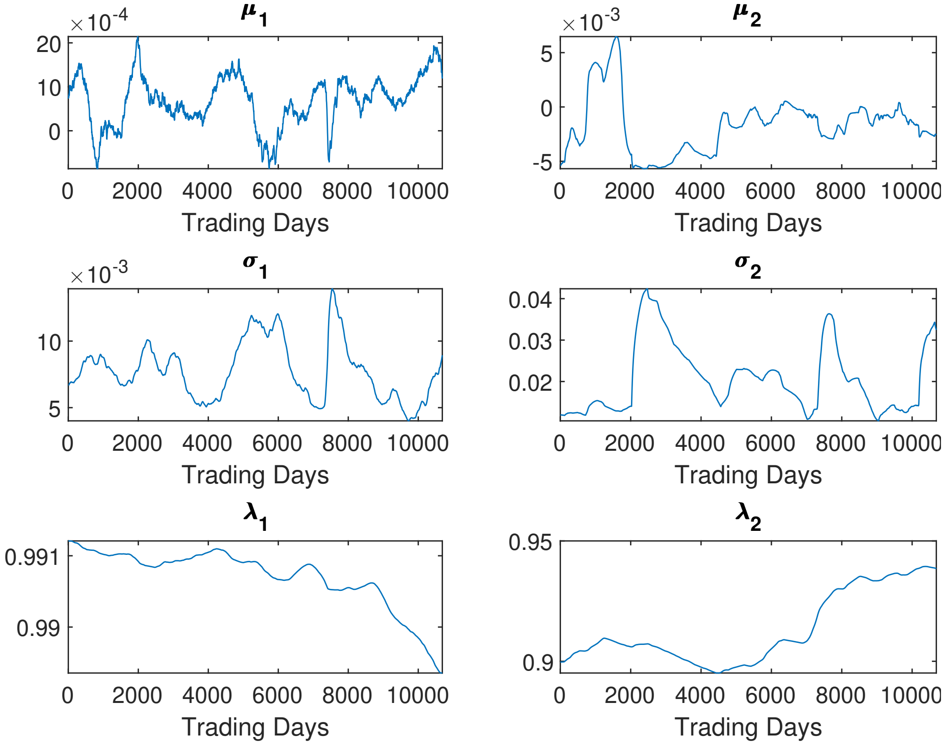

Figure 13: Parameter estimates for the Predictor-Corrector model applied on the S&P-500, using a window length of 450. Note the spike in σ_2 volatility around the 2000th trading day (Black Monday).

and the performance varies wildly between different data sets and window lengths.

A clear illustration of the irregular results of model performance is for the S&P-500 index with window length 450, where both the highest predictive log-likelihood and the lowest MSE were obtained by the regular and line-search quasi-Newton models respectively (see table 2). This is contrasted with the Nikkei index performance, where the Trust-Region and Predictor-Corrector models obtained the lowest MSE and highest predictive log-likelihood respectively. For the 1700 window length S&P-500 index, the greatest predictive log-likelihood was again obtained by the Trust-Region and Predictor-Corrector models whereas the lowest MSE was obtained by the regular quasi-Newton method. It should however be noted that the quasi-Newton model displayed irregular behavior (see figure 16) and as such, the result should be viewed with caution.

Nystrup et al. (2017) were able to read the window length from the parameter estimates due to the extreme nature in which the outliers (particularly the Black Monday date) impacted estimates. The same behavior can be seen in the estimates presented here, but with varying degrees between the different models. The quasi-Newton methods and Trust-Region methods were clearly able to reproduce this result, whereas the smoothing effects of the Predictor-Corrector scheme makes for a less abrupt transition (see e.g.

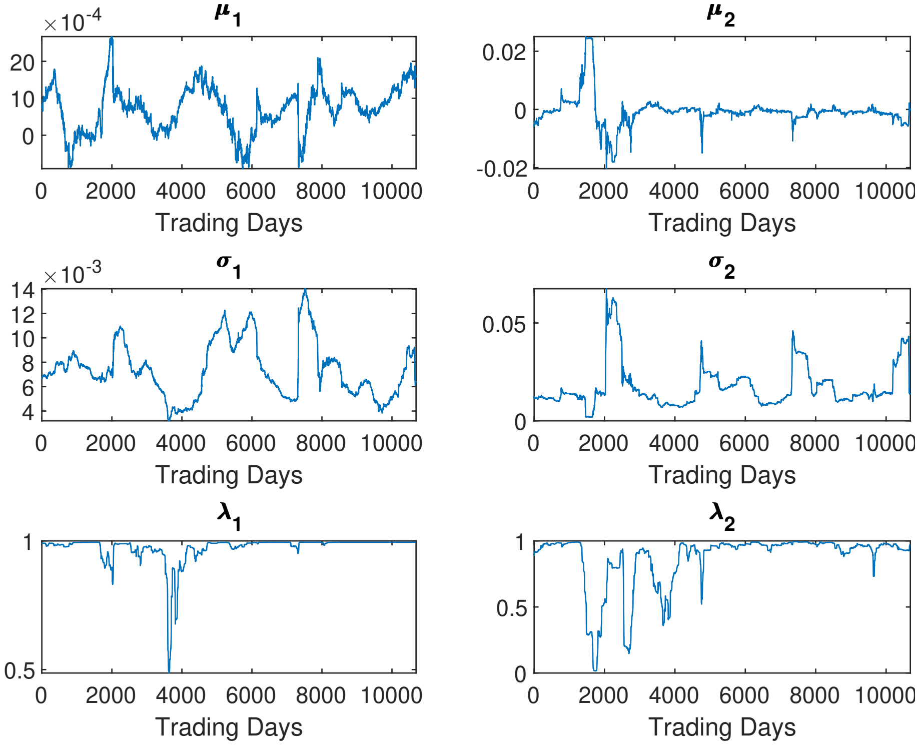

Figure 14: Parameter estimates for the Line Search model applied on the S&P-500, using a window length of 450. Note the spike in σ_2 volatility around the 2000th trading day (Black Monday), accompanied by a sharp dip in λ_2.

Figure 12, where the λ_1, λ_2 parameter estimates dip for exactly the window length 450 before returning close to the original estimates).

Further comparing the results of the 450 window length to the 1700 window length, there are great differences between how the transition probabilities λ_1, λ_2 vary throughout the sample. While the results of Nystrup et al. (2017) had the transition probability of the high-volatility state drop to roughly 0.6 around the Black Monday date (observed around the 2000th trading day) using a window length of 1700, the corresponding drop for the window length 450 was 0.4 for the quasi-Newton method and close to 0 for the Trust-Region model using a regular L_2 norm. As for the 1700 window length, none of the models covered achieved dips in the λ parameters close to the ones obtained by Nystrup et al. (2017) with the most prevalent dip observed by the Trust-Region model (see figure 19). This discrepancy can however be attributed to differences in parameter transformations and step sizes used by Nystrup et al..

In practice, these large dips in transition probabilities may be detrimental to model performance in e.g. state inference and when applied to trading strategies, since it increases the number of regime switches and consequently increasing transactional costs (Nystrup et al., 2020b). Since the Predictor-Corrector model succeeded in smoothing

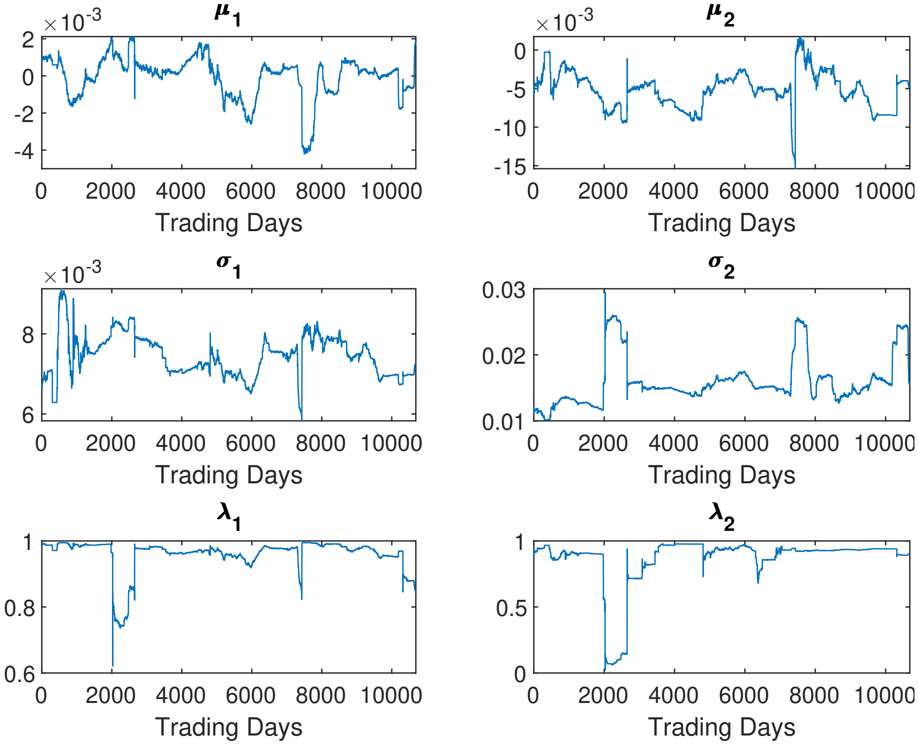

Figure 15: Parameter estimates for the Line Search quasi-Newton model applied on the S&P-500, using a window length of 450. Note the spike in σ_2 volatility around the 2000th trading day (Black Monday), accompanied by sharp dips in λ_1, λ_2.

out the impulses for λ_1, λ_2, this could constitute a practical improvement over the quasi-Newton method in state persistence. *It should be noted that attempts to reproduce the results of the Predictor-Corrector using a smaller step size for the quasi-Newton methods failed, resulting in substantially worse performance.*

5.2.2 Stability issues and transformations

There were great stability issues relating to the quasi-Newton method particularly for shorter window lengths, when fitting these models. In order to ensure stability, the parameters had to be transformed and bounded into specific intervals and slight changes to the nature of the transforms had a profound impact on the performance of the models.

These issues make it very difficult to outright compare the different models, since the effect of the transforms have to be taken into account. This makes it difficult to motivate that e.g. the Predictor-Corrector method might have an edge over the quasi-Newton method since it could help prevent overfitting based on outliers, when a similar effect could be produced by simply changing the parameter transforms for the quasi-Newton model. *Although the behavior of the Predictor-Corrector model could not be*

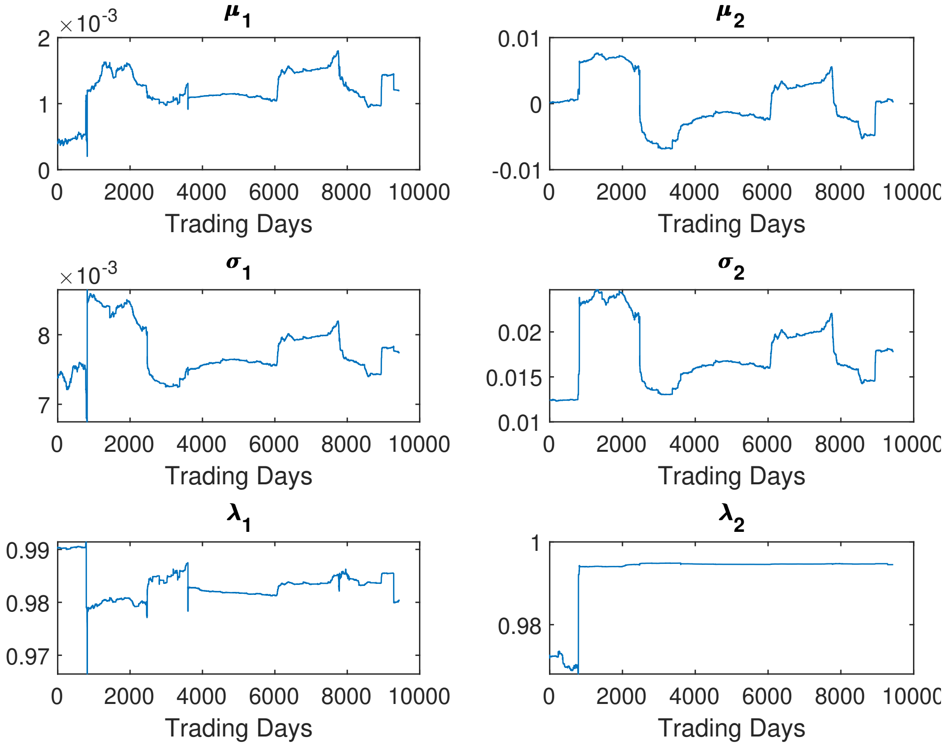

Figure 16: Parameter estimates for the regular quasi-Newton model applied on the S&P-500, using a window length of 1700. Note the spike in σ_2 volatility around the 700 trading day (Black Monday).

reproduced with the quasi-Newton model, it is not impossible that a similar result could have been obtained by using very specific parameter transformations. Hence, although the Predictor-Corrector model could constitute an improvement over the quasi-Newton method in the sense that controlling the persistence of the states can improve model performance, it may simply be because the quasi-Newton method with corresponding parameter transformations is not specified appropriately.

Nevertheless, some qualitative observation can be made in regards to stability. *Whereas the quasi-Newton required very specific parameter transforms tailored to specific data sets to ensure stability, the Trust-Region model presented was able to run without any transforms and only occasionally ran into boundary issues.* The same was true for the line-search algorithms, for which the backtracking line-search method is a very practical tool for avoiding boundary conditions.

All in all it is clear that in terms of ease of use and restrictiveness, *the quasi-Newton model performs poorly when compared to both the score-driven and trust-region alternatives, but in overall performance it is very difficult to definitively say which class of models performs the best.*

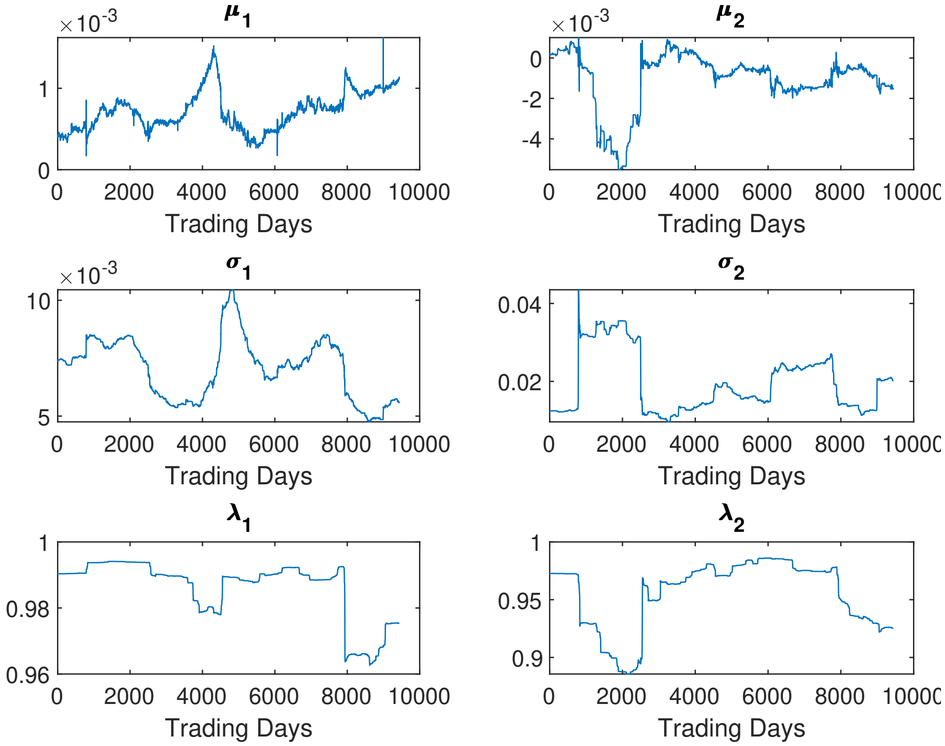

Figure 17: Parameter estimates for the Trust-Region model applied on the S&P-500, using a window length of 1700. Note the spike in σ_2 volatility around the 700 trading day (Black Monday).

5.2.3 Thoughts on sojourn time distribution

Nystrup et al. (2017) came to the conclusion that some of the performance issues of Hidden Markov models could be attributed to the fact that sojourn times are generally implicitly assumed to be geometrically distributed. This is partly motivated by the fact that when the transition probabilities are allowed to vary with time, they often vary quite a lot between different parts of the sampled data.

Based on this general observation, it is interesting to study the behavior of the parameter estimates of the model when applied to synthetic data, since the results of this paper shows that even when the underlying process is a homogeneous Markov chain, the transition probability estimates based on a moving window can vary greatly from the true values over time but it seems to do so in a recurring pattern with no apparent disruptions when other parameters change either abruptly or gradually.

This is not exactly the case for the financial data studied, where e.g. the Black Monday event of 1987 causes a substantial dip in the transition probabilities of the S&P 500 index as previously mentioned, especially for the window length 450. This is most extreme for the Trust-Region model, where the transition probability of the high-volatility state drops close to 0.

It is clear that this behavior is not consistent with the results obtained for synthetic data. However, at least part of the variation can be attributed to the outliers contained in the data, since a single extreme data point could, if permitted, be put into its own high volatility state. Consequently, this would cause the corresponding transition probability to drop considerably since the observed state sequence becomes a single point in the high volatility state followed by and preceded by low volatility states. The clearest example of this is the Black-Monday date previously discussed.

5.3 Trust Region Model Norm comparisons for Real Returns

Diagonal elements of A	Variance MSE	Predictive Likelihood
(1, 1, 5, 5, 0.5, 0.5)	4.38 (1.35, 9.92) $\cdot 10^{-7}$	3.36 (3.33, 3.38) $\cdot 10^{4}$
(1, 1, 1, 1, 1, 1)	4.38 (1.35, 9.82) $\cdot 10^{-7}$	3.36 (3.33, 3.39) $\cdot 10^{4}$
(1, 1, 0.1, 0.1, 10, 10)	4.36 (1.33, 9.87) $\cdot 10^{-7}$	3.37 (3.35, 3.40) $\cdot 10^{4}$
(1, 1, 0.05, 0.05, 50, 50)	4.43 (1.43, 10.02) $\cdot 10^{-7}$	3.37 (3.35, 3.40) $\cdot 10^{4}$
(1, 1, 0.01, 0.01, 100, 100)	4.46 (1.43, 10.01) $\cdot 10^{-7}$	3.36 (3.34, 3.39) $\cdot 10^{4}$

Table 5: Results for 450 trading days window length, S&P 500 index, using different norms for the Trust Region Model. Parenthesis contain bootstrapped 95% confidence intervals for the metrics.

Table 5 shows the mean squared error of the variance predictions and the predictive log-likelihood with corresponding bootstrapped two-sided 95% confidence intervals. Although these intervals are broad and all metrics for all models are contained in all confidence intervals, these intervals are likely inflated due to the effect of outliers (that is, a large portion of predictive log-likelihood and particularly the MSE is contributed from a few extreme observations).

As was intended, the transition probabilities λ_1, λ_2 were progressively less volatile with an increase in corresponding $A_{5,5}, A_{6,6}$ weights. Meanwhile, the parameters σ_1, σ_2 did not experience an increased variability despite the corresponding $A_{3,3}, A_{4,4}$ weights being progressively decreased. This may be due to the volatility parameters already reaching their global optimum, and that further availability is thus not reflected in the parameter estimation graphs.

Although the difference is small and well within these confidence intervals, the norms seem to impact the predictive log-likelihood and MSE in a systematic pattern with the higher MSE values acquired around the last 2 norms, and the lowest MSE at $4.36 \cdot 10^{-7}$ acquired using diagonal elements $(1, 1, 0.1, 0.1, 10, 10)$, in line with the Quasi-Newton Line-Search model used in the previous section. This suggests that there may be advantages to introducing a limitation to the way in which the transition probabilities are allowed to vary. Therefore, the Trust-Region model with a norm other than L_2 may constitute an improvement in a similar fashion to the Predictor-Corrector model previously discussed, since the variation of the transition probabilities can be minimized

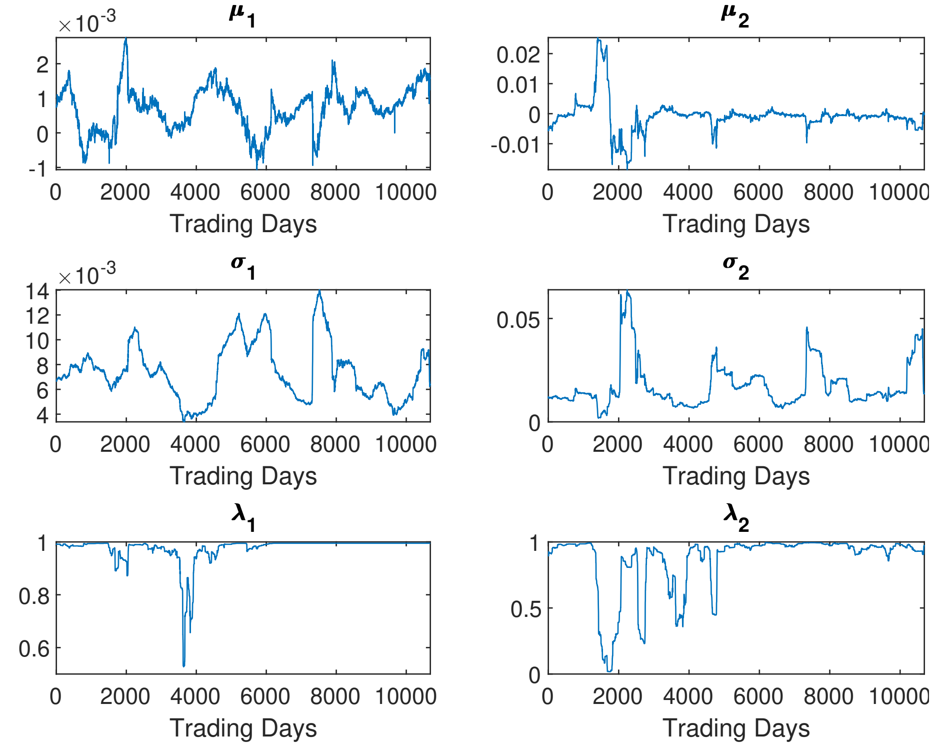

Figure 18: Parameter estimates for the Trust-Region model applied on the S&P-500, using a window length of 450. Note the spike in σ_2 volatility around the 2000 trading day (Black Monday). The diagonal elements of the norm was $A = (1, 1, 5, 5, 0.5, 0.5)$.

without penalty to the performance metrics and without the accompanied smoothing of the σ_1, σ_2 parameter estimates.

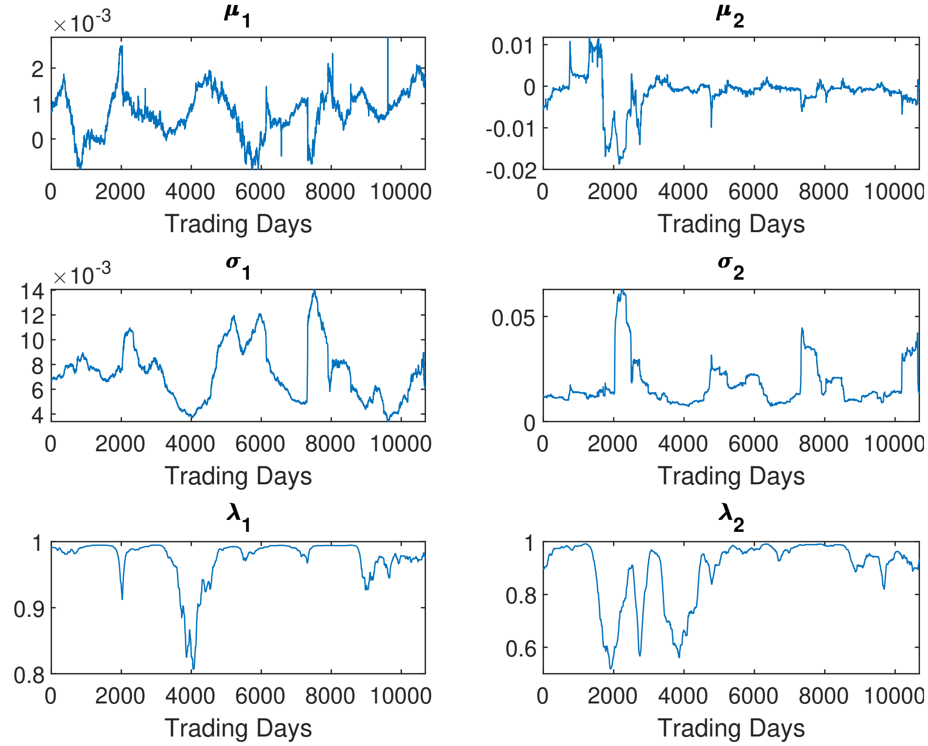

Figure 19: Parameter estimates for the Trust-Region model applied on the S&P-500, using a window length of 450. Note the spike in σ_2 volatility around the 2000 trading day (Black Monday). The diagonal elements of the norm was $A = (1, 1, 0.1, 0.1, 10, 10)$.

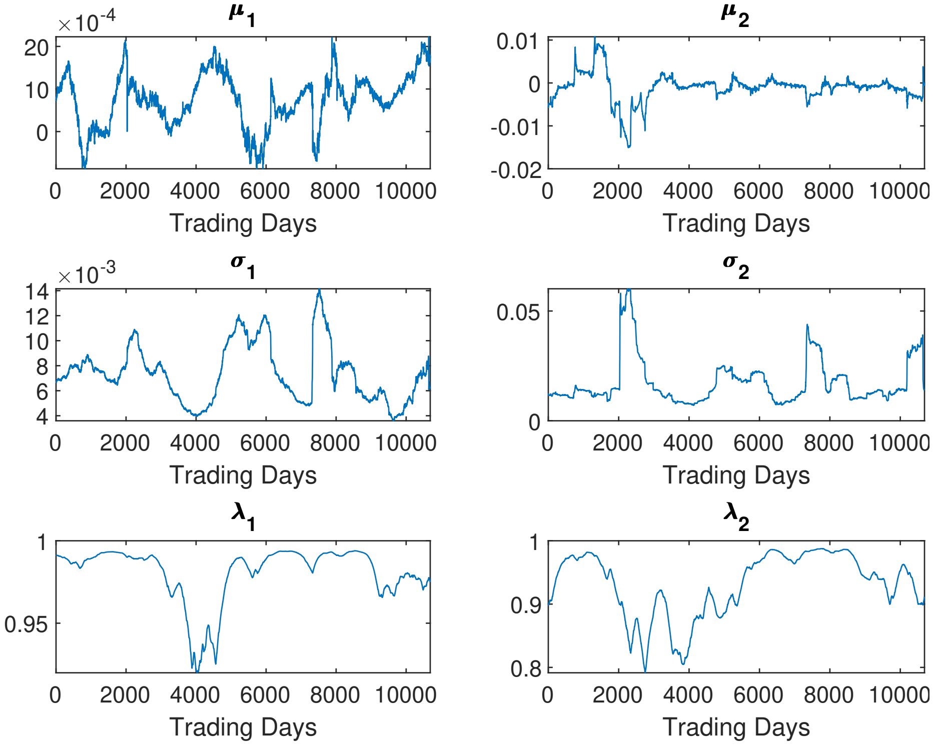

Figure 20: Parameter estimates for the Trust-Region model applied on the S&P-500, using a window length of 450. Note the spike in σ_2 volatility around the 2000 trading day (Black Monday). The diagonal elements of the norm was $A = (1, 1, 0.5, 0.5, 50, 50)$.

6 Conclusion and Future Work

The results of this thesis illustrates that there are areas where the quasi-Newton can be improved. Although it is difficult to compare model performance and thus definitively find such an improvement due to the impact parameter transformations have on model performance.

From a stability standpoint, the Trust-Region model was identified as a more robust alternative to the quasi-Newton method, although the performance metrics were not significantly different from the simpler model. Additionally, the Predictor-Corrector model showed a much smoother parameter transition for both synthetic and real data and although the performance metrics did not indicate an improvement, the persistence of the parameter estimates, especially the transition probabilities, could in and of itself constitute an improvement over the quasi-Newton model in practical applications.

Since the parameter transforms have such a profound effect on the performance of each model, a better way of improving performance of e.g. the quasi-Newton model could be to define parametric families of transformations and attempt to find transforms that are in some sense optimal for a given model and data set. This could further aid in developing and evaluating alternatives in a more definitive fashion.

7 References

References

Abrahamsen N, Nakovski D. 2021. Dynamic Asset Allocation based on Hidden Markov Model regime sequences. *Copenhagen Business School* Master's Thesis.

Bulla J. 2011. Hidden Markov models with t components. Increased persistence and other aspects. *Quantitative Finance.*

Bulla J, Mergner S, Bulla I, Sesboüé A, Chesneau C. 2011. Markov-switching asset allocation: do profitable strategies exist? *Journal of Asset Management* **12**: 310-321.

Byrd R, Khalfan H, Schnabel R. 1996. Analysis of a symmetric rank-one trust region method. *SIAM Journal on Optimization* Vol 6, **4**: 1025-1039.

Brown A, Bartholomew-Biggs M. 1989. Some Effective Methods for Unconstrained Optimization Based on the Solution of Systems of Ordinary Differential Equations. *Journal of Optimization Theory and Application* Vol. 62, **2**.

Bottou L, Curtis F, Nocedal J. 2018. Optimization Methods for Large-Scale Machine Learning. *Society for Industrial and Applied Mathematics.* **60**(2).

Cont, R. 2001. Empirical properties of asset returns: stylized facts and statistical issues. *Quantitative Finance* **1**(2): 223-236.

Creal D, Koopman SJ, Lucas A. 2013. Gerenalized autoregressive score models with applications *Journal of Applied Econometrics* **28**: 777-795

Creal D, Koopman SJ, Lucas A. 2008. A General Framework for Observation Driven Time-Varying Parameter Models. *Tinbergen Institute Discussion Paper* 08-108/4..

Griffiths, D. F., Higham, D. J. 2010. Numerical methods for ordinary differential equations: Initial value problems. *Springer*, London. ISBN 9780857291486.

Hamilton J, Lin G. 1996. Stock Market Volatility and the Business Cycle. *Journal of Applied Econometrics* **11**(5): 573-593.

Holst U, Lindgren G. 1991. Recursive Estimation in Mixture Models with Markov Regime. *IEEE Transactions on Information Theory* **37**(6): 1683-1690.

Hamilton J. 1989. A New Approach to the Economic Analysis of Nonstationary Time Series and the Business Cycle. *Econometrica* **57**(2): 357-384.

Levenberg K. 1944. A method for the solution of certain nonlinear problems in least squares. *Quart. Appl. Math.* **2**: 164-168.

Lystig TC, Huges JP. 2002. Exact computation of the observed information matrix for hidden markov models. *Journal of Computational and Graphical Statistics* **11**:

678-689.

Marquardt D.W. 1963. An algorithm for least squares estimation of nonlinear parameters. *SIAM Journal of Applied Mathematics* **11**: 431-441.

Nystrup P, Madsen E, Lindström E. 2017. Long memory of financial time series and hidden markov models with time-varying parameters. *Journal of Forecasting* **36**: 989-1002.

Nystrup P. 2017. Dynamic Asset Allocation. *Technical University of Denmark* PhD thesis.

Nystrup P, Kolm P. N., Lindström E. 2020b. Greedy Online Classification of Persistent Market States Using Realized Intraday Volatility Features. *Journal of Financial Data Science* **2**(3): 25-39.

Nystrup P, Lindström E, Madsen H. 2020a. Learning hidden Markov models with persistent states by penalizing jumps. *Expert Systems with Applications* **150**: 113307.

Rydén, T. 1997. On recursive estimation for hidden Markov models. *Stochastic Processes and their Applications.* **66**(1) 79-96.

Rydén T, Teräsvirta T, Åsbrink S. 1998. Stylized facts of daily return series and the hidden Markov model. *Journal of Applied Econometrics* **13**: 217–244.

Rydén T. 2008. EM versus Markov chain Monte Carlo for Estimation of Hidden Markov Models: A Computational Perspective. *Bayesian Analysis* **3**(4): 659-688.

Steihaug T. 1987. The Conjugate Gradient Method and Trust Regions in Large Scale Optimization. *SIAM Journal on Numerical Analysis* **20**(3).

Sun W, Yuan Y. 2006. Optimization Theory and Methods. *Springer-Verlag New York Inc.*. ISBN 9780387249759.

Turner R. 2008. Direct maximization of the likelihood of a hidden Markov model. *Computational Statistics and Data Analysis* **52**: 4147–4160.

8 Appendix

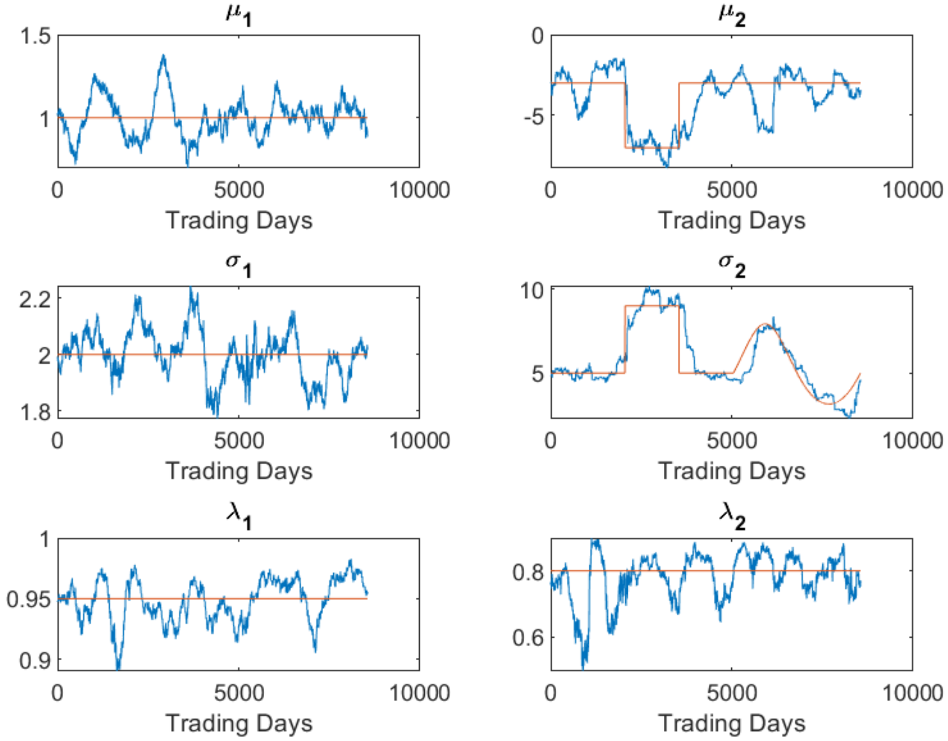

Figure 21: Line Search Newton, Synthetic data

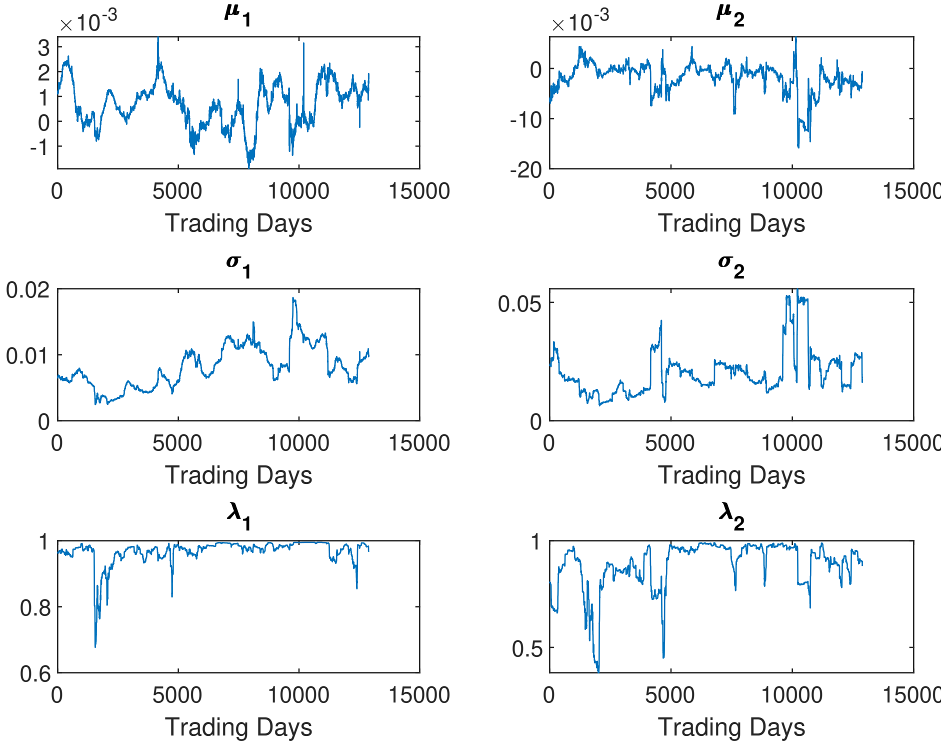

Figure 22: Trust Region Method, Window length 450 from the Nikkei index.

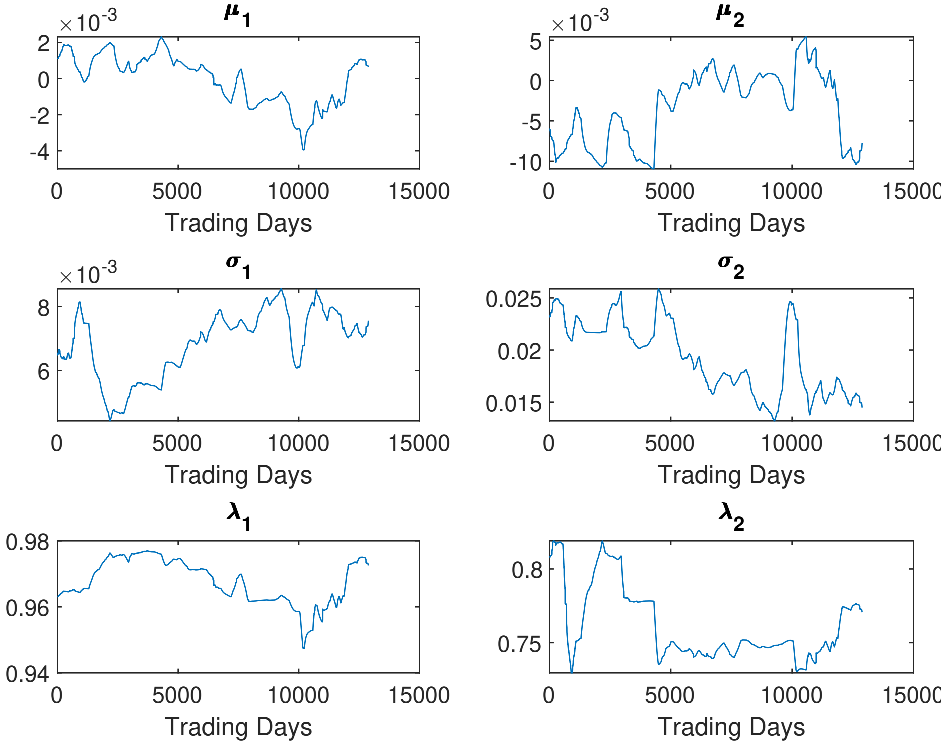

Figure 23: Newton Method, Window length 450 from the Nikkei index.

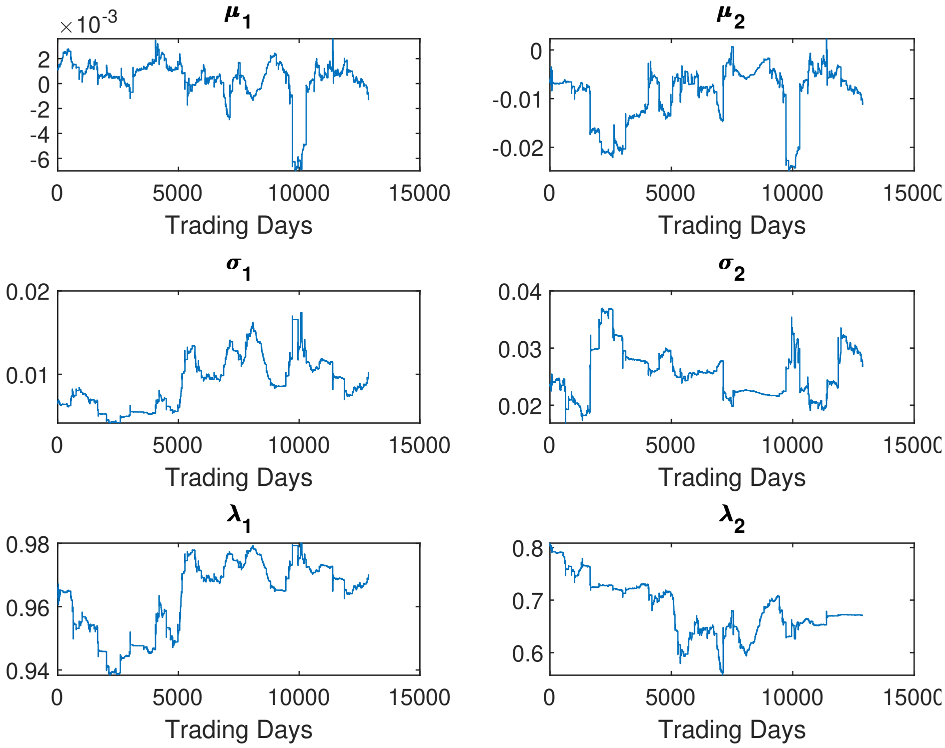

Figure 24: Line-Search Newton Method, Window length 450 from the Nikkei index.

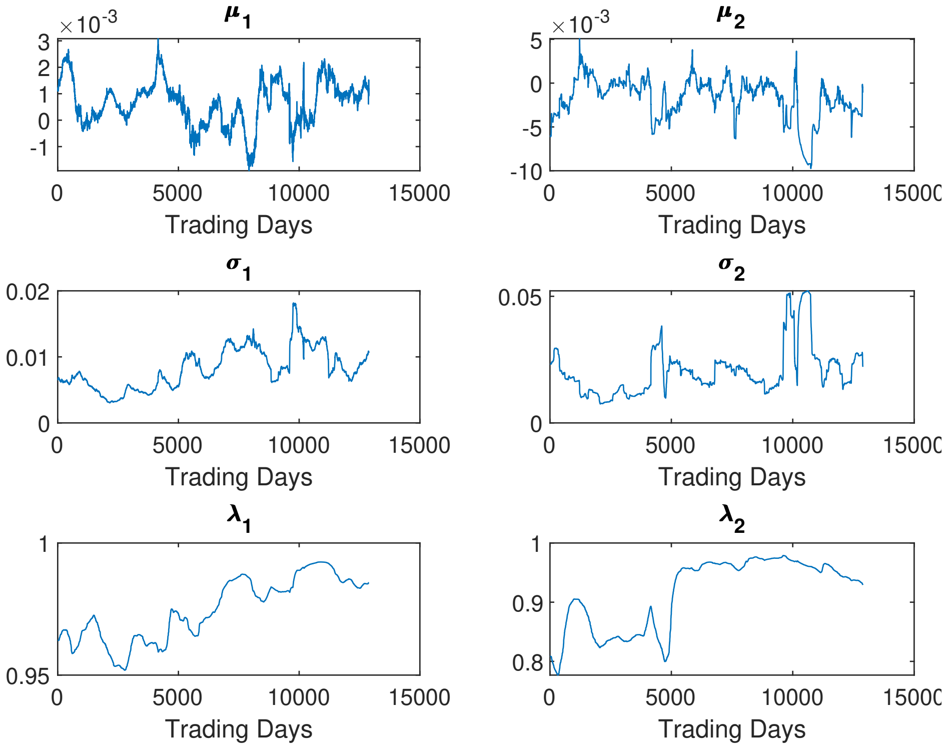

Figure 25: Line-Search Method, Window length 450 from the Nikkei index.

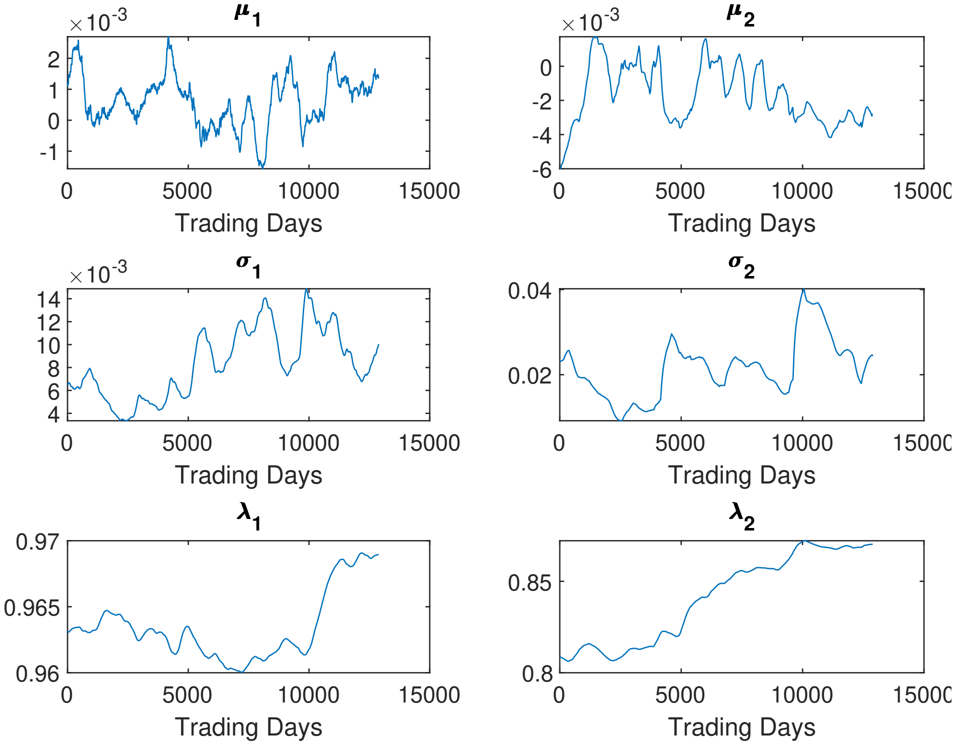

Figure 26: Predictor-Corrector Method, Window length 450 from the Nikkei index.

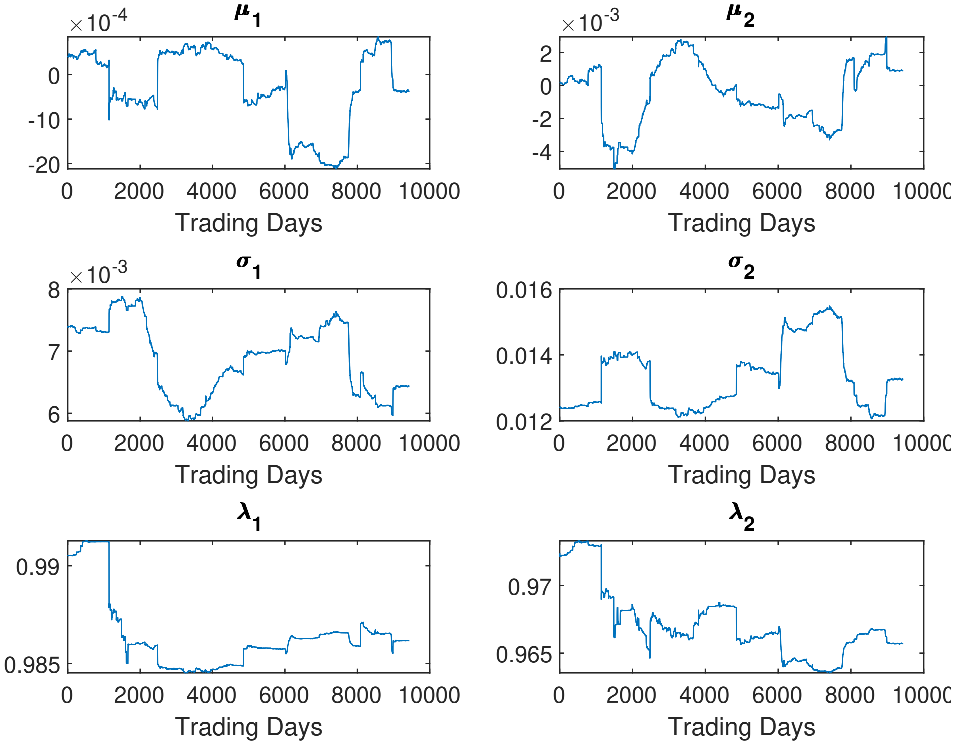

Figure 27: Line-Search Newton Method, Window length 1700 from the S&P 500 index.

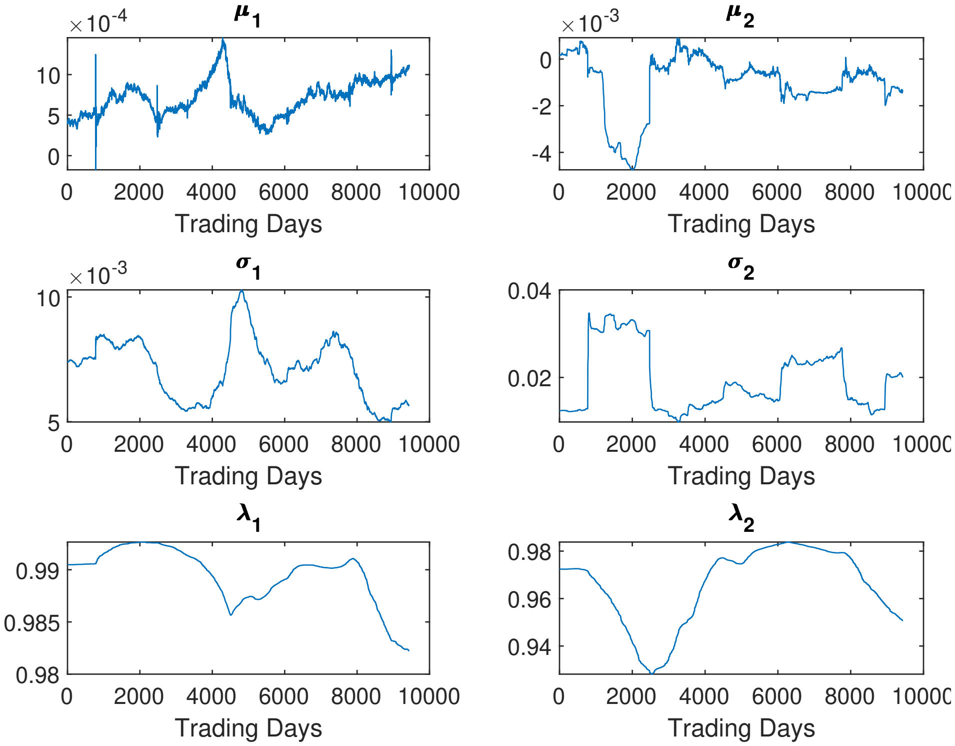

Figure 28: Line-Search Method, Window length 1700 from the S&P 500 index.