Lund University

Master's Thesis

Outlier-Robust Dynamic Portfolio Optimization based on Bear-Bull-Regimes

Author:
Emil Eliasson
Linus Hamlin

Supervisor:
Prof. Erik Lindström

May 30, 2018

LUND UNIVERSITY

Abstract

Master's Thesis

Outlier-Robust Dynamic Portfolio Optimization based on Bear-Bull-Regimes

by Emil Eliasson and Linus Hamlin

The work in this thesis is meant to improve an existing algorithm described in Nystrup (2017). As the original model uses a normal distribution to approximate the daily logarithmic returns, the authors of this thesis aim to improve the approximation by using Student's t-distribution which may be a better approximation of financial data. The algorithm uses a two state hidden Markov model to estimate the current state (also known as regime) of the market, bull or bear. Based on the estimation, predictions of future returns are made. The algorithm will then use this information to trade a risky asset, in this case the S&P 500 stock index. A portion of the available capital will be placed in the asset and the rest will be held in cash at the risk free rate. The portion of the available capital the algorithm is to invest in the risky asset is decided using model predictive control. Using model predictive control one is able to maximize the return for the entire portfolio over a future time horizon. In the maximization one is as well able to include transaction costs and a general aversion against both risk and trading.

The algorithm is able to obtain a greater return at a lower risk than just investing in a static portfolio of the stock index. This will yield a greater Sharpe ratio than the stock index. The resulting algorithm works the best if the portfolio is long-only, i.e. if the algorithm is not allowed to go short in the traded asset. Even though the long-short portfolio is able to yield a higher return to the investor, it will contain more risk and therefore have a lower Sharpe ratio. The final recommendation will therefore be for the investor to use the algorithm with a long-only portfolio restriction.

The algorithm allows the use of Bayesian optimization for obtaining optimal model hyperparameters, for instance the risk- or trading aversion, to maximize performance on the in-sample data set. The usage will however lead to a huge risk of over-fitting the model to the in-sample data set and the desired properties are most probably lost going out-of-sample. The hyperparameters should therefore be chosen manually with great care and thorough testing. Changing one hyperparameter may also lead to undesired effects as the many of the hyperparameters are mutually dependent making in-sample training more difficult.

Acknowledgements

We would like to thank our supervisor Erik Lindström at the Faculty of Engineering, Lund University, for his expertise and valuable opinions regarding the subject at hand as well as for his unceasing support and commitment throughout the entire course of the project.

Contents

List of Abbreviations

BHHH	**B**erndt-**H**all-**H**all-**H**ausman algorithm
BO	**B**ayesian **O**ptimization
CAPM	**C**apital **A**sset **P**ricing **M**odel
CDF	**C**umulative **D**istribution **F**unction
CML	**C**apital **M**arket **L**ine
EI	**E**xpected **I**mprovement
EM	**E**xpectation-**M**aximization algorithm
ETF	**E**xchange **T**raded **F**und
GAS	**G**eneralized **A**utoregressive **S**core model
GP	**G**aussian **P**rocess
HMM	**H**idden **M**arkov **M**odel
IID	**I**ndependently and **I**dentically **D**istributed
KPI	**K**ey **P**erformance **I**ndicator
MDD	**M**aximum **D**raw**D**own
ML	**M**aximum **L**ikelihood
LH	**L**ystig-**H**ughes algorithm
LL	**L**og-**L**ikelihood
MGF	**M**oment **G**enerating **F**unction
MPC	**M**odel **P**redictive **C**ontrol
PDF	**P**robability **D**ensity **F**unction
PI	**P**robability of **I**mprovement
QP	**Q**uadratic **P**rogramming
RW	**R**olling **W**indow
SV	**S**tochastic **V**ariable

Chapter 1

Introduction

1.1 Aim and scope of thesis

Markowitz (1952) laid the foundation of modern portfolio theory with an article that eventually earned him the Nobel prize. In short, the article explains the importance of maximizing the expected return in relation to the volatility to achieve an efficient portfolio. An efficient portfolio uses diversification to reduce specific risk, which works very well if the individual assets are not strongly positively correlated. The set of efficient portfolios form what is named the efficient frontier.

The reasoning was later developed by Sharpe (1964), who was awarded with a Nobel prize as well, into CAPM. In this model it is possible for the investor to invest capital at a risk-free return, r_{f}. The investor will then choose how much of their capital to invest at the risk-free rate and how much to invest at a tangent of the efficient frontier, for example the S&P 500 market portfolio. If the investor choses to invest at a higher systematic risk (i.e. the risk imposed by the volatility of the entire market) a larger portion of the capital will be invested in the index and the expected return will be higher. Every investment along the capital market line (CML) will have equal Sharpe-ratio.

Even though these discoveries were breaking new ground at the time and found a way to reduce risk while maintaining expected return, the models fail to reduce what is known as systematic risk. The ultimate aim of this thesis will therefore be to improve the work by Markowitz and Sharpe to the reduce systematic risk by reallocating the portfolio weights and move along the CML. At times when the volatility is high and the expected return low, a larger portion of the capital will be invested in the risk-free asset. When the opposite occurs, high expected return at low volatility, a larger portion of the available capital will be invested in the market portfolio. By investing with this strategy the authors hope to gain a higher return than the market portfolio is able to provide. This is known as *dynamic portfolio optimization*, and the term comes from the idea that allocation of the portfolio will be continuously reevaluated as new information regarding the current state of the financial market becomes available.

1.2 Current knowledge

In Bulla et al. (2011) it is concluded that it is possible to construct successful asset allocation strategies with the use of hidden Markov models (HMMs). An adaptive HMM with normal conditional distributions has also been proven to be a good fit for the data (Nystrup, Madsen, and Lindström, 2016). The fit is even better when leaving out less than 0.1 % of the observations that are most exceptional.

Nystrup, Madsen, and Lindström (2017) continues their investigations to implement a dynamic portfolio optimization strategy, by adaptively estimating the market returns using a HMM, with a number of states and different distributions. These

estimates are then used to trade a risky asset. An adaptive hidden Markov model implies that its parameters are time-varying and adaptively estimated. A variety of techniques of implementing this adaptive estimation exists in available literature. Below is an attempt to list some classes of adaptive estimation techniques:

Rolling window (RW) based implementations. The estimation is done by maximizing the likelihood for a rolling window of observations.

Generalized autoregressive score (GAS) model (Creal, Koopman, and Lucas, 2013), is based upon updating the parameters by using the gradient of the likelihood function (also known as score function). The algorithm in this thesis will be of the GAS model type.

Expectation-Maximization (EM) estimations are based on updating the parameters by using an EM algorithm as new observations are made available. See for example Nystrup et al. (2017, pp. 9-12).

For these different classes of adaptive HMM models the authors have identified two major subjects of which there exists literature. The first subject is investigating how well these models replicate some of the stylized facts of financial time series. The second is to construct a trading or asset allocation algorithm. A summary of some articles related to each subject and model are available in table 1.1, where one can also find appropriate literature in which the solution is provided. The table is also divided into whether the models examined in the articles uses Student's t-distribution or not.

Table 1.1: Table of subjects and classes of adaptively estimated HMM's containing literature covering them.

Class	t	Stylized facts	Trading algorithm
RW	No	Nystrup, Madsen, and Lindström (2016)	
RW	Yes	Nystrup, Madsen, and Lindström (2016)	
GAS	No	Nystrup, Madsen, and Lindström (2016)	Nystrup, Madsen, and Lindström (2017)
GAS	Yes		This thesis
EM	No		Nystrup et al. (2017)
EM	Yes		

1.3 Objective

The objective of this thesis will be to improve existing algorithms in order to further increase the return obtained using the investment strategy of combining an adaptive HMM with MPC.

In order for an improvement to be made the authors has chosen a HMM with two states; bull and bear. A *bull market* is a commonly used to term that refers to a market that is surging and with low volatility. A *bear market* refers to the opposite. The returns are represented with a normal distribution for the bull state case and the more heavy tailed Student's t-distribution in the bear state. Student's t-distribution will then hopefully be a better approximation to the most extreme observations. The improvement can also be seen as replacing the rolling window with a different adaptive estimation with exponentially decaying weighting of older observations.

1.4 Research questions

1. Will the algorithm be able to outperform the market index?
2. Can Bayesian optimization be used in order to optimize the hyperparameters of the trading algorithm?
3. Does short selling the risky asset improve the return of the trading algorithm, assuming no risk aversion? Will an increase in risk be compensated with an equal, or higher, Sharpe ratio as the long-only portfolio?

Chapter 2

Theory

To fully facilitate an understanding of the theory presented in this chapter, algorithm 2.1 illustrates an outline of the trading algorithm. One iteration of the loop could be one day, or one hour, depending on the preferences of the user and the data available. The underlying concept of the model is to make predictions of the, currently unknown, market returns for future days. Based on these predictions a decision is to be made whether to buy or sell the risky asset. It goes without saying that the better the estimations are the better the trades will be. The market estimations are to be performed using a hidden Markov model (HMM) in order to gain an estimate if the market is in a bull or bear state. If the model suggests, with a high probability, that the market is in a bull state a larger portion of the available capital will be allocated in the risky asset. Of course, the opposite will occur if the model estimate the market to be in a bear state.

This chapter begins with presenting the theory and implementation details by breaking down the trading algorithm into three parts, each presented in a separate section. The reader is referred to algorithm 2.1 for a short outline of how the loop is constructed. The three parts, or steps, can be summarized as: estimation, prediction and selection. The portfolio for which the algorithm dynamically optimizes consists of a risky asset and cash, deposited at the risk-free rate assumed to be $r_{\mathrm{f}} = 0$.

Algorithm 2.1 Simple psuedocode of dynamic portfolio allocation loop. A more detailed description can be found in algorithm 2.2.

```
Given hyperparameters and data
Initialize algorithm
repeat
    Update the estimated HMM parameters using the most recent observation.
    Make predictions of expected return and volatility in the market.
    Calculate the optimal trade based on the predictions and constraints.
    Execute the trade.
until the end of time
```

After the three steps has been described, section 2.4 describes how the parts are put together to construct the algorithm itself as well as how back-testing is done. Definitions of some performance measurements (also known as KPIs), calculated from the algorithm's resulting strategies, are presented there as well.

Finally, section 2.5 presents a technique for Bayesian optimization, which is used to tune and optimize the hyperparameters of the algorithm. It should be noted that Bayesian optimization was initially chosen to be the main focus of this thesis, prior to being changed to regime-based dynamic portfolio optimization.

2.1 Hidden Markov model

The model that attempts to replicate the behavior of the underlying financial market conditions is referred to as a hidden Markov model (HMM), described by Nystrup, Madsen, and Lindström (2016). The purpose of the model is basically to determine whether the market trend is "optimistic", which corresponds to a bull market state, or "pessimistic", which corresponds to a bear market state. The HMM also attempts to model what the implications of being in the respective state is. These implications are quantified as the HMM's parameters and they are allowed to vary in time as well.

In a HMM the distribution of the observations y_t will depend only on the state S_t of an underlying, unobservable, Markov chain. The sequence of states, S_t, is said to satisfy the Markov property (Nystrup, Madsen, and Lindström, 2015, p. 1532):

$$\begin{cases} \mathbb{P}(y_t \mid S_{1:T}, y_{-1}) & = \mathbb{P}(y_t \mid S_t) \\ \mathbb{P}(S_t \mid S_{1:t-1}) & = \mathbb{P}(S_t \mid S_{t-1}). \end{cases} \tag{2.1}$$

Since the HMM is very central for the understanding of the algorithm a graphical interpretation of equation (2.1) is shown in figure 2.1. It is worth noting that the return y_t will not depend on previous returns, $y_{0:t-1}$, but merely the state of the market, S_t, at time t.

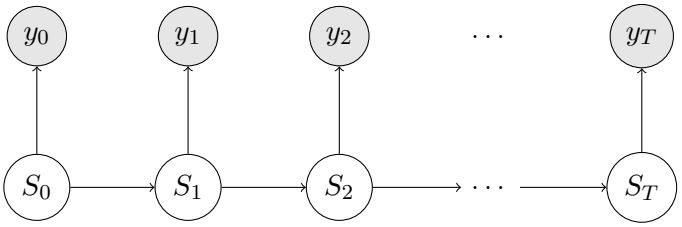

FIGURE 2.1: Illustration of a hidden Markov process.

As previously stated, the Markov chain will consist of two states with two different distributions of the returns. The first state will represent a bull market with normal distributed returns, and the second represents the bear market state with Student's t-distributed returns:

$$\mathbb{P}(y_t \mid S_t) = f_{S_t}(y_t) \sim \begin{cases} \mu_1 + \sigma_1 \epsilon_1, & \epsilon_1 \sim \mathcal{N}(0,1) \quad \text{if } S_t = 1 \\ \mu_2 + \sigma_2 \epsilon_2, & \epsilon_2 \sim t(\nu) \quad \text{if } S_t = 2 \end{cases}. \tag{2.2}$$

The conditional t-distribution in the bear state was chosen to make the algorithm more robust to statistical outliers in the returns (Nystrup, Madsen, and Lindström, 2016, p. 1000). The t-distribution is more heavy-tailed than the normal distribution, which is one of the stylized facts of financial time series (Lindström, Madsen, and Nielsen, 2015, p. 10).

The Markov chain setup is called a two-state HMM and will yield a transition matrix, i.e. a matrix consisting of the probabilities of transitioning to a state S_{t+1} at time $t+1$, given the state S_t at time t (Nystrup, Madsen, and Lindström, 2015, p. 1532):

$$\mathbb{P}(S_{t+1} = j \mid S_t = i) = \gamma_{ij} = [\mathbf{\Gamma}]_{ij} \tag{2.3}$$

$$\mathbf{\Gamma} = \begin{bmatrix} \gamma_{11} & \gamma_{12} \\ \gamma_{21} & \gamma_{22} \end{bmatrix} = \begin{bmatrix} \gamma_{11} & 1-\gamma_{11} \\ 1-\gamma_{22} & \gamma_{22} \end{bmatrix}. \tag{2.4}$$

A property that follows from (2.1) is that some expected value dependent on an observation from the HMM, given information $\mathcal{F}$, can be described as a convex combination:

$$\mathbb{E}\big[f(y_t) \mid \mathcal{F}\big] = \sum_{i=1}^{2} \mathbb{P}\big(S_t = i \mid \mathcal{F}\big) \cdot \mathbb{E}\big[f(y_t) \mid S_t = i\big] \tag{2.5}$$

under the condition that $\mathbb{E}[f(y_T) \mid S_t = i]$ exists for $i = 1, 2$.

2.1.1 Parameter estimation

Just like the HMM discussed by Nystrup, Madsen, and Lindström (2016, p. 992), the parameters will be allowed to vary in time, and the estimation is therefore done adaptively. Estimation of parameters for time series is traditionally done by maximizing the likelihood function, using the log-likelihood (LL) function, with respect to the parameters. The LL function of an HMM and its gradient is calculated using the Lystig-Hughes algorithm. The algorithm is described in detail in subsection 2.1.2.

A useful feature of the LL function is that it can be expanded into a sum of conditional LL functions using Bayes' theorem. Since it also makes sense to consider recent observations more important than older observations, it is possible to weight the terms in the log-likelihood function accordingly This results in the weighted LL function $\tilde{\ell}$. The weighted maximum likelihood (ML) estimator is thus defined as:

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \sum_{n=1}^{t} w_n \log\big(\mathbb{P}(y_t \mid y_{1:t-1}, \boldsymbol{\theta}\big) = \arg\max_{\boldsymbol{\theta}} \tilde{\ell}_t(\boldsymbol{\theta}) \tag{2.6}$$

where $y_{1:t}$ denotes the set of past observations $\{y_\tau\}_{\tau=1}^{t}$. Regular ML estimation corresponds to $w_n = 1$. The chosen weights in this report are exponential weights $w_n = \lambda^{t-n}$. where $0 < \lambda < 1$ is the forgetting factor. The forgetting factor is used similarly in recursive least squares estimation (Jakobsson, 2015, p. 278). The forgetting factor is calculated using the effective memory length N_{eff}:

$$N_{\text{eff}} = \frac{1}{1-\lambda}. \tag{2.7}$$

Since the weights are exponentially decaying, the weighted LL can be approximated with a truncated weighted LL function of a large enough length L.

Definition 2.1. A truncated weighted log-likelihood function is defined as:

$$\tilde{\ell}_{t,L}(\boldsymbol{\theta}) = \sum_{n=t-L+1}^{t} w_n \log\big(\mathbb{P}(y_t \mid y_{1:t-1}, \boldsymbol{\theta})\big) \to \tilde{\ell}_t(\boldsymbol{\theta}) \text{ as } L \to \infty \tag{2.8}$$

where $L \geq 1$ denotes the length (i.e. number of observations considered), w_n the weights and the y_t the observations.

A recursive estimation method, derived by maximizing the second-order Taylor expansion of $\tilde{\ell}$, can be written as (Nystrup, Madsen, and Lindström, 2016, p. 993):

$$\hat{\boldsymbol{\theta}}_t = \hat{\boldsymbol{\theta}}_{t-1} - \mathbf{H}_t^{-1} \nabla_{\boldsymbol{\theta}} \tilde{\ell}_t(\hat{\boldsymbol{\theta}}_{t-1}) \tag{2.9}$$

where $\mathbf{H}$ is the Hessian of $\tilde{\ell}$:

$$\mathbf{H}_t = \nabla^2_{\boldsymbol{\theta}\boldsymbol{\theta}} \tilde{\ell}_t(\hat{\boldsymbol{\theta}}_{t-1}).$$

This can be seen as a special case of a GAS model in Creal, Koopman, and Lucas (2013) or a second order stochastic approximation search (Spall, 2003, pp. 116-120). It can also be seen as a single step of Newton's optimization method and it is thus a necessary for the Hessian to be negative definite in order for the estimator to converge to a maximum (Boyd and Vandenberghe, 2004, pp. 71, 484).

The parameters must be unconstrained in order for equation (2.9) to converge. The parameters are therefore transformed in order for this condition to be fulfilled. The standard deviations σ_1, σ_2 and degrees of freedom ν are logarithmically parametrized (ν with an offset of 2 to ensure that the variance of the Student's t-distribution exists). Transition probabilities γ_{11} and γ_{22} are transformed with the logistic function. The reader is referred to appendix A.2 for full details regarding the parameter transformations. A summary of the HMM's parameters can be found in table 2.1.

The Hessian can be approximated, scaled and manipulated to suit the needs of the user. In this report the Hessian is approximated recursively as:

$$\hat{\mathbf{H}}_t = \hat{\mathbf{H}}_{t-1} - \frac{1}{t - t_0}\left(\left[\nabla_{\boldsymbol{\theta}} \tilde{\ell}_{t,L}(\hat{\boldsymbol{\theta}}_{t-1})\right]\left[\nabla_{\boldsymbol{\theta}} \tilde{\ell}_{t,L}(\hat{\boldsymbol{\theta}}_{t-1})\right]^{\mathrm{T}} + \hat{\mathbf{H}}_{t-1}\right). \tag{2.10}$$

The choice was made based on that the approximation above gave an acceptable trade-off between stability and variety of the estimated parameters. The resulting estimator for the parameters is then:

$$\hat{\boldsymbol{\theta}}_t = \hat{\boldsymbol{\theta}}_{t-1} - \frac{A}{N_{\text{eff}}}\left(\left[\hat{\mathbf{H}}_t\right]^{-1} \nabla_{\boldsymbol{\theta}} \tilde{\ell}_{t,L}(\hat{\boldsymbol{\theta}}_{t-1})\right) \tag{2.11}$$

Note that the Hessian's approximation also needs to be negative definite. There are many methods for modifying the Hessian (or its approximation) to ensure this numerically (Spall, 2003, p. 28). One such method could be to add a small value to it's diagonal until all of the Hessian's eigenvalues are negative, which is equivalent to the Hessian being negative definite (Boyd and Vandenberghe, 2004, p. 647). Although this kind of problem was not an issue with the implementation in this report, so no such modifications were done.

Other approximations of the Hessian not used in this thesis are found in the appendix A.3. Also note that usage of the truncated LL in equation (2.10) ensures that the computational complexity is limited as time passes and more observations are made available. This is useful, since otherwise the full LH algorithm would have to be run every time a new return is observed.

In order to initialize the HMM parameters at time t_0, the weighted ML estimate, was utilized. The ML estimate utilized the L_0 observations before t_0. Thus:

$$\hat{\boldsymbol{\theta}}_0 = \arg\max_{\boldsymbol{\theta}} \tilde{\ell}_{t,L_0}(\boldsymbol{\theta}). \tag{2.12}$$

The approximated Hessian is initialized as

$$\hat{\mathbf{H}}_0 = -\sum_{t=t_0^-}^{t_0} \left[\nabla_{\boldsymbol{\theta}} \mathbb{P}(y_t \mid y_{t_0^-:t}, \hat{\boldsymbol{\theta}}_0)\right]\left[\nabla_{\boldsymbol{\theta}} \mathbb{P}(y_t \mid y_{t_0^-:t}, \hat{\boldsymbol{\theta}}_0)\right]^{\mathrm{T}} \tag{2.13}$$

where $t_0^- = t_0 - L_0 + 1$.

TABLE 2.1: Summary of parameters of the HMM.

Parameter	Description	Transformation
γ_{11}	Probability to stay in bull state	Logistic
γ_{22}	Probability to stay in bear state	Logistic
μ_1	Mean of returns in bull state	None
μ_2	Mean of returns in bear state	None
σ_1	Standard deviation of returns in bull state	Logistic
σ_2	Scaling parameter for returns in bear state	Logistic
ν	Degrees of freedom for returns in bear state	Logistic & offset 2

2.1.2 Lystig-Hughes algorithm

The LL-function of the HMM, and it's gradient (also known as the score function), can be calculated with the Baum-Welch algorithm, a special case of the EM-algorithm applied to HMMs (Lystig and Hughes, 2002, p. 679). The original algorithm provided by Baum (1972) can compute the likelihood L_T with a computational complexity which is linear in T. The Baum-Welch algorithm is sometimes referred to as the forward-backward algorithm. A revised version of this algorithm, which is refereed to as the Lystig-Hughes (LH) algorithm, will be presented below. The algorithm is based on calculating the likelihood as the product of the conditional probabilities. In order to start the algorithm, the initial probabilities $\lambda_1(j)$ are calculated as:

$$\lambda_1(j) \equiv \mathbb{P}(y_1, S_1 = j) = f_j(y_1)\delta_j. \tag{2.14}$$

Here f_j denotes the densities imposed by state j from equation (2.2). The initial distribution of the states are denoted $\mathbb{P}(S_1 = j) = \delta_j$. The LH algorithm then continues by calculating the forward probabilities, $\lambda_t(j)$, for a two-state HMM as:

$$\lambda_t(j) \equiv \mathbb{P}(y_t, S_t = j \mid y_{1:t-1}) = \frac{f_j(y_t)}{\Lambda_{t-1}} \sum_{i=1}^{2} \gamma_{ij}\lambda_{t-1}(i),\ 2 \leq t \leq T, \tag{2.15}$$

where γ_{ij} are the transition probabilities, as in equation (2.3), and where:

$$\Lambda_t = \mathbb{P}(y_t \mid y_{1:t-1}) = \sum_{i=1}^{2} \lambda_t(i). \tag{2.16}$$

This allows the LL-function to be formulated as:

$$\ell_T(\boldsymbol{\theta}) = \log\left(L_T(\boldsymbol{\theta})\right) = \sum_{t=1}^{T} \log(\Lambda_t) \tag{2.17}$$

and more importantly, the truncated weighted LL as:

$$\tilde{\ell}_{T,L}(\boldsymbol{\theta}) = \sum_{t=1}^{L} w_t \log(\Lambda_t) \tag{2.18}$$

with some abuse of notation. The algorithm is initialized according to equation (2.14) using observation y_{T-L+1}. The truncated weighted LL-function is defined in definition 2.1.

In order to calculate the score function, the algorithm is initiated using the following equation:

$$\psi_1(j,\theta) \equiv \frac{\partial}{\partial\theta}\mathbb{P}(y_1, S_1 = j) = \left(\frac{\partial f_j}{\partial\theta}(y_1)\right)\delta_j + f_j(y_1)\left(\frac{\partial\delta_j}{\partial\theta}\right) \tag{2.19}$$

for every parameter θ in the model. The derivatives of the Gaussian and Student's t-distribution with regards to the distribution parameters are available in Appendix A.1. The algorithm continues by calculating:

$$\begin{aligned}\psi_t(j,\theta) \equiv & \frac{\frac{\partial}{\partial\theta}\mathbb{P}(y_{1:t}, S_t = j)}{\mathbb{P}(y_{1:t-1})} = \frac{1}{\Lambda_{t-1}}\sum_{i=1}^{2}\Bigg\{\psi_{t-1}(i,\theta)f_j(y_t)\gamma_{ij} + \\ & + \lambda_{t-1}(i,\theta)\left(\frac{\partial f_j}{\partial\theta}(y_t)\right) + \lambda_{t-1}(i,\theta)f_j(y_t)\left(\frac{\partial\gamma_{ij}}{\partial\theta}\right)\Bigg\}, \quad 2 \le t \le T.\end{aligned} \tag{2.20}$$

The derivative of the LL-function can then be calculated as:

$$\frac{\partial}{\partial\theta}\log\left(\mathbb{P}(y_{1:T})\right) = \frac{\partial}{\partial\theta}\ell_T(\boldsymbol{\theta}) = \frac{\Psi_T(\theta)}{\Lambda_T}, \tag{2.21}$$

and similarly the derivative of the truncated weighted LL-function as:

$$\frac{\partial}{\partial\theta}\tilde{\ell}_{T,L}(\boldsymbol{\theta}) = w_1\frac{\Psi_1(\theta)}{\Lambda_1} + \sum_{t=2}^{L} w_t\left(\frac{\Psi_t(\theta)}{\Lambda_t} - \frac{\Psi_{t-1}(\theta)}{\Lambda_{t-1}}\right). \tag{2.22}$$

with similar notation as in equation (2.18), and where:

$$\Psi_t(\theta) = \sum_{i=1}^{2}\psi_t(i,\theta). \tag{2.23}$$

The derivatives can then be used to construct the gradient.

The probability of being in state $S_t = i$ at time t given past observations, also known as the forward probability, can be calculated as:

$$\alpha_{t|t}(i) \equiv \mathbb{P}(S_t = i \mid y_{1:t}) = \frac{\lambda_t(i)}{\Lambda_t}. \tag{2.24}$$

Using equation (2.1) and (2.3) the probability of being in a certain state at future times can also be predicted. This will be used when calculating future predictions of the returns.

$$\begin{bmatrix}\alpha_{\tau|T}(1)\\ \alpha_{\tau|T}(2)\end{bmatrix}^{\mathrm{T}} = \begin{bmatrix}\mathbb{P}(S_\tau = 1 \mid y_{1:T})\\ \mathbb{P}(S_\tau = 2 \mid y_{1:T})\end{bmatrix}^{\mathrm{T}} = \begin{bmatrix}\lambda_T(1)/\Lambda_T\\ \lambda_T(2)/\Lambda_T\end{bmatrix}^{\mathrm{T}}\boldsymbol{\Gamma}^{(\tau-T)}, \; T \le \tau. \tag{2.25}$$

2.2 Prediction

Using the estimation technique for the HMM as described in section 2.1, one will obtain estimates of the HMM parameters as listed in table 2.1. In addition, the transition probabilities γ_{11}, γ_{22} and $\alpha_{t|t}(1)$ will be obtained as well.

In order to be able to use MPC, one then have to make predictions of the portfolio value, or return, for future points in time. Naturally, better predictions will yield a higher return in the portfolio. It will therefore be of utter importance to make these estimates as good as possible. Due to the formulation of the MPC optimization

problem, the ordinary daily returns, q, need to be predicted instead of the log-returns, r. One important attribute of the ordinary return is that it is supported on $[-1, \infty]$ but the log-returns are supported on $[-\infty, \infty]$.

Prediction of ordinary returns can be done using the HMM's parameters and the moment generating function (MGF) M. A demonstration of a simple example, for the log-normal case, is:

$$r \sim \mathcal{N}(\mu, \sigma) \implies \mathbb{E}[1 + q] = \mathbb{E}[e^r] = M(1) = e^{\mu + \sigma^2/2}. \tag{2.26}$$

One problem with using the Student's t-distribution in the bear state is that the MGF for the Student's t-distribution simply does not exist. In order to solve this problem a compromise had to be made. Specifically that the log-returns could no longer fall above or below a certain threshold. In other words, the distributions of the log-returns were truncated for the benefit of allowing the ordinary returns to be predicted. One explanation for truncating the distribution is that an exchange would issue a trading halt on the asset in question if the price would change too swiftly. This is also known as a *circuit breaker*. Nasdaq actually implements a circuit breaker based on the S&P-500 index on a threshold of 7%-20% depending on the time of day (Nasdaq, 2017).

The new truncated distributions of the log-returns can be converted to ordinary returns using what will be called the truncated MGF, notated as $\tilde{M}$. The truncation will be made at the bounds a and b. The bounds a and b could be dependent on the parameters but they were decided to be constant in this report. For further clarification, in contrast to equation (2.26), the expected ordinary daily return is now formulated as:

$$r \sim \mathcal{N}(\mu, \sigma) \implies \mathbb{E}[1 + q] = \mathbb{E}[e^r \mid a \leq r \leq b] = \tilde{M}(1) \tag{2.27}$$

and is generalized as:

Definition 2.2. The truncated MGF of a stochastic variable V truncated from below at a and from above at b is defined as:

$$\tilde{M}_V(t) = \mathbb{E}\big[e^{tV} \mid a \leq V \leq b\big]. \tag{2.28}$$

The special case of $a = -\infty$ corresponds to no truncation from below and $b = \infty$ to no truncation from above. Hence the truncated MGF is equal to the regular MGF if $(a, b) = \mp\infty$.

By using all parameters produced by the HMM estimation step described in section 2.1.1 the MGFs are evaluated for each state. The reader is referred to figure 2.2 for an illustrative representation of the procedure.

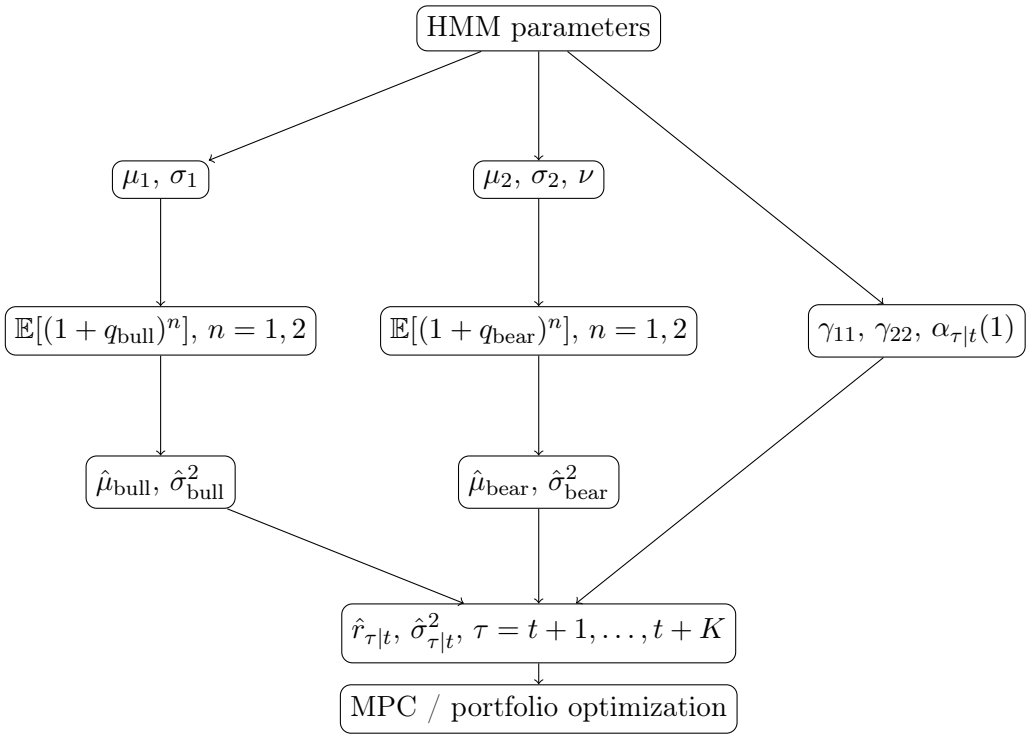

FIGURE 2.2: Image representation of the prediction scheme.

Firstly the truncated MGFs $\tilde{M}_G$ and $\tilde{M}_S$ of the Gaussian and Student's t-distribution respectively (c.f. subsection 2.2.1 and 2.2.2) are used to calculate moments of ordinary returns. These moments have to be calculated at the first and second orders ($n = 1$ and $n = 2$ in figure 2.2). This part can be summarized as calculating the predicted daily return and it's volatility using the following equations:

$$\hat{r}_{\text{bull}|t} = \mathbb{E}[q_{\text{bull}}] = \tilde{M}_G(1;\Theta_t) - 1 \tag{2.29}$$

$$\hat{r}_{\text{bear}|t} = \mathbb{E}[q_{\text{bear}}] = \tilde{M}_S(1;\Theta_t) - 1 \tag{2.30}$$

$$\hat{\sigma}^2_{\text{bull}|t} = \text{Var}[q_{\text{bull}}] = \tilde{M}_G(2;\Theta_t) - \left(\tilde{M}_G(1;\Theta_t)\right)^2 \tag{2.31}$$

$$\hat{\sigma}^2_{\text{bear}|t} = \text{Var}[q_{\text{bear}}] = \tilde{M}_S(2;\Theta_t) - \left(\tilde{M}_S(1;\Theta_t)\right)^2 \tag{2.32}$$

where the parameters and hyperparameters (including bounds a and b) at time t are denoted as Θ_t.

Lastly, the transition matrix $\mathbf{\Gamma}$ together with forward probability $\alpha_{t|t}(i)$ are used to predict the expected return $\hat{r}_{\tau|t}$ and the volatility $\hat{\sigma}_{\tau|t}$ as a mixture with the use of equation (2.5) and (2.25). This result in the following two equations:

$$\hat{r}_{\tau|t} = \alpha_{\tau|T}(1) \cdot \tilde{M}_G(1;\Theta_t) + \alpha_{\tau|T}(2) \cdot \tilde{M}_S(1;\Theta_t) \tag{2.33}$$

$$\hat{\sigma}^2_{\tau|t} = \alpha_{\tau|T}(1) \cdot \tilde{M}_G(2;\Theta_t) + \alpha_{\tau|T}(2) \cdot \tilde{M}_S(2;\Theta_t) - \hat{r}^2_{\tau|t}. \tag{2.34}$$

2.2.1 Truncated MGF of Gaussian distribution

For a normally distributed variable $G \sim \mathcal{N}(\mu, \sigma^2)$, the truncated MGF bound at a and b can be derived from the truncated log normal distribution (Jawitz, 2004):

$$\tilde{M}_G(t;\Theta) = \exp\left(t\mu + \frac{t^2\sigma^2}{2}\right) \cdot \frac{\phi(\beta - \sigma t) - \phi(\alpha - \sigma t)}{\Phi(\beta) - \Phi(\alpha)} \tag{2.35}$$

where Θ is the set of all relevant parameters and ϕ is the PDF of the standard normal distribution. The normalized lower and upper bounds are defined as:

$$\alpha = \frac{a-\mu}{\sigma}, \qquad \beta = \frac{b-\mu}{\sigma}.$$

2.2.2 Truncated MGF of Student's *t*-distribution

A non-standardized Student's t-distributed variable S can be expressed as $S = \mu + \sigma X$ where $X \sim t(\nu)$ is a standardized Student's t random variable. The truncated MGF of S is then expressed as:

$$\tilde{M}_S(t;\Theta) = e^{t\mu} \cdot \tilde{M}_X(\sigma t;\Theta), \tag{2.36}$$

where $\tilde{M}_X(t)$ is the truncated MGF of X (Kim, 2008):

$$\tilde{M}_X(t;\Theta) = \frac{\int_0^\infty e^{t^2/2\eta}\left(\Phi(\eta^{1/2}\beta - \eta^{-1/2}t) - \Phi(\eta^{1/2}\alpha - \eta^{-1/2}t)\right) \mathrm{d}H(\eta)}{F_\nu(b) - F_\nu(a)}. \tag{2.37}$$

Here Θ is the set of all relevant parameters and Φ is the CDF of the standard normal distribution. The normalized lower and upper bounds are defined as

$$\alpha = \frac{a-\mu}{\sigma}, \qquad \beta = \frac{b-\mu}{\sigma}.$$

The measure $\mathrm{d}H$ is the PDF of the gamma distribution with shape parameter $k = \nu/2$ and scale parameter $\theta = 2/\nu$. The PDF of the gamma distribution is given by (Johnson, Kotz, and Balakrishnan, 1994, p. 337):

$$h(x) = \frac{\mathrm{d}H}{\mathrm{d}x}(x) = \frac{1}{\Gamma(k)\cdot\theta^k} \cdot x^{k-1} \cdot \exp\left(-\frac{x}{k}\right). \tag{2.38}$$

The implementation used in this report utilized the `integral` function in MATLAB to evaluate the integral in equation (2.37), which uses an adaptive quadrature method (Shampine, 2008). Since the predictions are an important part of the algorithm, the stability and accuracy of evaluating the integral is also of importance. The `integral` function was invoked with relative tolerance 10^{-6} and absolute tolerance 10^{-10}. The integration interval in (2.37) was started at a small positive number ξ, instead of 0, which was found not to produce any `NaN` values.

One alternative for evaluating (2.37) is to use generalized Laguerre-Gauss quadrature, but this was not explored this thesis. Golub and Welsch (1969) shows how it is possible to find the quadrature rules needed to do this.

2.3 Portfolio Selection

Since the work of Merton (1969), and others, on formulating and examining the portfolio optimization problem, portfolio selection has since become more dynamic. Optimization of the portfolio has gone from single-period models to multi-period models, namely it is preferred to view trades of a longer time horizon than just one day ahead. The main reason for multi-period optimization as opposed to single-period is, in this case, a better cope of transaction costs and other constraints (Boyd et al., 2017, p. 45). For example, if the algorithm wishes to take a long position in an asset which is expensive to sell, it may be that the best decision is not to trade at all. The transaction costs related to unwinding the position will not be captured in the single-period model. However, the multi-period model may take both the costs of buying and selling the asset into account, which is highly preferable.

Portfolio optimization with a longer time-horizon has proven to be very different from the short term optimization. The goal of this thesis is to make a good portfolio with multi-period optimization and a long term investment strategy. In this case it is simply not possible to make better estimates than the average. The problem can therefore be simplified to choosing the trades over the next few days or weeks, which in turn will make the problem computationally feasible (Nystrup et al., 2017, p. 3). The reader is also reminded that the portfolio is constructed using one risky asset which is modelled using the HMM, and cash deposited at the risk-free rate $r_{\mathrm{f}} = 0$.

The first two following subsections describes how this approach of portfolio selection is done. The approach is referred to as model predictive control (MPC) in Nystrup, Madsen, and Lindström (2017), and has proven to be useful in combination with a HMM model similar to the one in this thesis. The solution to the specific optimization problem will then be described in the third section. The solution is executed and propagated into the next day (in the future) in the last section.

2.3.1 Model predictive control

The problem can be formulated as a stochastic control problem. Every day a decision is to be made in whether to trade the asset or not. A natural formulation in order to choose the trade is to maximize the expected portfolio value V_t over the future time horizon, K, which the multi-period model is decided upon. One can also make use of penalties for the chosen trades and positions during the horizon, ψ_τ, representing a risk- or trading aversion. The problem is therefore (Nystrup, Madsen, and Lindström, 2017, p. 5):

$$\text{maximize} \quad \mathbb{E}\Big[V_{t+K} - \sum_{\tau=t}^{t+K} \psi_\tau\Big]. \tag{2.39}$$

The value of the portfolio V_{t+K} at the end of the horizon is, of course, a stochastic variable which depends on the returns. The penalties, ψ_τ, could be stochastic as well. In addition, the problem will be subject to a set of constraints, for example no use of leverage or short selling the asset.

The maximization problem, equation (2.39), can be reformulated using simplifications and by replacing expected values with market predictions. The portfolio weight of the traded asset is denoted with w_τ at time τ. It may then be understood that the portion of the available capital allocated in cash is $1 - w_\tau$. For the purpose of this

thesis, the MPC problem was decided to be formulated as:

$$\begin{aligned} \text{maximize} \quad & \sum_{\tau=t+1}^{t+K} \Big(\hat{r}_{\tau|t} w_\tau - \gamma^{\text{risk}} \hat{\psi}^{\text{risk}}_{\tau|t}(w_\tau) - \gamma_1^{\text{trade}} \hat{\psi}_1^{\text{trade}}(w_\tau - w_{\tau-1}) \\ & - \gamma_2^{\text{trade}} \hat{\psi}_2^{\text{trade}}(w_\tau - w_{\tau-1}) \Big) \\ \text{subject to} \quad & w_{\min} \leq w_\tau \leq w_{\max}, \quad \tau = t+1, \ldots, t+K. \end{aligned} \tag{2.40}$$

For further clarification, a list of explanations of the equation's parts are listed below:

Weights w_τ is the variable which the problem is optimized with respect to. The current weight w_t is a given constant, since it is known what the current portfolio allocation is.

Predictions of future returns $\hat{r}_{\tau|t}$ and volatilities $\hat{\sigma}^2_{\tau|t}$ as described in section 2.2.

Aversion functions or penalty functions $\hat{\psi}$ are used to define what positions or trades the algorithm should avoid. These are discussed later in section 2.3.2.

Aversion parameters γ are scalars multiplied with respective aversion function. The purpose is to scale the aversion in the MPC strategy. The parameters are briefly discussed in 2.3.2.

Weight limits $w_{\max}$ and $w_{\min}$ are used to limit the use of leverage and short selling of the traded asset. A common choice is to use $w_{\min} = 0$ and $w_{\max} = 1$ (Boyd et al., 2017, p. 34).

2.3.2 Aversions

A variety of aversion functions are described in Boyd et al. (2017) and new functions can easily be designed to suit the needs of the user. The aversion functions used in this thesis will be described below.

The measurement of risk is traditionally measured by the variance, or volatility, of the traded asset. The measure was already used to optimize portfolios in the article by Markowitz (1952). The penalty for risk aversion, $\hat{\psi}^{\text{risk}}_{\tau|t}$, may be set as the estimated variance of the portfolio given the weights, and is thus defined as:

$$\hat{\psi}^{\text{risk}}_{\tau|t}(w_t) = w_t^2 \hat{\sigma}^2_{\tau|t} \tag{2.41}$$

where $\hat{\sigma}^2_{\tau|t}$ is calculated according to equation (2.34). The trading aversion can be set as the size of the trade, with a linear penalty:

$$\hat{\psi}_1^{\text{trade}}(z) = |z| \tag{2.42}$$

or a quadratic penalty, which could be explained by the fact that the size of a trade can move the price. The quadratic trading penalty is therefore used to discourage large trades (Boyd et al., 2017, p. 42) and is defined as:

$$\hat{\psi}_2^{\text{trade}}(z) = |z|^2 = z^2 \tag{2.43}$$

The non-negative scalar values γ^{risk}, γ_1^{trade} and γ_2^{trade} are used to scale the penalties imposed by the aversion functions. They can be chosen freely in order to tune the algorithm. In this thesis they will be regarded as hyperparameters which will be inserted by the user to the algorithm. They can also be defined as functions depending

on past portfolio performance. One example, by Nystrup et al. (2017), is when γ^{risk} is defined as a function in an attempt to limit the maximum drawdown of the portfolio.

2.3.3 Solving the optimization problem

Since the MPC approach requires a maximization problem to be solved, an optimization algorithm is to be chosen. The problem in (2.40) is a convex optimization problem (Boyd et al., 2017, p. 38). The implementation used in this thesis utilized the `CVXGEN` package to generate a tailored optimization algorithm for a specific time horizon K. The reader is referred to Mattingley and Boyd (2012) for information regarding the `CVXGEN` package, which is specifically designed for convex optimization problems in `Matlab`.

An alternative is to use slack variables to express the absolute value in equation (2.42), which allows the problem to instead be specified as a quadratic programming (QP) optimization problem (Boyd and Vandenberghe, 2004, p. 152). A third alternative is to reformulate the maximization problem as an ℓ_1-norm regularization problem (Boyd and Vandenberghe, 2004, p. 308). Both QP and ℓ_1-norm regularization problems are a well studied families of optimization problems and there are plenty of resources on numerical solvers for them.

2.3.4 Propagating decisions into the future

The optimization problem in equation (2.40) is to be solved for every day, $0 \leq t \leq T$, available in the data set. For each day, the weights, $w_{t+1}, \ldots, w_{t+K}$, over the future time horizon, K, will be determined. As stated before, the only trade executed is to go from the current weight, w_t, to the calculated weight for the first day of the time horizon, w^*_{t+1}. The other weights will not be used, and they are updated the next day when more information about the market has been made available. It is thus needed to describe how the value of the portfolio, and the capital allocated to the traded asset, changes due to trading throughout the period.

Trading and holding costs are important factors to take into consideration when designing the trading algorithm. In this thesis, the brokerage fee is a cost described as a linear function to the absolute size of the trade. The total value of the commission is thus reduced by some amount if trading is to be done before the next period, when the portfolio value is expected to be higher. The relationship between the portfolio value before and after the trade are described by:

$$V_t = V_t^+ + \kappa |u_t \cdot|, \quad u_t = V_t^+ w^*_{t+1} - V_t w_t \tag{2.44}$$

where κ is the commission rate, u_t the size of the trade performed and V_t^+ is the portfolio value post-trading. A realistic approximate commission rate is $\kappa = 0.001$ for an institutional investor (Nystrup, Madsen, and Lindström, 2017, p. 9) when trading ETFs. It should be noted that the commission rate could could be higher for other assets of different types and liquidities. Solving equation (2.44) allows the post-trade portfolio value to be calculated recursively as:

$$V_t^+ = V_t \cdot \min\left(\frac{1 + \kappa w_t}{1 + \kappa w^*_{t+1}} \, , \, \frac{1 - \kappa w_t}{1 - \kappa w^*_{t+1}} \right). \tag{2.45}$$

Holding costs were disregarded in this thesis, although a bit unrealistic. If the traded risky asset is an form of for example an ETF, a management fee would incur. It should be noted that when short selling, the holding costs could be even greater.

At the time when the next observation is made, the loop in algorithm 2.1 will be repeated for the post-trade portfolio. The value of the traded asset will have changed and the new value of the portfolio prior to the next portfolio update can be calculated as:

$$V_{t+1} = V_t^+ \cdot \left(w_{t+1}^* \cdot \exp(r_{t+1}) + (1 - w_t^*)\right) \tag{2.46}$$

where r_{t+1} is the realized log-return of the asset. The new current fraction, or weight, of the portfolio allocated to the asset can now be calculated as:

$$w_{t+1} = \frac{V_t}{V_{t+1}} w_{t+1}^* \cdot \exp(r_{t+1}). \tag{2.47}$$

2.4 Trading Algorithm and Back-Testing

Now the reader is supplied with enough theory to begin trading with the algorithm. A comprehensive overview, in the form of psuedocode, of how the trading algorithm is constructed using the steps presented in the previous sections can be seen in algorithm 2.2. The trading algorithm is tunable with a number of hyperparameters. These are compiled in table 2.2.

Algorithm 2.2 Psuedocode of the trading algorithm.

```
Given set of hyperparameters and data.
Initialize HMM parameters θ_0 with (2.12) and Ĥ_0 with (2.13)
for t = 1...T do
    Update the parameters θ_t for the HMM using the most recent
        observation with (2.11).
    if Do trading at period t then
        Update the predictions of r̂ and σ̂ with (2.33) and (2.34) using θ_t
        Calculate the optimal future portfolio weights, w_{t+1}, ..., w_{t+K}
         by solving (2.40).
        Execute the first trade w*_{t+1} − w_t and update portfolio
         using (2.45), (2.46) and (2.47).
    endif
endfor
Return trades u_t and portfolio values V_t.
```

Table 2.2: Summary of hyperparameters of the trading algorithm.

Parameter	Description	Used in
N_{eff}	Effective memory length in HMM	(2.11) and (2.7)
L	Truncated LL length	(2.11) and (2.10)
L_0	HMM estimation initialization length	(2.12) and (2.13)
A	HMM parameter convergence speed factor	(2.11)
a	Lower truncation bound for log-return prediction	(2.35) and (2.37)
b	Upper truncation bound for log-return prediction	(2.35) and (2.37)
K	MPC time horizon	(2.40)
γ^{risk}	Risk aversion	(2.40)
γ_1^{trade}	Linear trade aversion	(2.40)
γ_2^{trade}	Quadratic trade aversion	(2.40)
$w_{\min}$	Minimum weight constraint	(2.40)
$w_{\max}$	Maximum weight constraint	(2.40)
κ	Trading commission rate	(2.45)

The data used, i.e. the time series of log-returns, are split into three contiguous data sets or series similar to Nystrup et al. (2017, p. 13). An illustration of how the sets are divided can be seen in figure 2.3. The three sets are ordered and denoted as follows:

1. Convergence set: Firstly used to initialize the HMM parameters. No trading is done here. It is used to let the adaptive parameter estimation converge before trading is started.

2. In-sample data set: Trading starts at the beginning of this set. The performance from the trading done here are used to tune the hyperparameters.

3. Out-of-sample data set: Trading continues throughout this set. Performance from the trading done here are used to analyze and compare the results gathered from the in-sample data-set.

To evaluate the trading algorithm for a specific configuration of hyperparameters and a given data set, there are various performance measurements, referred to as key performance indicators (KPIs), that can be calculated. The formulas for calculating some relevant KPIs are presented in subsection 2.4.1.

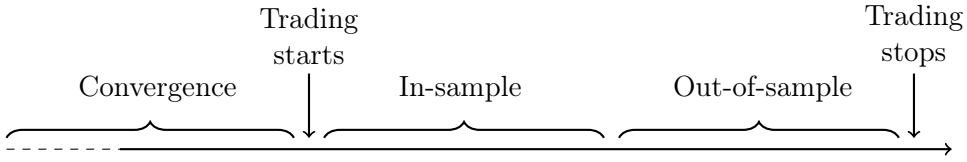

Figure 2.3: Illustration of timeline and how trading is divided into convergence, in-sample and out-of-sample data sets.

The hyperparameter optimization was done using Bayesian optimization (BO) with regards to a chosen KPI calculated from the trading done in the in-sample set. BO is covered in section 2.5. During the tuning procedure the objective function is the trading algorithm, the inputs are a subset of the hyperparameters in table 2.2 and the output is the KPI of choice. In this case the outputted KPI will be the annualized average return.

Regarding the trading algorithm, many of the hyperparameters are mutually dependent which will prove to make the in-sample training more challenging. Due to the

fact that the parameters can not be optimized one at a time but have to be optimized simultaneously (Nystrup et al., 2017, p. 15).

2.4.1 Key Performance Indicators

In order to calculate the performance measurements it is common practice to use the annualized values instead of the daily. The annualized log-returns of the portfolio are:

$$R_t = N_y \cdot \log\left(\frac{V_{t+1}}{V_t}\right) \tag{2.48}$$

where $N_y = 252$ is the number of trading days per year. The average return and standard deviation of a period of length N are calculated as:

$$\text{AR} = \bar{R}_p = \frac{1}{N}\sum_t R_t \quad \text{and} \quad \text{SD} = \sigma_p = \sqrt{\frac{1}{N}\sum_t \left(R_t - \bar{R}_p\right)^2}. \tag{2.49}$$

The Sharpe-ratio, which is a commonly used KPI, is calculated as (Sharpe, 1964):

$$\text{SR} = \bar{R}_p/\sigma_p. \tag{2.50}$$

The Calmar-ratio is defined as (Nystrup, Madsen, and Lindström, 2017, p. 9):

$$\text{CR} = \bar{R}_p/\text{MDD} \tag{2.51}$$

where MDD is the maximum drawdown:

$$\text{MDD} = \max\left\{1 - \frac{V_t}{V_s} \;:\; s < t,\; V_s > V_t\right\}. \tag{2.52}$$

The annualized average turnover (relative to the portfolio value) can be calculated as:

$$\text{AT} = \frac{1}{N}\sum_t \frac{|u_t|}{V_t} \tag{2.53}$$

where u_t is the sizes of the trades performed as in equation (2.44).

2.5 Bayesian Optimization

As previously discussed, there are a number of hyperparameters in the trading algorithm, which are up to the user to decide upon, seen in table 2.2. They can either be chosen beforehand or tuned for optimal performance of the algorithm. In order to obtain as high return as possible the specific subset of hyperparameters chosen to be tuned in this thesis will be marked as non-fixed in table 3.1.

As there is no analytical expression for the objective function, i.e. the return of the trading algorithm, it will have to be sampled with different parameter values in order to obtain the optimal set. Nystrup et al. (2017, pp. 15-16) uses a form of grid-search to optimize the MPC hyperparameters using the in sample data set. In this thesis, however, it will be attempted to make better use of the samples drawn by using Bayesian optimization (BO) and thus, hopefully, reduce the optimization time and yield better performance of the algorithm.

Bayesian optimization comes to great use when optimizing so called black box functions, i.e. functions which there is no explicit expression for but which you are

able to evaluate (Brochu, Cora, and de Freitas, 2010, p. 1). The evaluation of our trading algorithm, with respect to the hyperparameters, can be considered to be a black box function.

In order to illustrate the strength of BO, consider the following toy example originating from microeconomics: Say that you want to maximize the utility at a restaurant with respect to the price you are paying for the experience. This is a function which the is no known expression for. It is however reasonable to assume a dependence between price and utility. Further assume, that one were able to measure the utility for three different prices, 100, 200 and 400 SEK. An seen in figure 2.4 a second degree polynomial will, not surprisingly, manage to fit all the three measurements exactly. The belief that this is the correct model for the dependence will indicate that the utility will have a maximum either at zero cost or at the highest cost available, which seems unrealistic since the utility should not be ever increasing with price up to infinity.

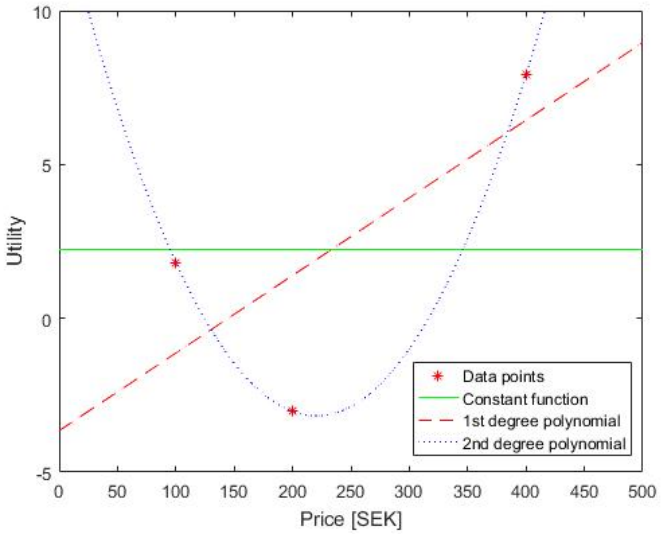

FIGURE 2.4: The data points and fitted utility functions for the restaurant example.

It is therefore impossible to know if the chosen order, p, of the polynomial is high enough. Fitting polynomials of order p will, in addition, require $p+1$ number of data points, which is not a desired property as function evaluations often are expensive.

This kind of reasoning may suggest another, non-parametric, approach to fit a function to the data set. In figure 2.5 predictions of what the utility at certain prices might be, using a Gaussian process, is illustrated. This approach leads to a much more reasonable result with a global maximum at 400 SEK, instead of an ever increasing utility with higher prices. As the true function is also displayed in this figure, one can see that the Gaussian process makes better use of the data available as the predictions are much better than the polynomial estimates above. The true curve needs to be estimated with a polynomial of an order much higher than $p = 2$, which is not possible without more data available.

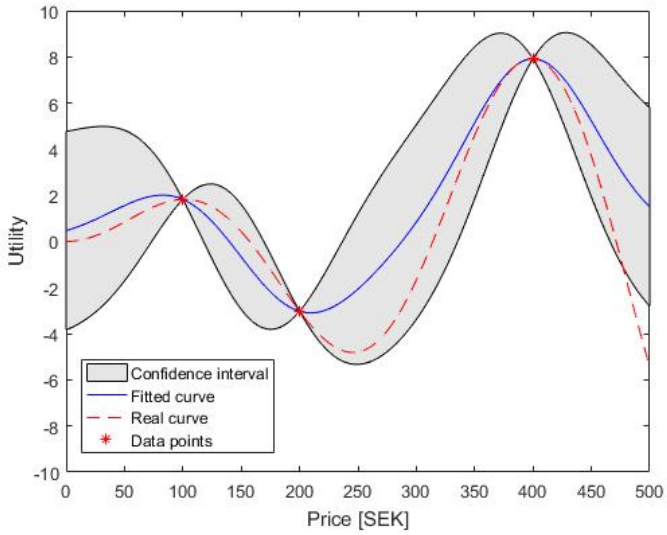

FIGURE 2.5: The real function compared with the prediction (and confidence interval) of a Gaussian process.

2.5.1 Gaussian Processes

Definition 2.3. A Gaussian process (GP) is formally defined as an infinite-dimensional stochastic process (Brochu, Cora, and de Freitas, 2010, p. 7). Every dimension correspond to a point $\mathbf{x}$ in a D-dimensional input-space $\mathbf{x} \in \chi \subset \mathbb{R}^D$. For a finite subset of input points $\mathbf{x}_i,\ i = 1, \ldots, n$ the process at those, denoted by $f(\mathbf{x_i})$ are normal distributed.

In the case of this thesis, $f(\mathbf{x_i})$ denotes the measured performance (e.g. average return) for some specific values for the tunable hyperparameters denoted $\mathbf{x}_i$. The property of normal distributed random variables has given the process its name, the Gaussian process. The Wiener process (a.k.a. Brownian Motion) is one example of a GP (Lindgren, Rootzén, and Sandsten, 2014, p. 117). If one assumes that the objective function is a GP, $f \sim \mathcal{GP}$, an implication is that the value of f at n points of $f(\mathbf{x}_i)$ is distributed as:

$$\mathbf{f}|\mathbf{X}, \boldsymbol{\theta} \sim \mathcal{N}(\mathbf{m}, \mathbf{K}) \tag{2.54}$$

where:

$$\mathbf{f} = \begin{bmatrix} f(\mathbf{x}_1) \\ \vdots \\ f(\mathbf{x}_n) \end{bmatrix}, \quad \mathbf{m} = \begin{bmatrix} m(\mathbf{x}_1) \\ \vdots \\ m(\mathbf{x}_n) \end{bmatrix}, \quad \mathbf{K} = \begin{bmatrix} k(\mathbf{x}_1, \mathbf{x}_1) & \ldots & k(\mathbf{x}_1, \mathbf{x}_n) \\ \vdots & \ddots & \vdots \\ k(\mathbf{x}_n, \mathbf{x}_1) & \ldots & k(\mathbf{x}_n, \mathbf{x}_n) \end{bmatrix} \tag{2.55}$$

$\mathbf{X}$ denotes all inputs $\mathbf{x}_i$ aggregated and m is the mean function, $\mathbf{K}$ the covariance matrix and $\boldsymbol{\theta}$ some set of parameters for $\mathbf{m}$ and $\mathbf{K}$.

The mean function was set to zero $m(\mathbf{x}) = 0$ in the implementation for this report. Another possibly viable alternative is to use a mean function of the form e.g. $m(\mathbf{x}) = \mathbf{c}^{\mathrm{T}}\mathbf{x} + \mathbf{x}^{\mathrm{T}}\mathbf{Q}\mathbf{x} + b$ where $\mathbf{c}$ is a vector, $\mathbf{Q}$ a diagonal matrix and b a constant.

The elements of the covariance matrix $\mathbf{K}$ in equation (2.56) are constructed using the function $k(\mathbf{x}, \mathbf{x}')$. This function is called a kernel function or covariance function (Rasmussen and Williams, 2006, p. 13). It can be a parametric function, whose parameters are called hyperparameters (but not the same hyperparameters as the trading algorithm's) and they will be collectively denoted as $\boldsymbol{\theta}$. The kernel function

is central on the behavior of the GP (Rasmussen and Williams, 2006, p. 12), and is discussed in more detail in subsection 2.5.2.

Recall that the probability density function of multivariate normal distributed vector $\mathbf{f}$ given mean $\mathbf{m}$ and covariance matrix $\mathbf{K}$ is defined as below (Eaton, 1983, p. 120):

$$\mathbb{P}(\mathbf{f} \mid \mathbf{m}, \mathbf{K}) = \frac{1}{\sqrt{\det(2\pi\mathbf{K})}} \exp\left(-\frac{(\mathbf{f}-\mathbf{m})^{\mathrm{T}}\mathbf{K}^{-1}(\mathbf{f}-\mathbf{m})}{2}\right). \tag{2.56}$$

For the application of Bayesian optimization, the GP is used to approximate the distribution of the objective function f using given samples $y_i,\ i = 1, \ldots, n$ which are (possibly) noisy samples of previous evaluations of f (Rasmussen and Williams, 2006, pp. 15-16). The noise is assumed to be Gaussian and can therefore be written:

$$y_i = f(\mathbf{x}_i) + \epsilon_i, \quad \epsilon_i \sim \mathcal{N}(0, \sigma_n^2) \tag{2.57}$$

where ϵ_i are i.i.d. with variance $\sigma_n^2 \geq 0$. A vector of samples $\mathbf{y} = [y_1 \ \ldots \ y_n]^{\mathrm{T}}$ will then be distributed as:

$$\mathbf{y}|\mathbf{X}, \boldsymbol{\theta} \sim \mathcal{N}(\mathbf{m}, \mathbf{K} + \sigma_n^2\mathbf{I}) \tag{2.58}$$

where $\mathbf{I}$ is the unit matrix.

If $i = 1, \ldots, m$ new samples y_{*i} are taken at new points $\mathbf{x}_{*i}$ (the aggregated new inputs is denoted $\mathbf{X}_*$) and if the new samples are assumed to be noisy in the same way as in equation (2.57) then the joint distribution is (Rasmussen and Williams, 2006, p. 16):

$$\begin{bmatrix}\mathbf{y}\\ \mathbf{f}_*\end{bmatrix}\Bigg| \mathbf{X}, \mathbf{X}_*, \boldsymbol{\theta} \sim \mathcal{N}\left(\begin{bmatrix}\mathbf{m}\\ \mathbf{m}_*\end{bmatrix}, \begin{bmatrix}\mathbf{K} + \sigma_n^2\mathbf{I} & \mathbf{K}_*\\ \mathbf{K}_*^{\mathrm{T}} & \mathbf{K}_{**}\end{bmatrix}\right) \tag{2.59}$$

where $\mathbf{f}_*$, $\mathbf{m}$, $\mathbf{m}_*$ and $\mathbf{K}$ are defined similarly as in (2.54). The blocks $\mathbf{K}_*$ and $\mathbf{K}_{**}$ of the covariance matrix are defined as:

$$\mathbf{K}_* = \begin{bmatrix} k(\mathbf{x}_1, \mathbf{x}_{*1}) & \ldots & k(\mathbf{x}_1, \mathbf{x}_{*m}) \\ \vdots & \ddots & \vdots \\ k(\mathbf{x}_n, \mathbf{x}_{*1}) & \ldots & k(\mathbf{x}_n, \mathbf{x}_{*m}) \end{bmatrix} \in \mathbb{R}^{n \times m} \tag{2.60}$$

and:

$$\mathbf{K}_{**} = \begin{bmatrix} k(\mathbf{x}_{*1}, \mathbf{x}_{*1}) & \ldots & k(\mathbf{x}_{*1}, \mathbf{x}_{*m}) \\ \vdots & \ddots & \vdots \\ k(\mathbf{x}_{*m}, \mathbf{x}_{*1}) & \ldots & k(\mathbf{x}_{*m}, \mathbf{x}_{*m}) \end{bmatrix} \in \mathbb{R}^{m \times m}. \tag{2.61}$$

Using equations (2.59) and (2.57) it can be derived that the new samples $\mathbf{y}_*$ has the following marginal distribution conditioned on the already available samples $\mathbf{y}$ (Eaton, 1983, p. 116):

$$\mathbf{y}_* \mid \mathbf{x}_*, \mathbf{y}, \mathbf{X}, \boldsymbol{\theta} \sim \mathcal{N}(\bar{\mathbf{f}}_*, \mathrm{Cov}(\mathbf{f}_*) + \sigma_n^2\mathbf{I}) \tag{2.62}$$

where the conditional mean is defined as:

$$\bar{\mathbf{f}}_* = \mathbf{m}_* + \mathbf{K}_*^{\mathrm{T}}(\mathbf{K} + \sigma_n^2\mathbf{I})^{-1}\mathbf{y} \tag{2.63}$$

and conditional covariance $\mathrm{Cov}(\mathbf{f}_*)$ as:

$$\mathrm{Cov}(\mathbf{f}_*) = \mathbf{K}_{**} - \mathbf{K}_*^{\mathrm{T}}(\mathbf{K} + \sigma_n^2\mathbf{I})^{-1}\mathbf{K}_*. \tag{2.64}$$

Some similarities can be seen between equations (2.63) and (2.64) and to when updating the estimated latent variables in a Kalman filter (Jakobsson, 2015, p. 292).

A problem which could occur, if there exists no or too little measurement noise

$\sigma_n = 0$, is that two measurements close to each will have the same numerical value. This will imply the matrix $\boldsymbol{K}$ to be ill-conditioned and non-invertible. The most general solution to this problem is to add a "nugget", $\delta > 0$, to the measurements to achieve numerical stability by replacing $\boldsymbol{K}$ with $\boldsymbol{K}_\delta = \boldsymbol{K} + \delta \boldsymbol{I}$ (Ranjan, Haynes, and Karsten, 2011, p. 369). Effectively, this corresponds to adding noise to the measurements, i.e. increasing σ_n^2.

2.5.2 Kernel functions

The covariance function, $k(\mathbf{x}_1, \mathbf{x}_2)$, for the Gaussian process can be defined in several different ways, with one of them being the squared exponential kernel in one dimension:

$$k(\mathbf{x}_1, \mathbf{x}_2) = \sigma_f^2 \exp\Big(-\frac{1}{2l^2}(\mathbf{x}_1 - \mathbf{x}_2)^2 \Big), \tag{2.65}$$

where the free parameters σ_f and l are called hyperparameters (Rasmussen and Williams, 2006, pp. 19-20). Note that these are different hyperparameters than those of the trading algorithm. Varying the kernel's hyperparameters will achieve different properties of the Gaussian process.

As seen in figure 2.6 increasing standard prior deviation σ_f will yield a larger variance in general, namely the confidence intervals of the estimated predictions will be larger. The length-scale parameter l will determine the width of the kernel. Roughly speaking the length-scale parameter will determine for how the distance, $r = |x_1 - x_2|$, two measurements will be dependent of each other. A higher value of l will increase this distance. In addition, as the inverse of the length-scale parameter determines how relevant the input is, a value of l which is large enough will eventually make the input irrelevant (Rasmussen and Williams, 2006, pp. 106-107). For multidimensional inputs it is also possible to use anisotropic measurements, such as the Mahalanobis distance.

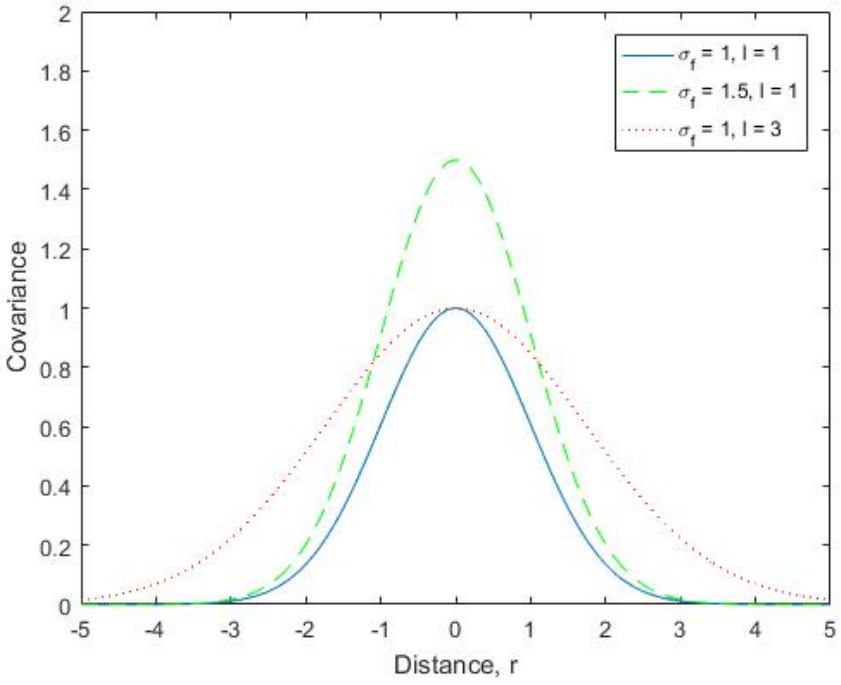

FIGURE 2.6: The squared exponential kernel, equation (2.65), with different values of the kernel's hyperparameters σ_f and l.

The squared exponential kernel, equation (2.65), has an infinite number of derivatives. This leads to that the covariance function will have mean square derivatives of all orders and the GP will therefore be very smooth. It can be argued that a such strong assumption about the smoothness is, in fact, unrealistic for a large number of real world applications. Usage of kernels from Matérn class can therefore be recommended instead. See table 2.3 for a variety of kernel functions. Still, the squared exponential kernel is probably most commonly used within the field (Rasmussen and Williams, 2006, p. 83).

TABLE 2.3: The expression of a few different kernels, with $r = x_1 - x_2$. The reader is referred to Rasmussen and Williams (2006, p. 94) for a more detailed table.

Covariance function	Expression
Constant	σ_0^2
Linear	$\sum_{d=1}^{D} \sigma_d^2 x_{1,d} x_{2,d}$
Polynomial	$(x_1 \cdot x_2 + \sigma_0^2)^p$
Squared Exponential	$\exp(-r^2 \,/\, 2l^2)$
Matérn	$\frac{1}{2^{\nu-1}-\Gamma(\nu)}(\frac{\sqrt{2\nu}}{l}r)^\nu K_\nu \left(\frac{\sqrt{2\nu}r}{l}r\right)$
Exponential	$\exp(-r \,/\, l)$
γ-exponential	$\exp(-(r \,/\, l)^\gamma)$
Rational quadratic	$(1 + r^2 \,/\, 2\alpha l^2)^{-\alpha}$
Neural network	$\sin^{-1}\left(\frac{2\tilde{x_1}^{\mathrm{T}} \sum \tilde{x_2}}{\sqrt{(1+2\tilde{x_1}^{\mathrm{T}} \sum \tilde{x_1})(1+2\tilde{x_2}^{\mathrm{T}} \sum \tilde{x_2})}}\right)$

2.5.3 Estimation of the Bayesian Hyperparameters

As shown in table 2.3 there are numerous kernels to choose from. What these kernels have in common is all of them have a set of hyperparameters, denoted $\mathbf{\Omega}$, which will change the properties of the GP. In order to optimize these parameters one can begin by measuring the different levels of inference.

At the bottom level there is the inference for the parameters, $\mathbf{w}$, of the black-box function originally in need for optimization (i.e. the trading hyperparameters). The posterior is written as:

$$\mathbb{P}[\mathbf{w} \mid \mathbf{y}, \mathbf{X}, \mathbf{\Omega}] = \frac{\mathbb{P}[\mathbf{y} \mid \mathbf{X}, \mathbf{w}] \cdot \mathbb{P}[\mathbf{w} \mid \mathbf{\Omega}]}{\mathbb{P}[\mathbf{y} \mid \mathbf{X}, \mathbf{\Omega}]}, \tag{2.66}$$

where $\mathbb{P}[\mathbf{w} \mid \mathbf{\Omega}]$ is the prior. The prior is chosen to reflect the knowledge about the parameters prior to seeing the data. If the knowledge about the parameters is vague, the prior is chosen to be broad in order to reflect this.

The posterior is then able to combine information about the parameters using both the prior and the data, through the likelihood. The denominator $\mathbb{P}[\mathbf{y} \mid \mathbf{X}, \mathbf{\Omega}]$, which is independent of the parameters, is called the marginal likelihood:

$$\mathbb{P}[\mathbf{y} \mid \mathbf{X}, \mathbf{\Omega}] = \int \mathbb{P}[\mathbf{y} \mid \mathbf{X}, \mathbf{w}] \cdot \mathbb{P}[\mathbf{w} \mid \mathbf{\Omega}] \cdot \mathrm{d}\mathbf{w}. \tag{2.67}$$

For the next level of inference the posterior for the kernel's hyperparameters, $\mathbf{\Omega}$, is expressed:

$$\mathbb{P}[\mathbf{\Omega} \mid \mathbf{y}, \mathbf{X}] = \frac{\mathbb{P}[\mathbf{y} \mid \mathbf{X}, \mathbf{\Omega}] \cdot \mathbb{P}[\mathbf{\Omega}]}{\mathbb{P}[\mathbf{y} \mid \mathbf{X}]}, \tag{2.68}$$

where the hyperprior $\mathbb{P}[\mathbf{\Omega}]$ is the prior for the hyperparameters. In practice the denominator of the posterior for the hyperparameters has been proven especially difficult to solve. As an approximation one may instead maximize equation (2.67) with respect to the kernel's hyperparameters $\mathbf{\Omega}$.

An approximation like this is named type II maximum likelihood (ML-II) and one should take great care since this type of approximation opens up for the possibility of over-fitting (Rasmussen and Williams, 2006, pp. 108-109). However, by doing this optimization instead, one can state the log-likelihood as:

$$\log \mathbb{P}[\mathbf{y} \mid \mathbf{X}, \mathbf{\Omega}] = -\frac{1}{2}\mathbf{y}^{\mathrm{T}}\mathbf{K}_y^{-1}\mathbf{y} - \frac{1}{2}|\mathbf{K}_y| - \frac{n}{2}\log(2\pi), \tag{2.69}$$

with $\mathbf{K}_y = \mathbf{K}_f + \sigma_n^2\mathbf{I}$ being the noisy covariance matrix of observations, $\mathbf{y}$.

The computational cost of calculating the derivative with respect to the hyperparameters, $\frac{\partial}{\partial \Omega_i} \log \mathbb{P}[\mathbf{y} \mid \mathbf{X}, \mathbf{\Omega}]$, of the marginal LL is relatively small (Rasmussen and Williams, 2006, pp. 114-115). A gradient-based numerical method for optimization of the hyperparameters is therefore recommended.

2.5.4 Acquisition Functions

In the search for the maximum of the objective function, f, the next step will be to use a, so called, acquisition function. The acquisition function is designed to guide the optimization algorithm to the next evaluation of the objective function using the estimated posterior. The next evaluation point of the posterior will therefore be chosen as the maximum of the acquisition function, figure 2.7. The maximum of the acquisition function should correspond to, potentially, high values of the objective function. This may either be a large prediction or a great variance of the posterior (Brochu, Cora, and de Freitas, 2010, p. 11). This is known as the exploration-exploitation trade-off.

In some literature the trade-off may also be refereed to as a multi-armed bandit dilemma. The idea behind this scenario is to say that one stands before two slot machines, also refereed to as a one-armed bandit, one will represent the exploration, where the variance is high, and the other to exploit what is already known, where the objective function's value is predicted to be great. The choice is then if one should pull the lever which explores, exploits or any mixture of the two, creating an infinite-armed bandit. Naturally one wishes to pull the lever with the highest possibility of gaining a reward. The trade-off will prove to be an important design choice in order to obtain a fast algorithm (Bubeck and Cesa-Bianchi, 2012, p. 9).

Another important property of the acquisition function is that it has to be easy to evaluate and maximize as this will determine the next point of the objective function to evaluate.

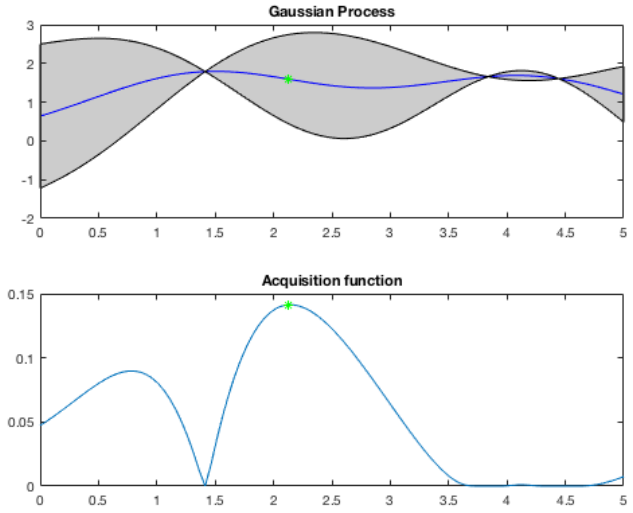

FIGURE 2.7: A GP fitted to a few data points with corresponding acquisition function (expected improvement) and the next evaluation point.

As in the case with kernels, there are a numerous different acquisition functions to choose from. In this section a few of the existing acquisition functions will be presented.

Probability of Improvement

One of the first acquisition functions was suggested by Kushner (1964), who though of maximizing the probability of improvement (PI) from the maximum observed value y^+:

$$\text{PI}(\mathbf{x}) = \mathbb{P}[f(\mathbf{x}) > y^+] = \Phi\left(\frac{\mu(\mathbf{x}) - y^+}{\sigma(\mathbf{x})}\right). \tag{2.70}$$

The problem with this formulation is that there is often a large probability of new function evaluations being infinitesimally greater than the currently observed largest value. The solution is therefore to add a exploration parameter $\xi \geq 0$:

$$\text{PI}(\mathbf{x}) = \mathbb{P}[f(\mathbf{x}) > y^+ + \xi] = \Phi\left(\frac{\mu(\mathbf{x}) - y^+ - \xi}{\sigma(\mathbf{x})}\right). \tag{2.71}$$

The exact value of ξ is left to the user to decide upon but it is suggested to start with a high value in the beginning of the optimization to drive exploration and then let it go towards zero by the end of the optimization scheme (Kushner, 1964).

Expected Improvement

A, perhaps, better approach than PI is not only to evaluate the probability of improvement but the magnitude of the improvement as well. An analytical expression for this acquisition function, called the expected improvement (EI), is defined as (Jones, Schonlau, and Welch, 1998, p. 471):

$$\text{EI}(\mathbf{x}) = \big(\mu(\mathbf{x}) - y^+\big)\Phi\left(\frac{\mu(\mathbf{x}) - y^+}{\sigma(\mathbf{x})}\right) + \sigma(\mathbf{x})\phi\left(\frac{\mu(\mathbf{x}) - y^+}{\sigma(\mathbf{x})}\right). \tag{2.72}$$

where $\phi(\cdot)$ and $\Phi(\cdot)$ denote the PDF and CDF of the standard normal distribution. An important property of this acquisition function is that it, unlike PI, does not necessary have to be myopic and a multi-step Bayesian optimizer is therefore a possibility (Brochu, Cora, and de Freitas, 2010, p. 14).

The exploration-exploitation trade-off can, in the same manner as with PI, be achieved by adding a parameter $\xi \geq 0$ (Lizotte, 2008, p. 49):

$$\mathrm{EI}(\mathbf{x}) = \big(\mu(\mathbf{x}) - y^+ - \xi\big)\Phi\left(\frac{\mu(\mathbf{x}) - y^+ - \xi}{\sigma(\mathbf{x})}\right) + \sigma(\mathbf{x})\phi\left(\frac{\mu(\mathbf{x}) - y^+ - \xi}{\sigma(\mathbf{x})}\right). \quad (2.73)$$

It is suggested by Lizotte (2008) that a value of $\xi = 0.01$ should work in almost every case. In addition Lizotte claims there is nothing to be gained from changing the value of ξ at each iteration of the algorithm. For instance to start with a large value of ξ which then is reduced towards the end of the optimization scheme.

2.5.5 Optimization Algorithm

Just as with the trading algorithm, the algorithm for BO is also built from a few different components. Namely a Gaussian processes working together with an acquisition function. The way the optimization is done is by evaluating the trading algorithm for different sets of hyperparameters and the goal is to obtain the set yielding the greatest average return of the in-sample data set.

Recall that the trading hyperparameters are mutually dependent, as discussed (Nystrup et al., 2017, p. 15). BO is therefore desirable as the number of function evaluations, i.e. the number of times the trading algorithm 2.2 has to be executed, will be kept to a minimum. For further clarification, pseudo-code for the full BO scheme can be seen in algorithm 2.3. Evaluation of the objective function (i.e. executing the trading algorithm) takes by far the most time in the optimization loop. Multiple samples can thus be gathered in parallel, thereby speeding up the time taken to optimize the trading hyperparameters.

Algorithm 2.3 Outline of the Bayesian optimization algorithm.

1: Given a black box objective function f.
2: Given a choice of a kernel function from table 2.3 and an acquisition function.
3: Evaluate the function with arbitrary parameters to obtain a few datapoints and store samples in set $\mathcal{D}$.
4: **while** Stopping criteria not met **do**
5: Fit GP to the data points by optimizing its hyperparameters
6: Find next evaluation point using the acquisition function
7: Evaluate f at the point determined by the acquisition function
8: Add the sample to set of known samples $\mathcal{D}$.
9: **endwhile**

Chapter 3

Method

3.1 The data

The data set consists of the daily closing prices of the S&P 500 stock index from the beginning of 1928 until the end of 2017. The index, S&P 500, was selected since it is the data set which is studied in the articles by Nystrup, Madsen, and Lindström. Data is collected for a long period of time since it often takes some amount of data in order to get good estimates of the parameters in the model. This implies that the whole data set will not be used for trading. The period of 2 Jan 1928 to 29 Dec 2005 will be the convergence set. The actual trading will start at 2 Jan 1996 and stop at the end of the data set, seen in figure 3.1. The period of 2 Jan 1996 to 29 Dec 2006 will serve as an in-sample data set. This set will be used to train the model. The model will then be tested on the the remainder of the period, 1 Jan 2007 to 29 Dec 2017, which is the out-of-sample data set. This means that it is merely these subsets of the data which will be used for the actual measure of performance. Note that both the in-sample and out-of-sample sets span over 11 year periods. The data was collected from the Bloomberg terminal which is available for members of Linc Lund University Finance Society.

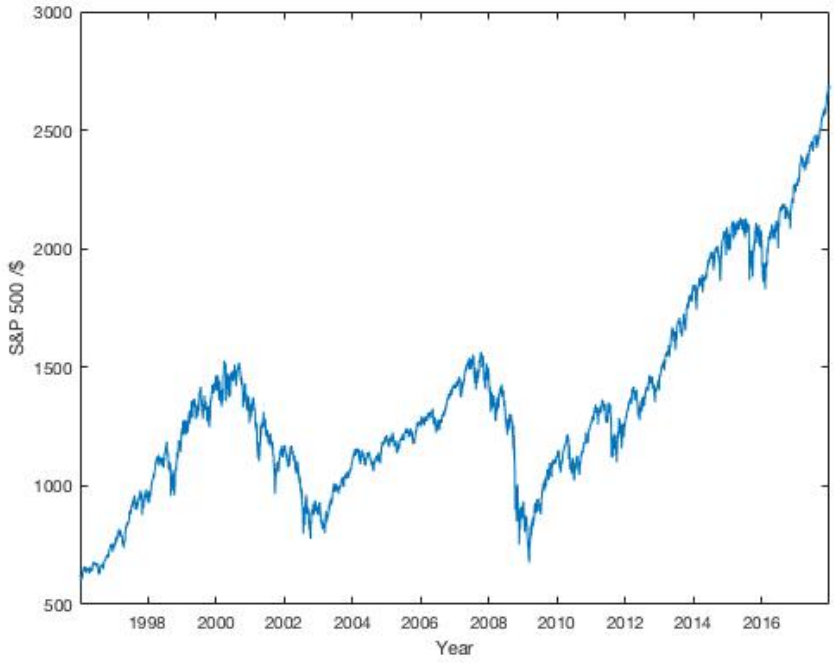

FIGURE 3.1: S&P 500-index during the trading period from 2 Jan 1996 to 29 Dec 2017.

The parameters for the model will be estimated based on the daily log-returns of the data:

$$r_t = \log\left(\frac{y_t}{y_{t-1}}\right)$$

where y_t is the price of S&P 500. They can also be used to estimate the ordinary daily returns:

$$q_t = \frac{y_t - y_{t-1}}{y_{t-1}}.$$

The support of the log-returns is $r_t \in \mathbb{R}$ and are often assumed to be normally distributed (Nystrup, Madsen, and Lindström, 2015, p. 1534). The ordinary returns are supported over $q_t \in (-1, \infty)$ and will be log-normal-distributed, if $r_t \sim \mathcal{N}$. How well the distributions fit the respective returns can be seen in figure 3.2. In figure 3.2 it can easily be seen that these assumptions about the distributions of the returns are not far-fetched.

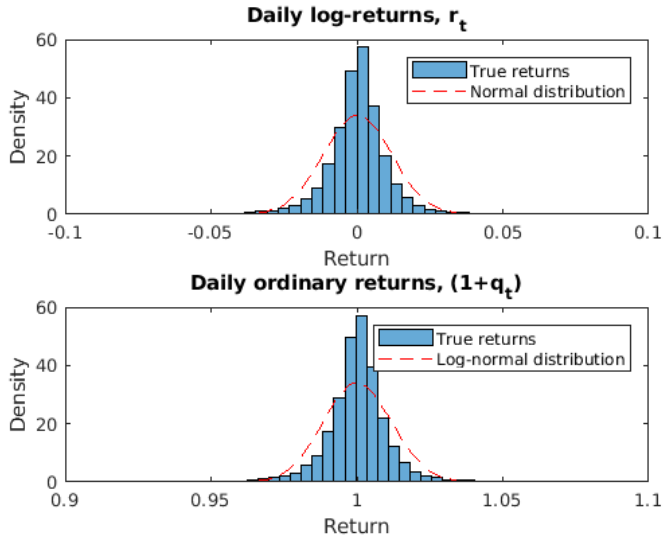

FIGURE 3.2: Histogram of the daily log- and ordinary returns from the S&P 500 stock index compared to a normal and log-normal distribution respectively.

This difference in distribution and support will prove to be more or less desired in different situations and both of them will be used. One type of return can easily be transformed to the other at the end of a calculation:

$$(1 + q_t) = \mathrm{e}^{r_t}. \tag{3.1}$$

As seen in figure 3.2 the approximation does not yield the perfect fit to the true market data from the S&P 500 stock index. The real market data have heavier tails than the normal distribution and this may be the reason for the bad fit of the outliers in the data using this distribution. In this thesis the normal distribution in one of the market states will be replaced with Student's t-distribution which is assumed to better approximate the outliers, i.e. heavy tails, in the data. The true market returns compared with the approximations using a normal and a Student's t-distribution can be seen in figure 3.3. In this example, Student's t-distribution fits the real market data to a greater extent than the normal distribution. It should be reiterated that heavy tails is an observed stylized fact in financial time series (Lindström, Madsen, and Nielsen, 2015, p. 10).

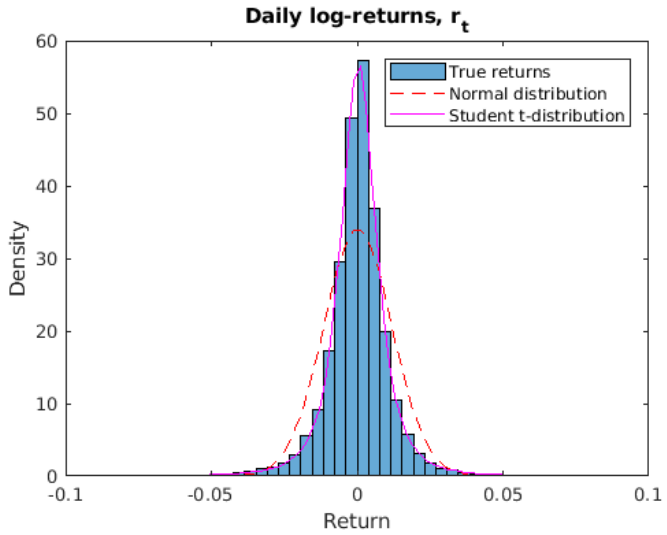

FIGURE 3.3: The daily log-returns from the S&P 500 stock index compared to a normal and Student's t-distribution.

3.2 Limitations of the Study

The study focuses on the mathematical point of view on the algorithm. The S&P 500 stock index is chosen for comparison as it is used in other studies. Due to the limitations of this study trading on other assets will not be covered in this thesis. The report does not include a discussion on which market conditions, or anomalies, that will impose to make this investment strategy successful. For further investigations it would be reasonable to run the algorithm on other assets as well. One would perhaps then be able to make some conclusions on which conditions the asset need to fulfill in order to make the strategy successful.

The algorithm is only tested on daily data. A possibility could be to run the algorithm on data which is sampled more frequently, for instance hourly, as well. Daily data was chosen for comparison to similar studies.

The portfolio consists of only one risky asset and the bank account. One may think that a portfolio consisting of multiple assets would be able to perform even better, c.f. Nystrup et al. (2017) and Nystrup et al. (2018). That would however increase the complexity of implementing the algorithm and is therefore disregarded. The scope of this thesis is therefore focused on trading the one asset of S&P 500.

The portfolio is optimized to deliver the strongest (largest) return possible, within the trading constraints $w_{\min}$ and $w_{\max}$. The risk of the portfolio is therefore not a priority for measurements of the performance. The tuning of the hyperparameters will be carried out with regards only to the portfolio return. The risk of the portfolio will then afterwards be displayed and discussed, but no consideration of the risk will be considered as the portfolio's hyperparameters are tuned. The design choice could be motived, for instance, with a parable to hedge funds where the return of the portfolio is considered more important than the risk.

3.3 Procedure

All computations were done in MATLAB. The portfolio in the first stage is long-only, i.e. there is no possibility to go short on the asset, and there is no use of leverage. Mathematically this is done by letting $(w_{\min}, w_{\max}) = (0, 1)$. The transactions costs are, as in the article by Nystrup et al. (2017, p. 15), assumed to be 10 basis points, $\kappa = 0.001$. Movements in the price of the risky asset, S&P 500, due to trading by the algorithm are ignored as the asset is considered to be liquid enough for this kind of assumption to be made. The portfolio is optimized to give the strongest possible return. Manual tuning of the hyperparameters will therefore not include any risk aversion as this may reduce the expected return of the portfolio. It should be noted that there are no trading costs included in the trading of the risk free asset as they are only related to trading with the S&P 500 stock index. A position in the risk-free asset will yield a risk-free return which in this thesis is assumed to be $r_{\mathrm{f}} = 0$. In addition, as discussed before, any holding costs associated with holding the risky asset are omitted.

As the goal of the portfolio is to yield a strong return, the optimization of the hyperparameters are optimized by measuring performance as the excess return. Although the performance of the whole portfolio, presented in tables in chapter 4, is measured by also including excess risk, Sharpe ratio, maximum drawdown, Calmar ratio and annual turnover calculated as in subsection 2.4.1. Which parameters that were designated for optimization are listed in table 3.1, where their respective allowed interval is listed as well. The fixed hyperparameters are also presented in table 3.1. The fixed values are assumed to be used if nothing else is stated. All "samples" from the BO, i.e. evaluations for specific hyperparameter configurations, were saved in order to find an efficient frontier.

The portfolio is later extended and allowed to go short in the risky asset as well, but without leverage for long positions. When the algorithm short sells the risky asset there are no holding cost related to the short position in the model. This implies that $(w_{\min}, w_{\max}) = (-1, 1)$. The algorithm was also tested with data simulated from a HMM as described in equation (2.2) using some realistic constant HMM parameters.

TABLE 3.1: Table regarding which hyperparameters are to be optimized or fixed. See table 2.2 for description of the hyperparameters.

Parameter	Fixed	Value
N_{eff}	No	$\in \{150, \ldots, 1200\}$
L	No	$= 4 \cdot N_{\mathrm{eff}}$
L_0	No	$= 5 \cdot N_{\mathrm{eff}}$
A	Yes	$= 1$
a	Yes	$= -1$
b	Yes	$= 1$
K	No	$\in \{5, \ldots, 100\}$
γ^{risk}	No	$\in [0, 5]$
$\gamma_1^{\mathrm{trade}}$	No	$\in [0, 10]$
$\gamma_2^{\mathrm{trade}}$	No	$\in [0, 0.5]$
$w_{\min}$	Yes	$= 0$
$w_{\max}$	Yes	$= 1$
κ	Yes	$= 0.001$

Chapter 4

Results

4.1 The Trading Algorithm

In order to get started, it will be reasonable to illustrate the results from the different parts of algorithm 2.1 in order for the reader get an understanding of the results as a whole. For the adaptive HMM parameter estimation, Nystrup et al. (2017) suggests an effective memory length $N_{\text{eff}} = 130$ days. In turn, for the given data set, this required the convergence parameter to be tuned down to $A = 0.5$ for numerical stability. The estimation of the parameters, figure 4.1, for the HMM is run for the whole data set, from 1928 to 2017, with an initialization period of $5 \cdot N_{\text{eff}} = 650$ days.

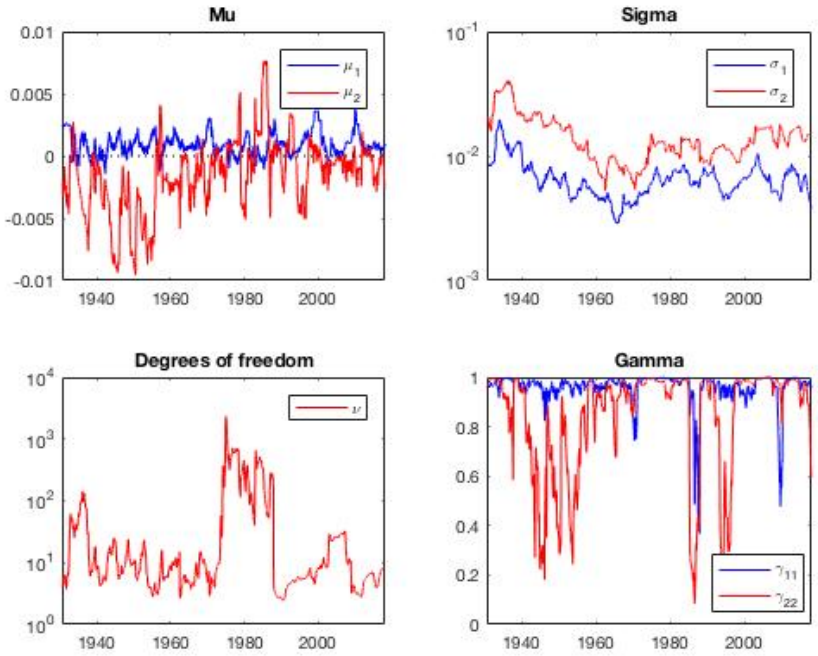

FIGURE 4.1: Estimated parameters in the HMM using $N_{\text{eff}} = 130$ days and $A = 0.5$.

Given the parameters in the HMM, one will then continue the algorithm to make predictions of the future return and variance. The result for one day in the future at any given time point during the trading period, from 1996 to 2017, can be seen in figure 4.2.

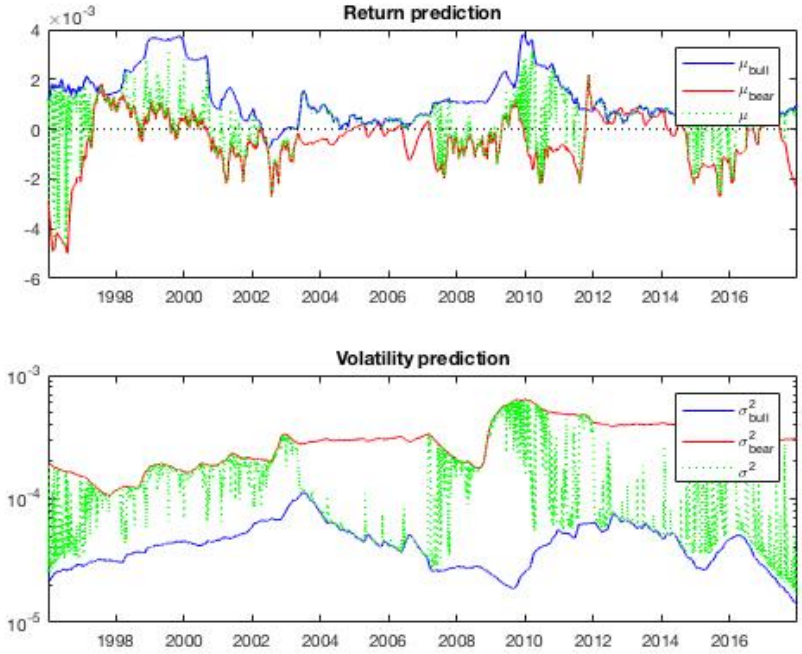

FIGURE 4.2: Predictions of the return and risk in the two states as well as the entire market during the trading period.

With predictions in place one will be able to move on to the actual MPC trading algorithm. The future time horizon was chosen to be $K = 100$ days and the other hyperparameters were set to be $\gamma^{\text{risk}} = 0$, $\gamma_1^{\text{trade}} = 10$ and $\gamma_2^{\text{trade}} = 0$. With these settings the portfolio is set to deliver a maximum return, without any respect to the risk. The trading is done with respect to constraints $(w_{\min}, w_{\max}) = (0, 1)$. The portfolio performance can be studied through table 4.1 and figure 4.3. In figure 4.3 the reader will see the portfolio performance compared to the traded asset together with the portfolio weight, w, and the HMM value α which can be seen as an estimation of the market state, bull or bear (c.f. equation (2.25)).

TABLE 4.1: The portfolio's KPIs compared to KPIs of the traded S&P 500 index.

KPI	Index	Portfolio
Average return:	0.0670	0.0898
Standard deviation:	0.1894	0.1326
Maximum drawdown:	0.5678	0.2118
Annual turnover:	N/A	1.1400
Sharpe ratio:	0.3539	0.6770
Calmar ratio:	0.1181	0.4240

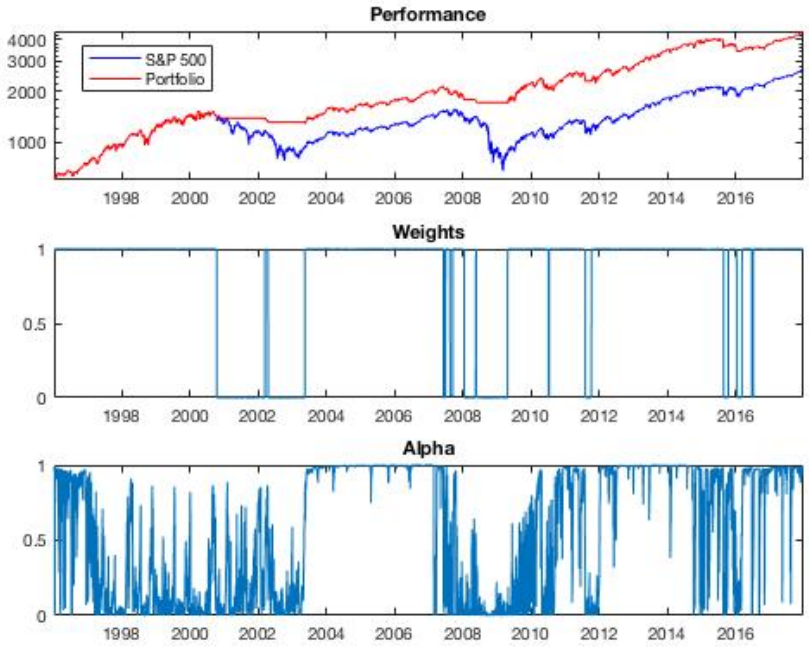

FIGURE 4.3: The portfolio performance compared to the traded asset, together with the portfolio weight and the HMM parameter $\alpha(1)$ during the trading period from 1996 to 2017.

4.2 Bayesian Optimization

As mentioned, in the real back-test of the algorithm the data is divided into two subsets, in- and out-of-sample. The in-sample data set goes from 2 Jan 1996 to 29 Dec 2006 and is used to determine the hyperparameters of the MPC trading algorithm. The performance of the algorithm, in-sample, is measured by the average return the portfolio will yield during the in-sample trading period. In other words, the optimization was done with respect to maximizing the average return. In the remainder of the period, 1 Jan 2007 to 29 Dec 2017, the hyperparameters are fixed to their respective optimal values according to the in-sample data set. Some of the tested sets of hyperparameters with corresponding in-sample and out-of sample performance measures can be seen in table 4.2. The evaluated sets of hyperparameters have been chosen with increasing average return and standard deviation, thereby maintaining a fairly constant Sharpe ratio, for the in sample data set. It was expected that the Sharpe ratio would remain constant for the out-of-sample data set as well. The excess return in relation the excess risk of the portfolio for all the tested parameter values, as chosen by the BO algorithm, can be seen in figure 4.4. The solid line in the figure represent the same set of hyperparameters, in- and out-of sample, displayed in table 4.2.

TABLE 4.2: Optimal hyperparameters of the trading algorithm and KPIs. Abbreviations indicates average return, standard deviation, maximum drawdown, Sharpe ratio and Calmar ratio. Subscript "i" indicates the KPI is calculated using in-sample trading and "o" out-of-sample trading. The columns are sorted from left to right in ascending order with regards to average return in-sample.

N_{eff}	179	326	340	383	285	435	213	213	213	171
K	6	57	54	97	62	67	58	31	80	56
γ^{risk}	0.345	4.955	4.815	4.982	2.082	1.854	2.242	1.565	0.080	0.058
γ_1^{trade}	9.382	6.636	1.100	8.920	5.232	5.415	1.007	5.003	5.717	1.718
γ_2^{trade}	0.328	0.265	0.389	0.105	0.273	0.341	0.486	0.118	0.163	0.015
AR_i	0.008	0.034	0.039	0.051	0.069	0.076	0.091	0.096	0.102	0.107
SD_i	0.029	0.045	0.055	0.068	0.095	0.108	0.119	0.123	0.129	0.134
MDD_i	0.111	0.076	0.074	0.094	0.166	0.173	0.193	0.193	0.193	0.193
SR_i	0.280	0.758	0.707	0.750	0.729	0.707	0.767	0.779	0.793	0.795
CR_i	0.074	0.449	0.529	0.546	0.415	0.441	0.473	0.496	0.529	0.553
AR_o	0.048	0.012	0.013	0.009	0.030	0.028	0.052	0.053	0.045	0.067
SD_o	0.079	0.046	0.040	0.038	0.106	0.075	0.128	0.145	0.168	0.121
MDD_o	0.122	0.148	0.161	0.113	0.279	0.156	0.286	0.341	0.444	0.298
SR_o	0.602	0.263	0.316	0.241	0.288	0.372	0.407	0.367	0.265	0.548
CR_o	0.391	0.081	0.078	0.081	0.109	0.179	0.182	0.155	0.100	0.223

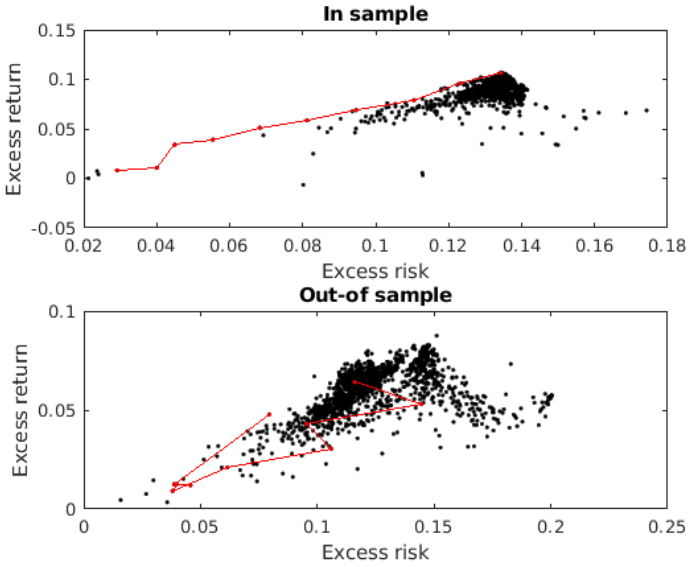

FIGURE 4.4: Comparison of the efficient frontier in-sample taken out-of-sample.

4.3 Long-Short Portfolio

Another interesting alternative to the long only portfolio is to give the algorithm permission to go short in the asset. If the predictions are correct, this will in theory yield the algorithm the ability to earn strong returns in both bear market as well as the bull market. This is achieved by changing the weighting constraints to $(w_{\min}, w_{\max}) = (-1, 1)$, in order to allow for short selling the risky asset. The result of the algorithm with and without the permission to go short in the traded asset, with the same

hyperparameter values as before: $\gamma^{\text{risk}} = 0$, $\gamma_1^{\text{trade}} = 10$ and $\gamma_2^{\text{trade}} = 0$, can be seen in table 4.3 and figure 4.5.

TABLE 4.3: The portfolio performance compared to the traded index without risk aversion.

Performance measure	Index	Long-only Portfolio	Long-short Portfolio
Annual return:	0.0670	0.0898	0.0927
Annual risk:	0.1894	0.1326	0.1891
Maximum drawdown:	0.5678	0.2118	0.3263
Annual turnover:	N/A	1.1400	3.4770
Sharpe ratio:	0.3539	0.6770	0.4901
Calmar ratio:	0.1181	0.4240	0.2840

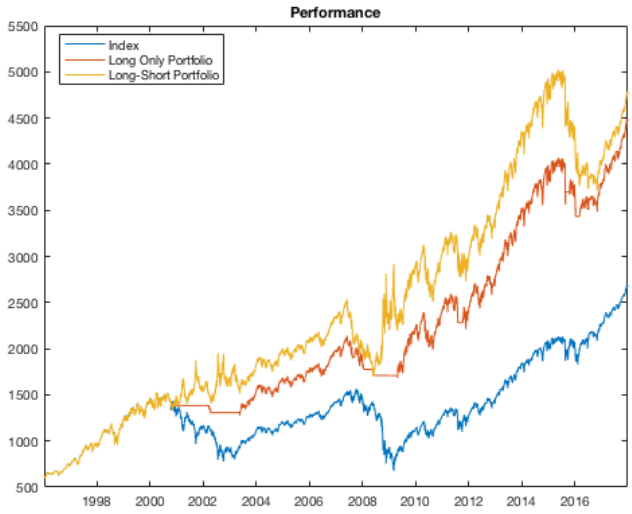

FIGURE 4.5: The portfolio performance, with and without the ability to go short, compared to the traded asset. No risk aversion was used. Note that the relatively large drawdown both with and without short selling around the 2016 mark is due to excessive trading (c.f. figure 4.3).

Chapter 5

Analysis

5.1 The Trading Algorithm

From the results of the portfolio performance, table 4.1 and figure 4.3, one can see that the portfolio succeeds to yield the investor a higher return at a lower risk than just investing in the stock index. In turn this will yield the investor an increased Sharpe-ratio which can be seen as successful. This is achieved with an annual turnover of AT = 1.14 which is not too large for an institutional investor to cope with.

Viewing the parameter estimates of the HMM one can see that the parameter $\alpha(1)$, figure 4.3, is fairly low between 2008-2010. This indicates that the algorithm estimates the market to be in its bear state during this period. Since we know the world to be in a financial crisis during this period of time this seems to be a good estimate. Viewing the portfolio weights, w, in the same figure one can see that the algorithm trades a lot before the actual crisis, at the end of 2007. The kind of behavior is, of course, not desired as it increases the annual turnover and decreases the return due to trading costs. Since the actual crisis hit the market in 2008 the algorithm is able to outperform the market, making up for the unnecessary trading in 2007. A drawdown occurs in 2016 when the estimates becomes unstable once again and the algorithm starts to trade back and forth. This time there is no decline in the market and the reallocation merely yield missed market returns and unnecessary transaction costs.

Among the parameter estimates from the HMM, see figure 4.1, the degrees of freedom, ν, is perhaps the one that varies the most. During a period of time, approximately between 1985-1995, the degrees of freedom rises up to somewhere between $\nu = 100$ and $\nu = 1000$. There is nothing that says this estimate is wrong and the algorithm will not crash because of this. The degrees of freedom will however determine the tails of the Student's t-distribution, which was chosen to make the model a better fit of outliers in the data. When the degrees of freedom increases the Student's t-distribution will have lighter tails and in the limit $\nu \to \infty$ the Student's t-distribution converges to the normal distribution (Johnson, Kotz, and Balakrishnan, 1995, p. 367). As the estimate of the degrees of freedom increases, as in this scenario, the algorithm will lose its robustness for outliers which is the reason for using Student's t-distribution instead of the normal distribution. Of course, this property is not what was intended but may be reasonable if the data was free of extreme returns during the period time. The algorithm then adjusts itself back to more a reasonable value of $\nu \approx 10$ which may indicate that the data then consists of more exceptional returns and the algorithm adjusts itself to the current market conditions. In this case it would though have been better to use a normal-distribution in both states as in Nystrup, Madsen, and Lindström (2016) since it contains one less parameter and thereby reduces the risk of over-fitting the model.

The estimates of the transition probabilities of γ_{11} and γ_{22}, see figure 4.1, can be seen to vary rapidly. It can also be seen that γ_{22} even reaches a value as low as $\gamma_{22} \approx 0.1$ in 1987. It would be intuitive that this jump was caused by the *Black Monday* crash of 19 Oct 1987 and the days thereafter. This rapid variation in the transition probabilities could indicate that the the algorithm's adaptive estimation method is not as stable as one would wish for. Very low values for the transition probabilities implies very short sojourn times. This means that the high volatility Student's t-distributed state translates to an outlier state, rather than to a market regime, during periods with a low value of γ_{22}. This is not desirable, since this state in the HMM is designed to model the bear market regime and is not intended to be an outlier state. The algorithm will therefore not have the same properties, during this brief period, as intended by the developers.

As a lower effective memory length N_{eff} increases the speed of convergence in the HMM, this constant should be fairly low to make the algorithm adapt to the most recent market conditions. A value of N_{eff} which is too low will however make the parameter estimates, see figure 4.1, unstable in the sense that the estimations take huge leaps from reasonable values to unreasonable. The case when the HMM becomes unstable can happen through multiple scenarios. The first scenario is when the magnitude of an estimated mean of a distribution becomes too large. The returns given from the data are then unlikely to fit into that state, making the transition probabilities to that state go down. The state will then go from a market regime to an outlier state as the probability drops. The probability may in some cases even turn to zero. The state will from then on be unable to recover and one may say that the state has collapsed. The same scenario can be achieved by estimating the variance of any distribution too low. The distribution will then be too narrow for any returns to fit the distribution and the state will collapse. All the scenarios will lead to one, or in some cases both, states collapsing which leads to a malfunction in the HMM, specifically in the LH algorithm. The algorithm will then asymptotically find the correct parameters again, but without any guarantees for the convergence speed (i.e. it will take a very long time). If only one state collapses, the algorithm will still be able to make predictions with the one state that is still "alive" (i.e. not collapsed). The predictions will however not nearly be as accurate with only one state alive as with both of them fully functioning. Great care will therefore have to be considered before changing the effective memory length, N_{eff}, of the HMM. The user will have to approve that estimates seems reasonable, by viewing a graph similar to figure 2.1 and then adjust the tuning constant, A, accordingly to make the model stable. The reader is referred to section 7.1 in the discussion for some suggestions on how to improve the stability in the parameter estimation.

5.2 Bayesian Optimization

Using Bayesian optimization one may be able to find the hyperparameters that are the most optimal for the in-sample data in terms of the return, as was done in this report. Although it could also have been done to maximize Sharpe ratio. Looking at the portfolios' KPIs in table 4.2, as the risk (measured as standard deviation) increases, from left to right, a rational investor would like a higher return as compensation for the increasing risk. The values in this case should correspond to the CML and the Sharpe ratio should therefore be constant. For the in-sample data set one can see that the Sharpe ratio remains fairly constant, except for the first value which can be seen as an outlier. This is also seen in figure 4.4 as the values along the efficient

frontier form a straight line, as does the CML. When the best parameter estimates are used for the out-of-sample data set, the Sharpe ratio (table 4.2) is no longer even close to constant implying the model has been over-fitted to the in-sample data set. The hypothesis is perhaps even more clearly illustrated in figure 4.4 as the efficient frontier for the in-sample data set is no longer efficient out-of-sample. The points are now close to randomly placed in the plot and there are lots of other hyperparameters that would have yielded a much better result in terms of both the Sharpe ratio as well as sheer excess return.

The usage of Bayesian optimization was an idea which seemed to be good in theory but proved to be merely moderate in reality. The authors believe that one of the reasons for this is fundamental to the data, that the market conditions changes over time. As the market conditions changes between the in and the out-of-sample data sets, the chosen hyperparameters will not yield the same properties of the algorithm in the two sets. The performance of the algorithm in the in sample data set can therefore not be seen as a warranty for the performance moving out-of-sample. One can conclude that this change of market conditions seems to not have been captured by the time-varying HMM parameters.

There is, however, also a possibility that the Bayesian optimization could have been done in a smarter, less crude, approach for better performance. The term "crude" is used here due to the fact that the BO algorithm was simply performed on a five-dimensional rectangular hyperparameter space. Five dimensions could have been too much to give the optimization scheme a fair chance, since the GP's need for samples grows exponentially with added dimensions. This is a well known fact in the optimization community and is referred to as the curse of dimensionality (Spall, 2003, p. 14). Although the results from using BO might not have been as good as the authors may have hoped for, the authors are still optimistic regarding this subject. BO is a method used in many applications as an intelligent approach to finding good hyperparameter values. This which could prove to be usable in the context of trading algorithms in the future. See for example Shahriari et al. (2016) for an overview of some applications were BO is used. Further investigations on this subject is therefore encouraged.

5.3 Long-Short Portfolio

The results from the long-short portfolio compared to the long only portfolio can be seen in table 4.3 and in figure 4.5. Comparing the two portfolios it is not hard to see that the long-only portfolio at most points in time performs close to the long-short portfolio. The risk of the long-short portfolio is higher than the long-only portfolio as it continues to take positions in the risky asset during the highly volatile bear market, whilst the long-only portfolio places most of its capital in the risk free asset (with no volatility). A rational, and risk avert, investor would want to be compensated for the greater risk with a higher return, maintaining an unchanged Sharpe-ratio. As the Sharpe ratio for the long-short portfolio is lower than for the long-only portfolio, see table 4.3, a rational investor is then likely to choose the long-only portfolio over the long-short portfolio. It should be reiterated that neither the long-only nor the long-short portfolio discussed here utilized the risk aversion trading (i.e. $\gamma^{\text{risk}} = 0$).

The reasons for the poor performance of the long-short portfolio may be a few. Most intuitively one may of course blame bad market predictions of the HMM. A prediction of negative returns in the future market which in reality turns out to be positive is not that bad for the long-only portfolio as there is no loss in this trade, but

no return either. The long-short portfolio will loose as much as the market goes up, which is a failure of double magnitude as for the long-only portfolio. For this reason it will be of utter importance to make reliable predictions of future market conditions for the long-short portfolio to make successful investments. This can be seen in figure 4.3 as the long-short portfolio seems to lose most of its value compared to both the index and the long-only portfolio during the time of 2007 and 2016. This is also the time where the long-only portfolio does a lot of unnecessary trades, as previously discussed. This can be seen as a clear example of when a wrong prediction will imply twice the failure for the long-short portfolio as it has to pay both the transaction costs as well the loss due the market going in a direction which was not predicted.

The final solution may however not be as easy as to blame the portfolio on poor market predictions. If a bear market is characterized with a higher variance than the bull market, this will imply greater returns, both positive as well as negative, than the bull market. If one uses this characterization of the bear markets instead of just negative market returns one can gain a better understanding of the algorithm's failures. The bear market may then often produce a huge downfall in the market, but then a rather fast recoil up to practically the same level as before. The long-short portfolio will then gain a positive return on the downfall but loose the returns on the recoil ending up with a total return over the period close to zero, just as the long-only portfolio will.

Chapter 6

Conclusion

The algorithm clearly has the ability to yield the investor a greater return at a lower risk than a static investment in the market portfolio. The investor should however be careful when selecting the model hyperparameters as there is no easy way to select the best configuration. The hyperparameters are additionally mutually dependent which argues for even more care when selecting the optimal set of hyperparameters (Nystrup et al., 2017, p. 15). The use of Bayesian optimization was tested to derive an optimal set of hyperparameters for the in-sample data. However, the optimization scheme seemed to lead to over-fitting the algorithm's hyperparameters and the performance was lost when used out-of-sample. The authors has the belief that Bayesian optimization has potential in this area, but further research has to be done. In the meantime, the best way to optimize hyperparameters, in-sample, may therefore be to tune them manually as in Nystrup et al. (2017, pp. 17-18).

The algorithm can with ease be configured to go short in the asset as well. This will however increase the risk of the portfolio without significantly increasing the return. The suggestion will therefore be to use the long only portfolio as it reduces the risk of the investment while still producing a strong return.

The goal of thesis was to make the algorithm by Nystrup more robust to outliers in the data by replacing the normal distribution with Student's t-distribution which has longer tails and may therefore yield a better fit to the outliers. The model is probably a better fit to the outliers than its successor, but it has the ability to lose its own robustness in terms of stability, due to numerical issues with the adaptive estimation. The estimation of the degrees of freedom, ν, in the Student's t-distribution may sometimes grow large making this distribution similar to the normal distribution. If this happens, the model will yield the same results as the one already examined in Nystrup (2017). The reason for this may be that the data contains few outliers and the normal distribution may therefore be the best fit of the data during these periods. This improved model should therefore be better or, in the worst case scenario, just as good as its predecessor.

Chapter 7

Discussion

The performance of the portfolio is measured on past index data. The future performance of the portfolio can therefore be questioned. The conditions in the market changes over time, which could be a reason for why the Bayesian optimization did not work as expected. As the market conditions changes so does the performance of the algorithm. Usage of the algorithm will therefore require constant monitoring of the performance of the portfolio. If the performance for some reason drops the algorithm should be stopped and reconfigured. The market conditions may even change to the point where the model will become obsolete. This is not discussed to its full extent in this thesis, but the fact is nevertheless true, the algorithm is trading on the anomalies in the market. The algorithm uses past prices in order to predict future movements in the market. If the so called efficient market hypothesis was true then the price set by the market would have been correct given the information available, at all times. This would cause the algorithm to not be able to exploit any discrepancies and would result in that the algorithm would not work. There is a possibility, that sometime in the future these anomalies will not exist to same extent in the market as they currently do. If the market were to become more efficient the model would probably have to be more advanced in order to trade on the few anomalies still present in the market.

The model was initially tested on data simulated from the two-state Markov chain. This was a way to simulate returns similar to real market returns in order to test the implementation of the model. The HMM will in this case estimate the parameters almost perfectly, implying the is nothing wrong with the parameter estimation in the HMM. The predictions and the MPC trading algorithm will yield a strong result as well. As there is an element of noise, or randomness, in the simulated data and the fact that the simulated market state was unobservable, no model will be able to execute every trade perfectly. The algorithm will however be able to yield a stronger return than a static portfolio on simulated market data at every try. The authors are therefore certain that nothing is wrong with the implementation of the model. In the case that the algorithm fails to yield a stronger return than the market portfolio, the problem is likely explained by that the model fails to approximate the market rather than that the implementation is erroneous.

Much effort was put into making the HMM's parameter estimation numerically stable and robust for different choices of hyperparameters. One aspect of this meant, in practice, choosing an appropriate approximation for the gradient and Hessian to the log-likelihood function. A variety of methods for approximating the Hessian were implemented and tested (see appendix A.3). The final choice was based on that the approximation was numerically stable and that the resulting parameter estimations were considered reasonable. Since implementation of this part of the algorithm was a bit of a struggle, one should be wary that the chosen solution could be suboptimal. The time taken to finish calculations in one run of the algorithm was also in the order of magnitude of ten minutes. This meant that the speed of the code was

also a factor in the choice of method that could have affected the optimality of the implementation. Other alternatives of approximations are presented in appendix A.3. Nevertheless, and as stated before, the choice was deemed to be satisfactory for the scope of this thesis. Additional ways to improve the stability of the estimates could for example be to introduce priors to the HMM's parameters or to use different parameter transformations with desired properties. The transformations could also be used to keep the transition probabilities within a more reasonable interval $[a, b]$ for some $0 < a < 1$.

7.1 Areas for further research

Looking at table 1.1, it can easily be seen that there are still areas left to be explored. Firstly, the model described in this thesis should be examined with the purpose of assessing how well it can reproduce the stylized facts. It could also be of use to compare the available classes of HMM models with each other, i.e. extending the work done in Nystrup, Madsen, and Lindström (2016). It would also be interesting to study the performance of different trading algorithms, based on different classes, and comparing the performance. A good suggestion of a trading algorithm for these experiments could be to continuing using the MPC algorithm just like in this thesis, since it can easily be used for all classes of HMMs.

There is also a possibility that an EM based adaptive HMM estimator could be both computationally fast and numerically robust. This model could be similar to the one in Nystrup et al. (2017, pp. 9-12), but making use of Student's t-distribution. The article by Liu and Rubin (1995) covers estimation using EM of Student's t-distribution. Bulla (2011, p. 461) also refers to some potentially interesting literature regarding the subject as well.

As previously stated, the authors believe that Bayesian optimization has potential to be useful for hyperparameter tuning, only that the BO was applied in a too crude way in this thesis. It is also theorized that dimensional could be the cause for the poor performance. One obvious solution is to simplify the problem by fixing even more hyperparameters. Another possible solution to the dimensionality problem is to use the fact that estimation can be done separately from trading. One could then first attempt to find good hyperparameters for the HMM parameter estimation (although it would require a measure for goodness-of-fit to the data, e.g. log-likelihood). Afterwards, a subset of those hyperparameters configurations could be used then optimize the trading hyperparameters for the MPC step with excess return as performance measurement. This would reduce the problems dimensionality, which could yield smarter sampling and thus possibly better sets of hyperparameters and overall better performance. Said approach was not investigated because it was considered out of scope for this thesis.

Yet another possible technique to improve performance is to use an acquisition function that attempts to discourage instable peaks and local optima in the objective function, such as the one presented in the article Nguyen et al. (2017). The use of this acquisition function could possibly further mend the problem with over-fitting the hyperparameters, but further investigation has to be performed to verify this.

Minor areas

For the interested reader; below is a list of some additional minor technical changes that could be investigated to improve the trading algorithm further. It should be noted that these points are regarded of minor importance from the authors current

point of view. They are not covered due to limitations, and time constraints, of the study.

1. Test the unimplemented approximation methods for the HMM's LL function's Hessian as presented in the appendix, section A.3.

2. Attempt to improve the algorithm to use more than one instrument simultaneously. There could e.g. be more than one asset that is traded upon and as well as a number of assets used to estimate the markets current regime (such as the volatility index VIX).

3. Investigate effect of letting the trade aversion parameter in equation (2.40) be defined as a function dependent on some variable, e.g. the portfolio's maximum drawdown. This is done i Nystrup et al. (2017), with a non-Student's t-distribution based model, yielding positive results.

4. Investigate the use of a hidden semi-Markov model, in order to model e.g. psychological effects of the market's expectation of the regimes' sojourn-times. A continuous-time Markov model could even be used to model non-equidistant samples to cope with weekends and banking holidays better. This is discussed in Nystrup, Madsen, and Lindström (2015) but not in the context of a trading algorithm.

5. Investigate the numerical stability when approximating the truncated MGF of Student's t-distribution in equation (2.37) using adaptive quadrature and generalized Laguerre-Gauss. Golub and Welsch (1969) describes some basics of Gauss quadrature and Shampine (2008) discusses MATLAB's adaptive quadrature implementation.

Appendix A

Miscellaneous formulas

A.1 Derivatives of distribution densities

The derivates of the PDFs for the observed log-returns with regards to the distributions' parameters are used in the LH algorithm (as described in subsection 2.1.2). On a side note; the following implication was used to simplify the process of deriving the derivatives of the PDFs w.r.t. to some arbitrary parameter θ:

$$g(\cdot) = \log f(\cdot) \implies \frac{\partial f}{\partial \theta} = \Big(\frac{\partial g}{\partial \theta}(\cdot)\Big) f(\cdot) \tag{A.1}$$

The normal PDF is given by (Johnson, Kotz, and Balakrishnan, 1994, p. 80):

$$\phi(x; \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \cdot \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \tag{A.2}$$

where the parameters are the mean μ and standard deviation $\sigma > 0$. The derivative w.r.t. μ is:

$$\frac{\partial}{\partial \mu}\phi(x; \mu, \sigma) = \frac{x-\mu}{\sigma^2} \cdot \phi(x; \mu, \sigma) \tag{A.3}$$

and w.r.t. σ:

$$\frac{\partial}{\partial \sigma}\phi(x; \mu, \sigma) = \left(-\frac{1}{\sigma} + \frac{(x-\mu)^2}{\sigma^3}\right) \cdot \phi(x; \mu, \sigma). \tag{A.4}$$

The Student's t-distribution's PDF (non-standardized) is given by (Johnson, Kotz, and Balakrishnan, 1995, p. 363):

$$t(x; \mu, \sigma, \nu) = \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\Gamma\left(\frac{\nu}{2}\right)} \cdot \frac{1}{\sigma\sqrt{\nu\pi}} \cdot Z^{-(\nu+1)/2}, \; Z = \left(1 + \frac{(x-\mu)^2}{\nu\sigma^2}\right) \tag{A.5}$$

where the parameters are mean μ, standard deviation $\sigma > 0$ and degrees of freedom $\nu > 0$. The derivative w.r.t. to μ can be calculated as:

$$\frac{\partial}{\partial \mu}t(x; \mu, \sigma, \nu) = -\frac{\nu+1}{2} \cdot \frac{\frac{\partial}{\partial \mu}Z}{Z} \cdot t(x; \mu, \sigma, \nu), \tag{A.6}$$

$$\frac{\partial}{\partial \mu}Z = -2\frac{(x-\mu)}{\nu\sigma^2}.$$

The derivative w.r.t. σ as:

$$\frac{\partial}{\partial \sigma}t(x; \mu, \sigma, \nu) = \left(-\frac{1}{\sigma} - \frac{\nu+1}{2} \cdot \frac{\frac{\partial}{\partial \sigma}Z}{Z}\right) \cdot t(x; \mu, \sigma, \nu), \tag{A.7}$$

$$\frac{\partial}{\partial\sigma}Z = -2\frac{(x-\mu)^2}{\nu\sigma^3}$$

and w.r.t. ν as:

$$\begin{aligned}\frac{\partial}{\partial\nu}t(x;\mu,\sigma,\nu) =& \left(\frac{1}{2}\psi\left(\frac{\nu+1}{2}\right) - \frac{1}{2}\psi\left(\frac{\nu}{2}\right) - \frac{1}{2\nu} - \right. \\ & \left. - \frac{1}{2}\log Z - \frac{\nu+1}{2}\cdot\frac{\frac{\partial}{\partial\nu}Z}{Z}\right)\cdot t(x;\mu,\sigma,\nu)\end{aligned}, \tag{A.8}$$

$$\frac{\partial}{\partial\nu}Z = -\frac{(x-\mu)^2}{\nu^2\sigma^2}$$

where $\psi(x) = \frac{\mathrm{d}}{\mathrm{d}x}\log\Gamma(x)$ is the digamma function.

A.2 Transformations

Two transformations are used to make parameters unconstrained in the algorithm described in this thesis; the exponential and logistic transformations.

The exponential function can be used to convert a non-negative variable to a real-valued variable in the parameter space. The exponential function is simply:

$$f(x) = e^x \in [0,\infty), \quad x \in \mathbb{R}. \tag{A.9}$$

It's inverse is:

$$f^{-1}(y) = \log(y) \tag{A.10}$$

and the derivative is:

$$f'(x) = f(x). \tag{A.11}$$

The logistic can be used to transform a probability $p \in [0,1]$ to the unconstrained real parameter space $\mathbb{R}$ (Boyd and Vandenberghe, 2004, p. 122). The logistic function (which is sometimes called the sigmoid function or the squashing function) is defined as:

$$f(x) = \frac{1}{1+e^{-x}} \in [0,1], \quad x \in \mathbb{R}. \tag{A.12}$$

It's inverse is:

$$f^{-1}(x) = -\log(y^{-1} - 1) \tag{A.13}$$

and the derivative is:

$$f'(x) = f(x)\cdot(1 - f(x)). \tag{A.14}$$

A.3 Alternative Hessian approximations

In this section a number of alternative approximations of the Hessian, usable in the GAS model type adaptive estimator (2.11), are presented. As previously stated, the adaptive estimator can also be derived from a second order stochastic approximation (Spall, 2003, pp. 116-117).

The first alternative is based on the fact that the true Hessian of the truncated weighted LL can simply be calculated using the LH algorithm (Lystig and Hughes,

2002, pp. 684-685). This would yield the equation:

$$\hat{\mathbf{H}}_t = \nabla^2_{\boldsymbol{\theta}\boldsymbol{\theta}} \tilde{\ell}_{t,L}(\boldsymbol{\theta}_{t-1}). \tag{A.15}$$

This approximation was not implemented nor tested due to time constraints.

A second alternative is an approximation which is related to the Fisher information matrix. This approximation would yield an adaptive estimator that is similar to the Berndt-Hall-Hall-Hausman, abbrevated BHHH, optimization algorithm (Berndt et al., 1974). The definition is as follows:

$$\hat{\mathbf{H}}_t = \sum_{\tau=t^-}^{t} \left[\nabla_{\boldsymbol{\theta}} \mathbb{P}(y_t \mid y_{:\tau-1}, \hat{\boldsymbol{\theta}}_{t-1})\right] \left[\nabla_{\boldsymbol{\theta}} \mathbb{P}(y_t \mid y_{:\tau-1}, \hat{\boldsymbol{\theta}}_{t-1})\right]^{\mathrm{T}}, \; t^- = t - L + 1. \tag{A.16}$$

This approximation was not used in this thesis since it resulted in numerically unstable estimations of the HMM's parameters. Although, this approximation is used, with good results, in Nystrup, Madsen, and Lindström (2016) for a non-Student's t-distribution based HMM.

A third alternative utilizes the fact that the inverse of the Hessian can be updated directly using the matrix inversion lemma (Jakobsson, 2015, p. 309). This is based on the update:

$$\hat{\mathbf{H}}_t = \lambda \hat{\mathbf{H}}_{t-1} + \mathbf{d}_t \mathbf{d}_t^{\mathrm{T}} \tag{A.17}$$

where λ is the forgetting factor and:

$$\mathbf{d}_t = \nabla_{\boldsymbol{\theta}} \tilde{\ell}_{t,L}(\boldsymbol{\theta}_{t-1}) - \nabla_{\boldsymbol{\theta}} \tilde{\ell}_{t-1,L}(\boldsymbol{\theta}_{t-2}). \tag{A.18}$$

This results in the recursive approximation of the Hessian's inverse:

$$\hat{\mathbf{H}}_t^{-1} = \frac{1}{\lambda} \left(\hat{\mathbf{H}}_{t-1}^{-1} - \frac{\hat{\mathbf{H}}_{t-1}^{-1} \mathbf{d}_t \mathbf{d}_t^{\mathrm{T}} \hat{\mathbf{H}}_{t-1}^{-1}}{\lambda\big(1 + \mathbf{d}_t^{\mathrm{T}} \hat{\mathbf{H}}_{t-1}^{-1} \mathbf{d}_t / \lambda\big)} \right). \tag{A.19}$$

This is somewhat similar to the recursive estimator used in the recursive prediction error method (Jakobsson, 2015, pp. 279-283). This approximation was also not implemented due to time constraints in the study.

Bibliography

Baum, Leonard E. (1972). "An inequality and associated maximization technique in statistical estimation of probabilistic functions of a Markov process". In: *Inequalities* 3, pp. 1–8.

Berndt, E. K., B. H.Hall, R. E.Hall, and I. A. Hausman (1974). "Estimation and Inference in Nonlinear Structural Models". In: *Annals of Economic and Social Measurement* 3.4, pp. 653–665. URL: http://www.nber.org/chapters/c10206.

Boyd, Stephen and Lieven Vandenberghe (2004). *Convex Optimization*. Cambridge: Cambridge University Press.

Boyd, Stephen et al. (2017). "Multi-Period Trading via Convex Optimization". In: *Foundations and Trends in Optimization* 3, pp. 1–76. URL: https://arxiv.org/abs/1705.00109.

Brochu, Eric, Vlad M. Cora, and Nando de Freitas (2010). "A Tutorial on Bayesian Optimization of Expensive Cost Functions, with Application to Active User Modeling and Hierarchical Reinforcement Learning." In: *ArXiv e-prints*. arXiv: 1012.2599. URL: https://arxiv.org/abs/1012.2599.

Bubeck, Sébastien and Nicolò Cesa-Bianchi (2012). "Regret Analysis of Stochastic and Nonstochastic Multi-armed Bandit Problems". In: *Foundations and Trends in Machine Learning* 5, pp. 1–122. URL: https://arxiv.org/abs/1204.5721.

Bulla, J. (2011). "Hidden Markov models with t components. Increased persistence and other aspects". In: *Quantitative Finance* 11.3, pp. 459–475. URL: https://www.tandfonline.com/doi/abs/10.1080/14697681003685563.

Bulla, J. et al. (2011). "Markov-switching asset allocation: Do profitable strategies exist?" In: *Journal of Asset Management* 12.5, pp. 310–321. URL: https://link.springer.com/article/10.1057%2Fjam.2010.27.

Creal, Drew, Siem Jan Koopman, and André Lucas (2013). "Generalized Autoregressive Score Models with Applications". In: *Journal of Applied Econometrics* 28, pp. 777–795. URL: http://onlinelibrary.wiley.com/doi/10.1002/jae.1279/abstract.

Eaton, Morris L. (1983). *Multivariate statistics : a vector space approach*. Chichester, New York: Wiley.

Golub, Gene H. and John H. Welsch (1969). "Calculation of Gauss Quadrature Rules". In: *Mathematics of Computation* 23.106, pp. 221–230. URL: http://www.jstor.org/stable/2004418.

Jakobsson, Andreas (2015). *An Introduction to Time Series Modeling*. 2nd ed. Lund: Studentlitteratur AB.

Jawitz, James W. (2004). "Moments of truncated continuous univariate distributions". In: *Advances in Water Resources* 27.3, pp. 269–281. URL: http://www.sciencedirect.com/science/article/pii/S0309170803001787.

Johnson, Norman L., Samuel Kotz, and N. Balakrishnan (1994). *Continuous univariate distributions*. 2nd ed. Vol. 1. New York: Wiley.

— (1995). *Continuous univariate distributions*. 2nd ed. Vol. 2. New York: Wiley.

Jones, Donald R., Matthias Schonlau, and William J. Welch (1998). "Efficient Global Optimization of Expensive Black-Box Functions". In: *Journal of Global Optimization* 13, pp. 455–492. URL: https://link.springer.com/article/10.1023/A:1008306431147.

Kim, Hea-Jung (2008). "Moments of truncated Student-t distribution". In: *Journal of the Korean Statistical Society* 37.1, pp. 81–87. URL: http://www.sciencedirect.com/science/article/pii/S1226319208000082.

Kushner, Harold J. (1964). "A New Method of Locating the Maximum Point of an Arbitrary Multipeak Curve in the Presence of Noise". In: *Journal of Basic Engineering* 86, pp. 97–106. URL: http://fluidsengineering.asmedigitalcollection.asme.org/article.aspx?articleid=1431594.

Lindgren, Georg, Holger Rootzén, and Maria Sandsten (2014). *Stationary Stochastic Processes for Scientists and Engineers*. Boca Raton: CRC Press.

Lindström, Erik, Henrik Madsen, and Jan Nygaard Nielsen (2015). *Statistics for Finance*. Boca Raton: CRC Press.

Liu, Chuanhai and Donald B. Rubin (1995). "ML estimation of the t distribution using EM and its extensions, ECM and ECME". In: *Statistica Sinica* 5.1, pp. 19–39. URL: http://www3.stat.sinica.edu.tw/statistica/j5n1/j5n12/j5n12.htm.

Lizotte, Daniel James (2008). "Practical Bayesian Optimization". PhD thesis. University of Alberta. URL: http://www.csd.uwo.ca/~dlizotte/publications/lizotte_phd_thesis.pdf.

Lystig, Theodore C. and James P. Hughes (2002). "Exact Computation of the Observed Information Matrix for Hidden Markov Models". In: *Journal of Computational and Graphical Statistics* 11.3, pp. 678–689. URL: http://www.jstor.org/stable/1391119.

Markowitz, Harry (1952). "Portfolio Selection". In: *The Journal of Finance* 7, pp. 77–91. URL: http://onlinelibrary.wiley.com/doi/10.1111/j.1540-6261.1952.tb01525.x/full.

Mattingley, Jacob and Stephen Boyd (2012). "CVXGEN: a code generator for embedded convex optimization". In: *Optimization and Engineering* 13.1, pp. 1–27. URL: https://link.springer.com/article/10.1007%2Fs11081-011-9176-9.

Merton, Robert C. (1969). "Lifetime Portfolio Selection under Uncertainty: The Continuous-Time Case". In: *The Review of Economics and Statistics* 51.3, pp. 247–257. URL: http://www.jstor.org/stable/1926560.

Nasdaq, Inc. (2017). *Circuit Breaker*. URL: https://www.nasdaqtrader.com/trader.aspx?id=CircuitBreaker (visited on 04/03/2018).

Nguyen, Thanh Dai, Sunil Gupta, Santu Rana, and Svetha Venkatesh (2017). "Stable Bayesian Optimization". In: *Advances in Knowledge Discovery and Data Mining*. Proceedings of PAKDD 2017, pp. 578–591.

Nystrup, Peter (2017). "Dynamic Asset Allocation - Identifying Regime Shifts in Financial Time Series to Build Robust Portfolios". PhD thesis. Technical University of Denmark.

Nystrup, Peter, Henrik Madsen, and Erik Lindström (2015). "Stylised facts of financial time series and hidden Markov models in continuous time". In: *Quantitative Finance* 15.9, pp. 1531–1541. URL: https://www.tandfonline.com/doi/full/10.1080/14697688.2015.1004801.

— (2016). "Long Memory of Financial Time Series and Hidden Markov Models with Time-Varying Parameters". In: *Journal of Forecasting* 36, pp. 989–1002. URL: http://onlinelibrary.wiley.com/doi/10.1002/for.2447/full.

— (2017). "Dynamic portfolio optimization across hidden market regimes". In: *Quantitative Finance*. URL: https://www.tandfonline.com/doi/full/10.1080/14697688.2017.1342857.

Nystrup, Peter, Stephen Boyd, Erik Lindström, and Henrik Madsen (2017). "Multi-Period Portfolio Selection with Drawdown Control". Read by authors in Nystrup (2017). To appear in *Annals of Operations Research*.

Nystrup, Peter et al. (2018). "Dynamic Allocation or Diversification: A Regime-Based Approach to Multiple Assets". In: *The Journal of Portfolio Management* 44 (2), pp. 62–73. URL: http://jpm.iijournals.com/content/44/2/62.

Ranjan, Pritam, Ronald Haynes, and Richard Karsten (2011). "A Computationally Stable Approach to Gaussian Process Interpolation of Deterministic Computer Simulation Data". In: *Technometrics* 53.4, pp. 366–378. URL: https://www.tandfonline.com/doi/abs/10.1198/TECH.2011.09141.

Rasmussen, Carl Edward and Christopher K. I. Williams (2006). *Gaussian Processes for Machine Learning*. Cambridge: The MIT Press. URL: http://www.gaussianprocess.org/gpml/chapters/.

Shahriari, Bobak et al. (2016). "Taking the Human Out of the Loop: A Review of Bayesian Optimization". In: Proceedings of IEEE 2016 (Volume 104, Issue 1), pp. 148–175.

Shampine, L. F. (2008). "Vectorized adaptive quadrature in MATLAB". In: *Journal of Computational and Applied Mathematics* 211.2, pp. 131 –140. URL: http://www.sciencedirect.com/science/article/pii/S037704270600700X.

Sharpe, William F. (1964). "Capital Asset Prices: A Theory of Market Equilibrium Under Conditions of Risk". In: *The Journal of Finance* 19.3, pp. 425–442. URL: https://onlinelibrary.wiley.com/doi/10.1111/j.1540-6261.1964.tb02865.x.

Spall, James C. (2003). *Introduction to Stochastic Search and Optimization: Estimation, Simulation, and Control*. Hoboken: Wiley. URL: https://onlinelibrary.wiley.com/doi/book/10.1002/0471722138.