Bayesian Networks for Asset Management and Financial Risk

Denis de Montigny

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
of
University College London.

Department of Computer Science
University College London

July 27, 2020

I, Denis de Montigny, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the work.

Abstract

This thesis explores the use of Bayesian networks to develop "views" for a Black-Litterman asset allocation model, and determines whether they can help in the creation of better investment portfolios. Views represent an investor's expectations of the future performance of a company's shares: an estimate of expected return, and a measure of the uncertainty of this estimate. This thesis aims to automate the creation of views and to pioneer intelligent portfolio construction as part of an algorithmic asset management process.

Portfolio construction in asset management is typically performed with the objective of maximizing expected return and minimizing expected risk. One model which is used for this purpose is the Black-Litterman model. This model requires an input from the portfolio manager: an estimate of future returns for each asset included in the portfolio. This research develops predictions of returns and an estimate of the uncertainty of this prediction using Bayesian networks, as input for a Black-Litterman model. This hybrid approach is then tested under trading conditions. This research also develops a scientific framework and platform which allows for the rapid analysis of any algorithm for generating views as well as any general portfolio construction model.

This research is important because the results contribute to a new generation of asset management by combining classical theories with computational intelligence, automating and enhancing part of the process. The ability to rapidly include, analyze, and compare models developed by other researchers would greatly facilitate the communication of research results with the asset management industry.

This research comprises three experiments:

1. **A Benchmark Portfolio Using Traditional Algorithms**. The first experiment establishes a benchmark, using a standard Black-Litterman approach, against which further models will be compared. It also develops and validates the analysis platform. The performance of this benchmark portfolio is found to be below the performance of the index on which it is based for virtually all periods, on a risk adjusted basis. It is concluded that it is interesting to investigate whether a Bayesian network may be used to develop views which would enhance the performance of the portfolio.

2. **Generating Views Using a Bayesian Network**. The second experiment investigates the use of Bayesian networks for asset management as input to a Black-Litterman model. Different approaches are compared using linear regression, autoregression, GARCH, EGARCH and EGARCH-M. A limited set of factors is used, including the price of oil, the spread between 10 year and 2 year treasuries, the spread between BAA and AAA bonds, and a stress index developed by the Federal Reserve Bank of St-Louis (FRED)[1]. The resulting return predictions are compared to realized returns. While the main objectives of this experiment were met, further research is required in order to build Bayesian networks which better predict returns as well as estimates of the uncertainty of the predictions.
3. **Algorithmic Asset Management System**. This experiment extends the platform developed in the previous experiments by including more detailed analysis, and further develops the Bayesian network of the second experiment to apply it to daily use in a trading environment using out of sample data to validate results. This experiment demonstrates how Bayesian networks may be used to generate views, constructs portfolios, and analyses their performance. In support of the hypothesis that markets are efficient, the portfolios constructed in this experiment did not significantly outperform their benchmark. Nevertheless, as their performance was sometimes above and otherwise not far below their benchmark, these results are encouraging. Further research may yield better results. The robust scientific framework and platform finalized in this experiment can be used in order to easily explore and compare other Bayesian networks, as well as other machine learning models.

This research makes the following contributions to science:

1. **A Benchmark Portfolio Using Traditional Algorithms**. Developing a platform providing back-testing, comparison, and basic analysis of algorithms to generate views or other inputs for potentially any general portfolio construction model. This platform may ultimately help improve communication of basic research in portfolio modelling and analysis with the asset management industry by providing a robust and reliable auditing tool.
2. **Generating Views Using a Bayesian Network**. Demonstrating that Bayesian networks, using a variety of models and including external factors, may be used to effectively model risk and returns.
3. **Algorithmic Asset Management System**. Providing an analysis platform with full flexibility to use different means of analysis, algorithms for generating *views*, portfolio management models, and data sets. Demonstrating the robustness of an approach for generating *views* based on Bayesian networks in actual trading conditions.

[1]The FRED provides historical data as part of its mandate to provide high-quality economic research `https://fred.stlouisfed.org/`.

Impact Statement

This research is expected to lead to a greater understanding of the application of artificial intelligence in asset management, and to lead to a framework which can be used to facilitate communication of academic research results with the asset management industry.

Convincing members of the asset management industry that a new approach to asset management has value is a long process. Proving that the new approach would have worked in the past using a backtest is necessary but not sufficient[2]. A backtest is necessary since, if the approach would not have worked in the past, it is considered unlikely to work in the future. Failure to use separate training and testing datasets, failure to account for the fact that the dataset may exclude companies which have gone bankrupt, and unrealistic assumptions regarding liquidity and market impact are some reasons a backtest is considered insufficient. Even careful academic research is controversial. For example, while Fama and French (1992) argue they have uncovered important results, Black (1993a) argues these results are due to data mining and are unlikely to work in the future. Convincing asset managers of the usefulness of a new approach requires, in practice, that the approach be shown to work on live data over a three to five year period. In the current rapidly evolving environment, one may suspect that, by the time an approach can be shown to work, the underlying technology has already been widely adopted and the approach is no longer useful. A scientific framework and platform which, itself, can be shown to be useful in the early discovery of valuable new approaches would be of practical interest.

[2]See for example http://methodical.co.za/literature/ive-never-seen-bad-backtest/

Acknowledgements

I would like to thank my PhD supervisors, Professor Philip Treleaven and Dr. Paolo Barucca for their unfailing guidance and enthusiasm. I am very grateful for the support and inspiration that I have received over the past years.

I also want to thank my wife Nathalie, and my children, Héloïse and Evann. This thesis would not have been possible without your continued support and encouragement. I dedicate this thesis to my family.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This chapter introduces the motivations and objectives for the research, briefly describes the design, implementation, and testing of the experiments, lists its contributions to science, and presents the structure of the thesis.

Portfolio construction in asset management is typically performed with the objective of maximizing expected return and minimizing expected risk. One model which is used for this purpose is the “Black-Litterman” model. This model requires an input from the portfolio manager: an estimate of future returns for each asset included in the portfolio, as well as a measure of the uncertainty of this estimate. These two numbers are commonly referred to as “views”. This research investigates the application of Bayesian networks to develop views for a Black-Litterman model.

This hybrid approach is then tested under trading conditions. This research also develops a back-testing framework which allows for the rapid analysis of any algorithm for generating views as well as any general portfolio construction model. The use of this scientific platform aims to pioneer intelligent portfolio construction as part of an algorithmic asset management process.

1.1 Research Motivation

One key objective of portfolio construction is to produce investment portfolios with the highest expected return for an expected level of risk. In a mean-variance framework, risk is quantified as variance of returns, and the objective is to maximize expected mean returns for a given level of variance. The Black-Litterman model is one model which may be used to construct mean-variance efficient portfolios which incorporate user-generated views.

This research is motivated by the fact that views for a Black-Litterman model are subjective, expensive, and time-consuming to produce. They may also be sub-optimal. They are subjective because they depend on expert knowledge. They are expensive and time-consuming to produce because they rely in part on fundamental research, analysis of events, and data. They may be sub-optimal when asset managers are not experts in all aspects of the assets in their portfolio.

This research investigates Bayesian networks to generate views because they can encode expert knowledge and combine it with data, and they may provide an improvement over the current, largely manual, process. Models based on Bayesian networks are also interesting to investigate for the following reasons (Koller and Friedman, 2009; Lauritzen, 1996; Rebonato and Denev, 2014). First, stress scenarios can be included in these models by specifying the conditions under which tail events may be expected to happen, given current market conditions. This may be interesting as the Black-Litterman model assumes returns are normally distributed while there is considerable evidence that they are not (Cont, 2001; Fama, 1965b; Mandelbrot, 1963). Second, the networks offer some level of transparency because relations between factors may be established. This helps managers understand the networks and communicate their conclusions. Third, because graphs are modular, they can be extended with new factors without impacting the whole network. Fourth, efficient computer algorithms to work with graphs are available. For example, methods of approximate inference exist where exact inference is NP hard.

Probabilistic graphical models are an active area of research at University College London including work on representation, inference, learning, applications to time series models, and to mobile pedestrian localization (Barber, 2016; Barfuss et al., 2016; Bracegirdle, 2013; Oikonomou-Filandras, 2016).

In applying Bayesian networks to develop views, this thesis implicitly investigates whether these models can be used to generate new information which is not reflected in current market prices, in possible contradiction to some forms of the efficient market hypothesis.

1.2 Research Objectives

The main objective of this research is to determine whether Bayesian networks may be used to generate views that result in portfolios with improved risk/return characteristics.

This main objective may be divided as follows:

- Develop a classical Black-Litterman model which serves as a benchmark against which to compare other portfolios.
- Develop a platform to analyze and compare portfolio allocation models. Test the platform by including historical price data for an index and a simple equal-weighted portfolio construction model.
- Enhance the classical Black-Litterman model with a Bayesian network and determine

whether returns can be shown to improve.

- Enhance the Bayesian network using different algorithms and determine whether returns can be shown to improve.
- Enhance the analysis platform into a comprehensive tool which is capable of using different analysis methods. Integrate the results obtained in this thesis to provide a flexible tool enabling the use of different equity selection, machine learning, portfolio generation, and portfolio analysis models.

1.3 Research Methodology

A back-testing platform is developed and tested by including historical price data for the selected equity index, the NASDAQ-100, as well as a simple portfolio model which allocates equal weights to each equity in the index. Standard metrics are used to compare the performance of portfolios created using these models including annual return, variance, and measures such as the Sharpe ratio.

A standard unconstrained Black-Litterman model is implemented to develop a standard model portfolio against which other portfolios developed using Bayesian networks are compared.

This research is comprised of three experiments, based on the following data set :

1.3.1 Data Set

Work is based on the NASDAQ-100 (the Index), which represents the 100 largest companies listed on the NASDAQ weighted by market capitalization, excluding companies in the financial sector. It is the basis for one of the most actively traded exchange traded funds in the world, the Powershares QQQ (the ETF). The Index was formed on January 31, 1985.

The Index is reviewed on a quarterly basis and rebalanced on an annual basis, based on market values at end of October and shares outstanding at the end of November of each year. Rebalancing is published beginning December.[1]

The list of stocks which comprised the Index on a quarterly or monthly basis for the period from 01/01/1995 to 30/09/2017 was obtained from Siblis Research[2]. Historical price data adjusted for stock splits and dividend payments for these stocks, as well as the value of the Index itself, was initially obtained from Yahoo and Quandl[3], and then from Interactive Brokers. When this data is incomplete - for example, historical data is not always available for companies which no longer exist today - it may be necessary to reduce the dataset by

[1]Further information on the index: `http://www.nasdaq.com/markets/indices/nasdaq-100.aspx`, https://indexes.nasdaqomx.com/docs/NDX_Fundamentals.pdf, and `https://en.wikipedia.org/wiki/NASDAQ-100`

[2]Last accessed 27.03.2018: `http://siblisresearch.com/data/historical-components-nasdaq/`

[3]Accessed via API or web scraping. `https://finance.yahoo.com` and `https://www.quandl.com/tools/api`

excluding certain equities.

1.3.2 A Benchmark Portfolio Using Traditional Algorithms

The first experiment establishes a benchmark against which the performance of other models are compared. This is useful for two reasons. First, although the annual return of the benchmark portfolio could be expected to be close to the annual return of the ETF and the Index, returns may not be identical for several reasons. For example, the ETF will pay a number of fees, including management fees and brokerage fees. In addition, although both the Black-Litterman portfolio and the Index contain the same securities, the allocation calculated by the Black-Litterman algorithm is unlikely to correspond exactly to the one used for the Index and the ETF. Second, the final experiment is based on a subset of the Nasdaq-100 Index. The QQQ ETF is therefore not a suitable benchmark against which to compare the performance of the portfolios constructed. For these reasons, explicitly creating a benchmark portfolio ensures that differences in performance between the benchmark and the portfolios generated using other models are attributable to these models.

The dataset does not include all stocks which comprise the Index. Returns of portfolios constructed using this reduced dataset will not be comparable to the returns of the ETF or the Index. This experiment constructs a benchmark using this reduced dataset. It therefore provides an objective and consistent basis on which to compare the returns of models making use of Bayesian networks, which also use this reduced dataset.

Given historical prices and the market capitalization of shares underlying the index, the Black-Litterman model is used to extract equilibrium returns for these shares, based upon the correlations of the changes in their prices, and their relative weights by market capitalization. These equilibrium returns are not modified by views, and are used directly in a mean-variance optimizer to produce the final benchmark portfolio weights.

The annual returns on the benchmark portfolio are compared to the returns on the Index and the ETF. In principle, the returns on the benchmark portfolio are expected to exceed the returns of the ETF by around 8 to 10 basis points, which correspond to the administration and management fees of the fund.

In order to compare different portfolio construction models, an analysis platform is developed and tested. This platform enables rapid back-testing and analysis of models.

1.3.3 Generating Views Using a Bayesian Network

The second experiment investigates the use of Bayesian networks for asset management. The Bayesian networks are constructed so as to be flexible, and integrate the scientific platform being developed.

Bayesian networks are used to construct views, consisting of predicted equity returns as well as a level of confidence in these predictions. Returns are assumed to be a function of a number of factors, including past returns and changes in oil prices, inflation, and interest rates. The functional forms modelled include GARCH, EGARCH, and EGARCHM.

The results are compared to realized returns over the period.

1.3.4 Algorithmic Asset Management System

The third experiment integrates the first and second experiments to provide a flexible tool enabling the use of different equity selection, machine learning, portfolio generation, and portfolio analysis algorithms. It applies this tool in a trading environment, using daily market data.

The data are a subset of the equities which are part of the Nasdaq-100 Index. The 10 equities having returns with the highest correlation to the returns on the QQQ ETF are selected for inclusion in the portfolios.

Factors which could be included in the analysis are discussed. As the factor data must be available at time of trading, proxies are considered for some factors which have been used in research. Principal Component Analysis is used to reduce the dimensions of the factor space to reduce computation time.

Portfolios are created using the Black-Litterman model developed in Experiment 1 and the Bayesian networks developed in Experiment 2 for views.

The performance of the portfolios is compared to the performance of a "No Views" benchmark portfolio and an "Equal Weights" portfolio.

1.4 Contributions to Science

This research makes the following contributions to science:

1. **A Benchmark Portfolio Using Traditional Algorithms**. This experiments develops a platform providing back-testing, comparison, and basic analysis of algorithms to generate *views* and potentially any general portfolio construction model. This platform may ultimately help improve communication of cutting edge research in portfolio modelling and analysis with the asset management industry by providing a robust and reliable auditing tool.
2. **Generating Views Using a Bayesian Network**. This experiment demonstrates how Bayesian networks may be used to effectively model equity returns and uncertainty of returns. It explores and compares Bayesian networks using GARCH, EGARCH, and EGARCHM models more thoroughly than previous research.
3. **Algorithmic Asset Management System**. This experiment demonstrates how Bayesian networks may be used to generate views to construct portfolios. It applies this approach with Bayesian networks in daily trading conditions, utilizes these views to create portfolios, and demonstrates tools to compare the performance of

these portfolios. In support of the hypothesis that markets are efficient, the portfolios constructed in this experiment were below though close to their benchmark. This experiment also finalizes an analysis platform with full flexibility to use different means of analysis, algorithms for generating views, portfolio management models, and data sets. It finally identifies further areas of research in portfolio construction.

1.5 Structure of the Thesis

The thesis is organized as follows.

- Chapter 2 - Background and Literature Review. This chapter presents background information and research on the key concepts relevant to the research topic, including algorithmic asset management, asset allocation models, and Bayesian networks.
- Chapter 3 - Experimental Data and Scientific Platform. This chapter describes the data used in this thesis, and the design and testing of the analysis platform.
- Chapter 4 - A Benchmark Portfolio Using Traditional Algorithms. This chapter presents the design, implementation, testing, and results of the first experiment which establishes a benchmark portfolio against which other models will be compared.
- Chapter 5 - Generating Views Using a Bayesian Network. This chapter presents the design, implementation, testing, and results of the second experiment which investigates the use of probabilistic graphical models for asset management. Different representations will be compared.
- Chapter 6 - Algorithmic Asset Management System. This chapter presents the design, implementation, testing, and results of the third experiment which integrates the first and second experiments to provide a flexible tool enabling the use of different equity selection, machine learning, portfolio generation, and portfolio analysis algorithms. This tool is used with out of sample daily market data.
- Chapter 7 - Conclusions and Future Research. The final chapter provides general conclusions which can be drawn from the research performed, highlights key findings and contributions to current knowledge, and indicates potential areas for further research.

Chapter 2

Background and Literature Review

This chapter presents background information and research on the key concepts relevant to the research topic, including asset allocation models, risk, algorithmic asset management, and Bayesian networks.

2.1 Context

Asset allocation is the process of constructing an investment portfolio of risky assets in such a way as to meet an investor's objectives. The goal is to determine the portfolio weights or the percentage of the portfolio to invest in each asset.

In general, individual weights may be negative if short selling is allowed. The constraint that all weights must be positive is often imposed in implementation.

The investor's first objective is normally assumed to be to maximize expected return, however there may be other objectives, including for example socially responsible investing in which certain companies are avoided on the basis that they do not meet an investor's social, environmental, governance, or similar requirements. These types of objectives are often implemented as constraints by the selection of an appropriate benchmark against which the performance of the portfolio is measured, and by the selection of an appropriate universe of assets from which the portfolio is constructed.

Investors are normally assumed to be risk averse. They are assumed to prefer a higher return, and to prefer lower risk. The level of an investor's risk aversion may be influenced by a variety of factors, including the investor's age, liquidity requirements, and personality. Different investors may therefore have different levels of risk aversion.

Risk is generally defined as risk of loss, and often measured using the variance of returns on the asset over an appropriate time horizon. Other measures of risk exist, including Value-at-Risk (VaR) and conditional VaR (CVaR, also called expected shortfall).

2.2 Bayesian networks

2.2.1 Computational Methods

2.2.1.1 Introduction

This research aims to apply computational methods to improve the asset management process. Computational methods can be broadly grouped into three fields:

- Computational Statistics
- Artificial Intelligence
- Complex Systems

Computational statistics refers to computationally intensive statistical methods which require optimized algorithms to solve otherwise intractable problems. Examples include bootstrap and Monte Carlo simulation. Markov Chain Monte Carlo is discussed in detail in Section 2.2.6.

Many definitions of artificial intelligence exist, including "the study of how to make computers do things at which, at the moment, people are better" (Rich and Knight, 2009). Subfields in artificial intelligence include knowledge-based AI where an expert encodes knowledge in a model; logic-based AI which uses logical rules for representing knowledge and solving problems; evolutionary algorithms; and machine learning. Subfields of machine learning include unsupervised learning where the goal is to learn patterns in data without additional explicit information; supervised learning, to learn a function f^* which approximates a true function f where $\boldsymbol{y} = f(\boldsymbol{x})$, given a training set comprised of values for vectors $\boldsymbol{x}$ and $\boldsymbol{y}$; and reinforcement learning where the goal is to maximize total net rewards for actions taken. Machine learning is further discussed in section 2.2.1.2.

Complex systems refers to "... an ensemble of many elements which are interacting in a disordered way, resulting in robust organisation and memory" (Ladyman, Lambert, and Wiesner, 2013). Examples of approaches to work with complex systems include agent-based systems.

The use of computational methods in asset management is an active area of research.

2.2.1.2 Machine learning

This research is mostly concerned with supervised machine learning using data.

Some machine learning algorithms such as logistic regression, clustering, linear discriminant analysis, and support vector machines are used to classify an input space into a limited number of classes. The output of these algorithms is in a sense discrete, while the output desired in this research are values for expected returns and variances, which are assumed to be continuous values. Many machine learning algorithms can be considered (Russell and Norvig, 2010).

Given input values for $\boldsymbol{x}$ and $\boldsymbol{y}$ **linear regression** learns the parameters $\boldsymbol{a}$ of a linear regression function $\boldsymbol{y} = \boldsymbol{a}\boldsymbol{x} + \boldsymbol{\epsilon}$ by minimizing an error function. Error functions include ordinary least squares, which selects those parameters $\boldsymbol{a}$ which minimize the function $S = \sum_i (y_i - a_i x_i)^2$. The approach can be generalized by modifying the regression function and the error function. The EGARCH Eqs. (2.38) and (2.39) can be considered a more complex type of regression.

Artificial neural networks are collections of units which are loosely based on the concept of biological neurons. Each unit, when receiving an input from predecessors, may process it and send further inputs to other units connected to it. These networks may be used in both supervised and unsupervised learning, and are extremely powerful as they may learn any arbitrary function. One drawback of this class of algorithms is that they function as black boxes: it is not clear how the results are obtained. Another is the potential for overfitting.

Random forests are a collection of uncorrelated decision trees. The individual decision trees are created during the training phase. The outputs of the model are calculated based on an average value of the results of each tree. An interesting aspect of this algorithm is its resistance to overfitting, however they may be complex to analyze.

Probabilistic graphical models, including Markov and Bayesian networks, use a graph to represent a distribution over a multi-dimensional space by encoding the conditional dependence between random variables. Markov models are undirected, potentially cyclic, graphs, while Bayesian Networks are directed acyclic graphs. The two models are closely related and both can be used to represent a given distribution. Bayesian networks are discussed in section 2.2.

2.2.1.3 Computational Methods in Asset Management

As discussed in Section 2.3, the general goal of asset management is the creation of an optimal portfolio. Computational methods may be used in different ways to attain this overall goal.

A number of researchers have attempted to address the concerns with mean-variance optimization discussed in Section 2.3.3 by adding constraints of various types (Metaxiotis and Liagkouras, 2012). The problem, however, then becomes difficult to solve using traditional methods. Deng, Lin, and Lo (2012), Golmakani and Fazel (2011), Zhang et al. (2010), and Zhu et al. (2011) propose particle swarm optimization algorithms to obtain the optimal Mean-Variance portfolio when a broader range of constraints is imposed, including limiting turnover or selecting portfolios with few active positions in order to reduce transaction costs. Chen et al. (2014) explore the use of an artificial bee colony algorithm to select optimal portfolios assuming returns are fuzzy numbers and in the presence of transaction costs. Doerner et al. (2006) and Sefiane and Benbouzian (2013) apply ant colony algorithms to the optimization process. Metaxiotis and Liagkouras (2012) provide a review of a broad range of approaches that have been explored.

These papers only seek to improve the optimization process, without incorporating additional insights investors may have. This thesis investigates the use of computational methods for this later process.

2.2.2 Bayesian Networks

A Bayesian network is a graphical model which represents the conditional independence of random variables via a directed acyclic graph. The nodes of the graph are random variables - such as inflation expectations, industrial production, or returns on assets. The variables are also referred to as factors. The edges represent an assumption of conditional dependence between the random variables - for example returns on assets may be conditionally dependent on inflation expectations. An *absence* of an edge expresses an assumption of conditional *independence*.

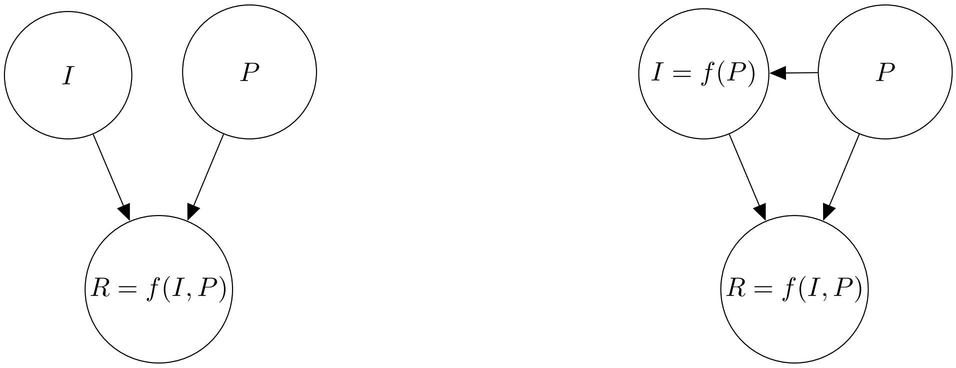
The use of a Bayesian network in this thesis is partly motivated by a need to quantify the uncertainty of predictions. As discussed in Section 2.3.3, the Black-Litterman model requires that the portfolio manager provide views for the assets in the portfolio. Views are defined as a prediction of expected returns, and an estimate of the uncertainty of this prediction. Bayesian network can provide results as random variables with a probability distribution. The mean and variance of the distribution can be used as the prediction and the measure of uncertainty.

An additional motivation for the use of Bayesian networks is the possibility to introduce expert knowledge in the form of prior expectations.

Inference is the process of answering questions about the values of unobserved variables in the graph, given data. As discussed later, exact inference in Bayesian networks is NP-hard. Although even approximate inference is, in principle, NP-hard, approaches such as Markov Chain Monte Carlo (MCMC) can often be used to perform approximate inference.

To describe a Bayesian network, two things are required: the topology of the graph, and the relationship between the factors in the graph. If the factors take on discrete values, conditional probability distributions may be used. Otherwise the functional form of the relationship must be described and the parameters of this function become the variables of interest, the values of which can be learned from data.

It is incorrect though sometimes intuitively useful to think of the graph in terms of causality: changes in inflation expectations “cause” changes in asset returns. This is incorrect because a Bayesian network can be created which has an edge pointing in the other direction but that still correctly represents the conditional independence between these variables. Nevertheless,



(a) Naive Bayes: Returns as a function of inflation and production

(b) Dependent factors: Consider inflation as a function of production

Figure 2.1: A naive Bayesian network, where all factors are assumed to be conditionally independent, contrasted with a network with factors which are conditionally dependent. Including conditional dependencies in a graph can be useful for scenario analysis.

particularly when interpreting the results of the Bayesian network, it is useful to construct the network in a way which is plausible from the point of view of causality.

2.2.3 Motivation

Bayesian networks may be well suited to help address some criticisms of the Black-Litterman model discussed in Section 2.3.3. First, equity returns are treated as random variables with a probability distribution. Predictions can therefore be made and include both an expected return and a measure of uncertainty such as variance. Second, distributions other than the normal distribution may easily be used. Third, models which take into account observed volatility clustering may be used to describe the relationship between factors in the Bayesian network.

Standard models lack transparency, are difficult to challenge, and are computationally inefficient due to the fact that the entire joint probability distribution between all factors must be calculated. Bayesian networks are a natural solution to these issues. They are tractable, intuitive, can be easily extended, and allow the incorporation of expert knowledge.

2.2.4 Model

In a naive Bayesian network[1] as in Figure 2.1a the factors themselves are assumed to be conditionally independent of each other. This simple model can be useful. Figure 2.1b presents a slightly more complicated network where inflation is assumed to be conditionally dependent on industrial production. Where both production and inflation are always observed, this conditional dependence is of no importance. However where either inflation or production are unknown, or if a simulation of an increase in inflation or production is desired, knowledge of the other factor can be useful. In either case, the parameters of the functions relating different factors can be inferred from data.

[1] Notation for drawing the Bayesian networks is from (Dietz, 2010).

2.2.5 Inference of GARCH parameters

Consider the case where returns follow a process defined as

$$r_t = \varepsilon_t \tag{2.1}$$

$$\sigma_t^2 = \sum_{i=1}^{n} \theta_i f_i + \theta_{n+1}\sigma_{t-1}^2 + \theta_{n+2}\varepsilon_{t-1}^2 \tag{2.2}$$

$$\varepsilon_t = \sigma_t \epsilon_t, \tag{2.3}$$

where the f_i's are exogenous factors and $\boldsymbol{\theta}$ are paramters. This is a GARCH process with n exogenous regressors. The vector $\boldsymbol{\theta}$ is a random variable with distribution $p(\boldsymbol{\theta})$. The mean and standard deviation of the components of $\boldsymbol{\theta}$ are to be inferred from data $\boldsymbol{x}$ using Bayesian inference. This question has been studied extensively (Anyfantaki and Demos, 2012, 2016; Bauwens and Lubrano, 1998; Cuervo, Achcar, and Barossi-Filho, 2014; Takaishi, 2006, 2013; Vrontos, Dellaportas, and Politis, 2000).

Bayesian inference begins with Bayes' rule. In this case this is

$$p(\boldsymbol{\theta}|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\boldsymbol{x})}, \tag{2.4}$$

where $p(\boldsymbol{\theta}|\boldsymbol{x})$, the posterior, is the value of the parameters $\boldsymbol{\theta}$ given the data $\boldsymbol{x}$. $p(\boldsymbol{x}|\boldsymbol{\theta})$ is called the likelihood, $p(\boldsymbol{\theta})$ the prior, and $p(\boldsymbol{x})$ the marginal likelihood or the evidence. The joint distribution of $\boldsymbol{\theta}$ and $\boldsymbol{x}$ is

$$p(\boldsymbol{\theta}, \boldsymbol{x}) = p(\boldsymbol{x}|\boldsymbol{\theta})p(\boldsymbol{\theta}) \tag{2.5}$$

marginalising

$$p(\boldsymbol{x}) = \int_{\boldsymbol{\theta}} p(\boldsymbol{x}|\boldsymbol{\theta})p(\boldsymbol{\theta})d\boldsymbol{\theta}. \tag{2.6}$$

In general, $\boldsymbol{\theta}$ is multidimensional. In these cases, $p(\boldsymbol{\theta})$ is likely to be intractable, and the calculation of this integral is NP hard (Cuervo, Achcar, and Barossi-Filho, 2014; Takaishi, 2013). Various approaches can be used to perform approximate inference, including Expectation Maximization, Variational Bayes, and Markov chain simulation (Markov Chain Monte Carlo or MCMC) (Gelman et al., 2014). The approach used in this thesis is MCMC.

2.2.6 Markov Chain Monte Carlo

Markov Chain Monte Carlo creates a large number of samples to approximate a target distribution $p(\boldsymbol{\theta}|\boldsymbol{x})$. It does this by drawing candidate values for $\boldsymbol{\theta}$ from an approximate probability distribution, and accepting or rejecting the candidate based upon some measure of fit with the target probability distribution in order to create a Markov chain, where a Markov chain is defined as a sequence of random variables $(a_1, a_2, ...a_n)$ where, for all

$i > 1$, the distribution of a_i only depends on a_{i-1}. Inference can then be performed using the Markov chain represented by the accepted samples.

It is typically not necessary to draw samples from the entire parameter space. Probability mass is often concentrated in a smaller part of parameter space called the "typical set" (Andrew, 2004). The objective is therefore to find the typical set and search it efficiently. Several algorithms exist to do this, including random walk metropolis (Metropolis et al., 1953), Gibbs sampling (Geman and Geman, 1984), Hamiltonian Monte Carlo (HMC) (Duane et al., 1987; Neal, 2011), and the No U-Turn Sampler (NUTS) (Hoffman and Gelman, 2014). These algorithms differ in the way they generate candidate values for $\boldsymbol{\theta}$ and in the way they accept or reject candidate solutions.

NUTS, which is an enhancement of HMC, is used in this thesis. HMC introduces an auxiliary variable $\boldsymbol{r}$. In a fictitious Hamiltonian system, $\boldsymbol{\theta}$ corresponds to the position of particles, and $\boldsymbol{r}$ to their momentum. The auxiliary variables are drawn from a random distribution, yielding the joint density

$$p(\boldsymbol{\theta}, \boldsymbol{r}) \propto \exp\left(\mathcal{L}(\boldsymbol{\theta}) - \frac{1}{2}\boldsymbol{r} \cdot \boldsymbol{r}\right), \tag{2.7}$$

in which, assuming normally distributed returns, the log-likelihood function $\mathcal{L}(\boldsymbol{\theta})$ is, apart from a constant term, (Bollerslev (1986))

$$\mathcal{L}(\boldsymbol{\theta}) = -\frac{1}{2N}\sum_{n=1}^{N}\left(\log \sigma_n^2 + \frac{\epsilon_n^2}{\sigma_n^2}\right). \tag{2.8}$$

HMC searches the typical set by providing a sequence of M candidate values for $\boldsymbol{\theta}$. Given a number of steps L, a step-size δ, setting the starting value of $\tilde{\boldsymbol{\theta}}$ to the present value of $\boldsymbol{\theta}_{m-1}$ and starting value of $\tilde{\boldsymbol{r}}$ to the value of $\boldsymbol{r}_0$ drawn from the multivariate normal distribution, the candidates are produced by running the **leap-frog** Algorithm (1) L times (Hoffman and Gelman, 2014)

Algorithm 1 Leap-frog algorithm

procedure LEAP-FROG($\tilde{\boldsymbol{\theta}}, \tilde{\boldsymbol{r}}, \delta$)
  $\tilde{\boldsymbol{r}} \leftarrow \tilde{\boldsymbol{r}} + (\delta/2)\nabla_{\boldsymbol{\theta}}\mathcal{L}(\tilde{\boldsymbol{\theta}})$
  $\tilde{\boldsymbol{\theta}} \leftarrow \tilde{\boldsymbol{\theta}} + \delta\tilde{\boldsymbol{r}}$
  $\tilde{\boldsymbol{r}} \leftarrow \tilde{\boldsymbol{r}} + (\delta/2)\nabla_{\boldsymbol{\theta}}\mathcal{L}(\tilde{\boldsymbol{\theta}})$,
  return $\tilde{\boldsymbol{\theta}}, \tilde{\boldsymbol{r}}$
end procedure

where $\nabla_{\boldsymbol{\theta}}$ is the gradient with respect to $\boldsymbol{\theta}$. The algorithm produces a new proposed draw comprised of a pair of vectors $\tilde{\boldsymbol{\theta}}_{\boldsymbol{m}+1}$ and $\tilde{\boldsymbol{r}}_{\boldsymbol{m}+1}$. It accepts this draw with probability equal to

$$\alpha = \min\left(1, \frac{\exp(\mathcal{L}(\tilde{\boldsymbol{\theta}}) - \frac{1}{2}\tilde{\boldsymbol{r}} \cdot \tilde{\boldsymbol{r}})}{\exp(\mathcal{L}(\boldsymbol{\theta}_{m-1}) - \frac{1}{2}\boldsymbol{r}_0 \cdot \boldsymbol{r}_0)}\right). \tag{2.9}$$

The algorithm continues until M samples have been accepted.

2.2.7 Summary

This section presented Bayesian networks. It presented the motivation for the use of Bayesian networks to generate views for a Black-Litterman model discussed in Section 2.3.3, and how a GARCH model can be used in this approach. It discussed inference of the GARCH parameters using Markov Chain Monte Carlo, and the No U-Turn Sampler used in this thesis.

2.3 Asset Allocation Models

This section presents two asset allocation models: mean-variance optimization pioneered by Harry Markowitz, and the Black-Litterman model proposed by Fischer Black and Robert Litterman.

2.3.1 Mean-Variance Optimization

2.3.1.1 Model

Markowitz defines a **mean-variance optimal portfolio** as a portfolio of investment securities that presents both (Markowitz, 1952, 1955):

1. Maximum expected return for a given level of expected risk.
2. Minimum expected risk for a given level of expected return.

Consider a portfolio composed of N risky assets. Let $\boldsymbol{\mu}$ be the vector of the mean return, in excess of the risk free rate, of each asset in the portfolio, $\boldsymbol{w}$ the vector of the weight of each asset in the portfolio, and $\boldsymbol{\Sigma}$ the $N \times N$ covariance matrix of asset returns.

The expected return μ_p and variance σ_p^2 of this portfolio are

$$\mu_p = \boldsymbol{w}^{\mathsf{T}}\boldsymbol{\mu} \tag{2.10}$$

$$\sigma_p^2 = \boldsymbol{w}^{\mathsf{T}}\boldsymbol{\Sigma}\boldsymbol{w}. \tag{2.11}$$

For an investor with a required rate of return in excess of the risk free rate of μ_0, this portfolio is a mean-variance optimal portfolio if it meets the condition (Fabozzi, Focardi, and Kolm, 2006, p. 23)

$$\min \boldsymbol{w}^\intercal \boldsymbol{\Sigma} \boldsymbol{w}, \tag{2.12}$$

subject to the constraints

$$\mu_0 = \boldsymbol{w}^\intercal \boldsymbol{\mu} \tag{2.13}$$

$$\sum_i w_i = 1. \tag{2.14}$$

Let $\boldsymbol{I}^\intercal$ be a $1 \times N$ matrix of ones, $[1, 1, \ldots, 1]$. Markowitz (1955) provides a detailed solution to this optimization problem which can be expressed as (Fabozzi, Focardi, and Kolm, 2006, p. 24)

$$\boldsymbol{w} = \boldsymbol{g} + \boldsymbol{h}\mu_0 \tag{2.15}$$

$$\boldsymbol{g} = \frac{1}{ac - b^2} \boldsymbol{\Sigma}^{-1}(c\boldsymbol{I} - b\boldsymbol{\mu}) \tag{2.16}$$

$$\boldsymbol{h} = \frac{1}{ac - b^2} \boldsymbol{\Sigma}^{-1}(a\boldsymbol{\mu} - b\boldsymbol{I}) \tag{2.17}$$

$$a = \boldsymbol{I}^\intercal \boldsymbol{\Sigma}^{-1} \boldsymbol{I} \tag{2.18}$$

$$b = \boldsymbol{I}^\intercal \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu} \tag{2.19}$$

$$c = \boldsymbol{\mu}^\intercal \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}. \tag{2.20}$$

The weights $\boldsymbol{w}$ are the output of the mean-variance optimization. The covariance matrix $\boldsymbol{\Sigma}$ can be estimated based upon the covariance of historical returns (Satchell and Scowcroft, 2000). Black and Litterman (1992) and He and Litterman (1999) explore various methods of estimating $\boldsymbol{\mu}$, including using historical average returns, equal means, risk-adjusted equal means, and an equilibrium approach discussed in Section 2.3.3. This thesis follows the later equilibrium approach when implementing mean-variance optimization.

The set of portfolios with optimal weights $\boldsymbol{w}^*$, meeting these conditions for all attainable returns μ_0, is called the **efficient set**. The optimal weights including the constraint $w_i > 0$, $\forall w_i$ can be obtained in python by convex optimization using the *cvxopt* package.

2.3.1.2 Critique

This section presents a critique of the algorithm, focusing on issues in its application.

Given return and variance assumptions, the portfolios in the efficient set promise the best return for a given level of risk. They are the first potential benchmarks for portfolio managers (Bevan and Winkelmann, 1998, p. 10). There may, however, be reasons to digress from the proposed weights $\boldsymbol{w}^*$ in constructing an investment portfolio.

Optimal portfolio weights are very sensitive to the return assumptions used (Black and Litterman, 1992). A small increase in the expected returns of one asset was found in some cases to set the weights of half the securities in the portfolio to zero, without impacting either portfolio expected return nor variance (Best and Grauer, 1991). Michaud (1989) states that this instability is in part due to ill-conditioning of the covariance matrix.

When no constraints are imposed, optimal weights often include significant short positions. When constraints on short positions are imposed, the result is often allocations of zero weights to most assets - which Black and Litterman (1992) and Firoozye and Blamont (2003) call "corner solutions".

Factors such as liquidity or market capitalizations are not taken into consideration (Black and Litterman, 1992; Michaud, 1989). Allocated weights may be so large that investors would be acquiring a significant percentage of the outstanding shares of a company. It may be impossible to perform these trades without impacting the transaction price.

Return and variance estimates must be given for all assets. However, the investor may not be an expert on all asset classes, and may have a view only on a subset of all assets in the portfolio. In this case the investor would provide a view where one is held, and default assumptions for all other assets. The model does not allow the manager to distinguish between strongly held views and simple assumptions. As it treats both equally, the result may be a portfolio which does not reflect the investor's views (Black and Litterman, 1992; Firoozye and Blamont, 2003). Michaud (1989) notes that this problem extends to inherently different levels of uncertainty related to different asset classes.

2.3.2 Enhancements to Mean-Variance Optimization

Many of the criticisms of the mean-variance optimization model are due to errors in the estimation of inputs (Harris, Stoja, and Tan, 2017). Harris, Stoja, and Tan (2017) and Lejeune (2011) survey a number of approaches which have been investigated to address this issue, including Bayesian (Black and Litterman, 1992; Jorion, 1991), Robust Optimization (Huang et al., 2010; Stinstra and Hertog, 2008; Tütüncü and Koenig, 2004), and Stochastic Programming (Bonami and Lejeune, 2009). In this thesis, we focus on the Black-Litterman model as it has become one of the most widely used approaches in practice, and it resolves many of the issues discussed above.

2.3.3 The Black-Litterman Asset Allocation Model

2.3.3.1 Introduction

The Black-Litterman model (Black and Litterman, 1992) was proposed in order to address some of the perceived problems with mean-variance optimization. Combining mean-variance optimization with the capital asset pricing model (CAPM) (Lintner, 1965; Sharpe, 1964), it allows investors to express views only on certain assets, to express different levels of uncertainty for each view, and integrates these views with default return and covariance parameters for all assets.

This section presents the core elements of the model which are necessary for implementation.

2.3.3.2 Model

According to Black and Litterman (Black, 1993b; Black and Litterman, 1992), using historical data to calculate expected returns is unreliable. The Black-Litterman approach is therefore based on theory. The formula for equilibrium excess returns, $\mathbf{\Pi}$, follows from the CAPM, using the market capitalizations of assets for $\boldsymbol{w}$, the equilibrium market weights (He and Litterman, 1999; Idzorek, 2007; Satchell and Scowcroft, 2000)

$$\mathbf{\Pi} = \delta \mathbf{\Sigma} \boldsymbol{w}, \tag{2.21}$$

where δ is the investor's risk aversion.

Assuming the expected return of the market portfolio $E(r)$ is normally distributed with mean μ_m and variance σ_m^2 so that

$$E(r) \sim (\mu_m, \sigma_m^2) \tag{2.22}$$

then the risk aversion parameter δ is defined as (Satchell and Scowcroft, 2000, p. 139)

$$\delta = \frac{\mu_m}{\sigma_m^2}. \tag{2.23}$$

We then formulate a set of views on the market. Following Idzorek (2007), a view i is expressed as a set of weights $\{p_{i1}, \cdots, p_{in}\}$, an expected return q_i for a portfolio with these weights, and a variance ω_i. For a set of k views, we define

$$\boldsymbol{P} = \begin{bmatrix} p_{11} & \cdots & p_{1n} \\ \vdots & \ddots & \vdots \\ p_{k1} & \cdots & p_{kn} \end{bmatrix} \tag{2.24}$$

$$\boldsymbol{Q} = \begin{bmatrix} q_1 \\ \vdots \\ q_k \end{bmatrix} \tag{2.25}$$

$$\boldsymbol{\Omega} = \begin{bmatrix} \omega_1 & 0 & \cdots & 0 & 0 \\ 0 & \omega_2 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \omega_{k-1} & 0 \\ 0 & 0 & \cdots & 0 & \omega_k \end{bmatrix}. \tag{2.26}$$

The views are required to be independent. They are therefore uncorrelated, and thus $\boldsymbol{\Omega}$ has zero weights on all non-diagonal elements. Idzorek states that the $\omega's$ may, by default, be set to the elements of the $k \times 1$ vector $\boldsymbol{P}(\tau\boldsymbol{\Sigma})\boldsymbol{P}^\intercal$, where τ is a measure of the uncertainty of the prior covariance matrix, $\boldsymbol{\Sigma}$. In this case

$$\boldsymbol{\Omega} = \begin{bmatrix} \tau(\boldsymbol{p}_1\boldsymbol{\Sigma}\boldsymbol{p}_1^\intercal) & 0 & \cdots & 0 & 0 \\ 0 & \tau(\boldsymbol{p}_2\boldsymbol{\Sigma}\boldsymbol{p}_2^\intercal) & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \tau(\boldsymbol{p}_{k-1}\boldsymbol{\Sigma}\boldsymbol{p}_{k-1}^\intercal) & 0 \\ 0 & 0 & \cdots & 0 & \tau(\boldsymbol{p}_k\boldsymbol{\Sigma}\boldsymbol{p}_k^\intercal) \end{bmatrix}. \tag{2.27}$$

The diagonal elements of $\boldsymbol{\Omega}$ can also be set to the variance of the views if these are available.

In implementation, care must be taken to formulate equations in such as way as to take into consideration that $\boldsymbol{P}$ and $\boldsymbol{\Omega}$ may not be invertible.

Mean returns $\boldsymbol{\mu}$, the posterior variance $\boldsymbol{M}$, and the sample variance $\boldsymbol{\Sigma}_p$ combining market equilibrium conditions and investor views are given by (Kolm and Ritter, 2017; Walters, 2008)

$$\boldsymbol{\mu} = \boldsymbol{\Pi} + \tau\boldsymbol{\Sigma}\boldsymbol{P}^\intercal[(\boldsymbol{P}\tau\boldsymbol{\Sigma}\boldsymbol{P}^\intercal) + \boldsymbol{\Omega}]^{-1}[\boldsymbol{Q} - \boldsymbol{P}\boldsymbol{\Pi}] \tag{2.28}$$

$$\boldsymbol{M} = ((\tau\boldsymbol{\Sigma})^{-1} + \boldsymbol{P}^\intercal\boldsymbol{\Omega}\boldsymbol{P})^{-1} \tag{2.29}$$

$$\boldsymbol{\Sigma}_p = \boldsymbol{\Sigma} + \boldsymbol{M}. \tag{2.30}$$

We can rearrange Eq. (2.21) to solve for portfolio weights given a set of returns $\mathbf{\Pi}$ and the sample variance matrix $\mathbf{\Sigma}_p$ in the absence of constraints (He and Litterman, 1999; Mankert, 2006; Walters, 2008)

$$\boldsymbol{w}^* = \mathbf{\Pi}(\delta\mathbf{\Sigma})^{-1}. \tag{2.31}$$

Inverting $\mathbf{\Sigma}$ may be avoided by rewriting (2.31) and solving for $\boldsymbol{w}^*$ in

$$(\delta\mathbf{\Sigma})\boldsymbol{w}^* = \mathbf{\Pi}. \tag{2.32}$$

2.3.3.3 Critique

The Black-Litterman model assumes that the correlation matrix is stable over time (Harris, Stoja, and Tan, 2017). For example, He and Litterman (1999) use 20 years of historical data to estimate $\mathbf{\Sigma}$ in their implementation. The model also assumes that returns are normally distributed (Harris, Stoja, and Tan, 2017). In fact, there is evidence that returns are not normally distributed, return distributions may not be stable, ergodicity may not hold, tails are not symmetrical, correlations increase when markets fall, and volatility exhibits clustering (Cont, 2001; Rebonato and Denev, 2012). Finally the construction of the views required by the model is itself not trivial.

The model has been extended using a variety of approaches, including some based on machine learning and artificial intelligence.

2.3.4 Enhancements to the Black-Litterman Model

Fabozzi and Kolm (2007) and Kolm, Tütüncü, and Fabozzi (2014) survey recent enhancements to the canonical Black-Litterman model. Some of these are the result of modern advances in machine learning and computing power, as well as the availability of large amounts of data, which now make it feasible to use more complex algorithms than have been used in the past.

The canonical model assumes that returns are normally distributed, using variance as a risk measure (Giacometti et al., 2007). As there is evidence that returns are not normally distributed (Cont, 2001; Fama, 1965b; Mandelbrot, 1963), there has been interest in relaxing these assumptions. One approach, for which fully commented code is available online, will be discussed in more detail.

Meucci (2006a,b) developed a methodology to extend the Black-Litterman model to non-normally distributed markets and views using copula-opinion pooling (COP). Their approach also allows investors to express their views in different ways, for example using a uniform distribution. The COP approach defines an N-dimensional variable $\boldsymbol{M}$ which can represent any set of random variables for factors that can characterize the noise in market returns, including market returns. The distribution of this variable is the prior which is defined "by means of sophisticated proprietary models" (Meucci, 2006a). Stress-testing of correlations and non-linear views are not permitted by the model. While the model may be applicable more broadly than the canonical model, no evidence is provided that the portfolios generated using this approach result in higher returns than those generated by the canonical model.

Meucci (2008) proposes a "fully flexible" alternative model to incorporate general views in non-normal markets using an entropy pooling approach. As in their previous work, the approach begins with an N-dimensional vector of factors $\boldsymbol{X}$ which drive the market. Views are defined as generic functions, which are not necessarily linear. Weights are defined using an optimization function with a cost function they call a "subjective index of satisfaction" defined using a number of alternative approaches including Value at Risk, a utility function, or a spectral risk measure, all subject to investment constraints. The posterior is determined by minimizing the relative entropy between the market distribution with and without views

$$\boldsymbol{X} \sim f_{\boldsymbol{x}} \tag{2.33}$$

$$\boldsymbol{w}^{*} \equiv \underset{\boldsymbol{w}\in\mathcal{C}}{\operatorname{argmax}} \left(\mathrm{S}(\boldsymbol{w}; f_{\boldsymbol{x}})\right) \tag{2.34}$$

$$\boldsymbol{V} \equiv \mathrm{g}(X) \sim \tilde{f}_{\boldsymbol{v}} \tag{2.35}$$

$$\mathcal{E}(\tilde{f}_{\boldsymbol{x}}, f_{\boldsymbol{x}}) \equiv \int \tilde{f}_{\boldsymbol{x}}(\boldsymbol{x}) \left(\ln \tilde{f}_{\boldsymbol{x}}(\boldsymbol{x}) + \ln f_{\boldsymbol{x}}(\boldsymbol{x})\right) d\boldsymbol{x} \tag{2.36}$$

$$\tilde{f}_{\boldsymbol{x}} \equiv \underset{(f\in V)}{\operatorname{argmin}} \left(\mathcal{E}(f, f_{\boldsymbol{x}})\right), \tag{2.37}$$

where f is a general probability density function, S is a "subjective index of satisfaction", $\boldsymbol{w}$ are asset weights, $\mathcal{C}$ are a set of constraints, V are views expressed as as functions of the market $\boldsymbol{X}$, $f \in V$ represents all distributions consistent with the views. Except in special cases, the solution to the minimization problem must be found numerically. While this approach is again more broadly applicable than the canonical Black-Litterman model as it can, in principle, address a portfolio including asset classes such as options, no empirical evidence regarding the performance of portfolios generated using this approach is presented.

(Lejeune, 2011) derived a model using value at risk, again without empirically testing the proposed model.

Giacometti et al. (2007) applied normal, t-student, and stable distributions for returns, and value at risk, conditional value at risk, and dispersion-based risk measures to estimate equilibrium returns. In all, ten models are used to produce return estimates. They compared these return estimates to actual returns in the next period, and found that an α-stable return distribution combined with dispersion-based risk measures provided the best forecast. While they indicate that their forecasts were then used to produce a portfolio, they do not report the performance of these portfolios. It is therefore unclear whether their improved forecasts resulted in improved performance.

The Black-Litterman model is the subject of considerable research on aspects beyond the use of non-normal return distributions. For example, Mankert (2006) explores links with behavioral finance; Beach and Orlov (2007) use GARCH-derived views as inputs to the model; Cooper, Molyboga, and Molyboga (2016) integrate the model with exotic beta; and Creamer (2015) bases views on news sentiment analysis and high-frequency data. Almgren and Chriss (2006) replace expected returns with a sorting criteria. Qian and Gorman (2001) extend Black-Litterman to expressing views on volatility and correlations. Zhou (2009) includes information contained in the data-generating process. Meucci (2009) considers how to modify the risk factors underlying the market, rather than just the returns themselves. Kolm and Ritter (2017) generalize the Black-Litterman model and provide views on factor risk premia. Avramov and Zhou (2010) reviews a number of recent studies, including some that question the assumption that returns are independent and identically distributed and others that explore regime change and stochastic volatility.

The objective of Beach and Orlov (2007) is similar to one of the objectives of this research as they seek to generate a proxy for user views as input to a Black-Litterman model. Their approach uses a EGARCH-M(1,1) model (Nelson (1991)) with regressors z_1 and z_2. They describe their model as

$$y_t = \boldsymbol{x}_t^\intercal \boldsymbol{\upsilon} + \delta \hat{\sigma}_t^2 + \boldsymbol{\psi} \boldsymbol{z}_{1t} + \varepsilon_t \tag{2.38}$$

$$\log(\sigma_t^2) = \omega + \beta \ \log(\sigma_{t-1}^2) + \alpha \left| \frac{\varepsilon_{t-1}}{\sigma_{t-1}} \right| + \gamma \frac{\varepsilon_{t-1}}{\sigma_{t-1}} + \boldsymbol{\varphi} \boldsymbol{z}_{2t}, \tag{2.39}$$

where y_t is the asset return, $\boldsymbol{x}^\intercal$ is a vector of historical returns, $\varepsilon_t = \sigma_t \eta_t$, $\eta_t \sim \mathcal{N}(0, 1)$, $\hat{\sigma}_t$ is the variance estimated using EGARCH, $\alpha, \beta, \gamma, \delta, \omega$ are real numbers, and $\boldsymbol{\upsilon}, \boldsymbol{\varphi}, \boldsymbol{\psi}$ are vectors.

The justification for the use of this model is that market volatility is higher when the market falls then when it increases by the same amount. This asymmetry can be represented in EGARCH, (Eq. (2.39)).

ARCH-M models the dependence between return and risk by including the conditional variance in the mean equation (Eq. (2.38)).

Focusing on allocation to 20 countries, they selected regressors based on macroeconomic factors such as the growth in industrial production, inflation, the return on the USD index relative to major currencies, the difference in the yield on BAA and AAA bond indices, the difference in the three-month Eurodollar yield and the three-month treasury bill yield, and the percentage change in the world spot price of oil.

Constructing portfolios using an iterative process which modified τ in Eq. (2.28) and (2.29) to keep portfolio risk within certain limits, they report positive results for this "risk-reduced portfolio".

For reasons discussed in Section 3.2.2, these positive results may be due in part to the use, in back testing, of information which was not available at the time. If so, then their results may be unreliable.

This body of research indicates the existing broad interest in improving the way the Black-Litterman model is implemented. This model requires that the user define views for some or all assets in the portfolio. Views are defined as a prediction of expected returns, and an estimate of the uncertainty of this prediction. As discussed in Section 2.2, a Bayesian network can provide results as random variables with a probability distribution. Implicit in a probability distribution is the notion that the result is uncertain. Bayesian networks can be used to quantify the uncertainty of their predictions, and are therefore well-suited to this requirement. This thesis contributes to research on the use of the Black-Litterman model and focuses on determining whether Bayesian networks may be useful in generating views for this model. For this purpose, a canonical Black-Litterman model is used.

2.3.5 Asset Allocation and the Efficient Market Hypothesis

Fama (1965a) discussed the notion of efficient markets as one where securities prices fully reflect all available information. Fama (1970) introduced different types of market efficiency: the **strong** form which assumes market prices reflect all available information, including insider information, the **semi-strong** form where market prices are assumed to instantly reflect the value of new publicly available information, and the **weak form** where market prices are assumed to reflect all past publicly available information - allowing time for the market to react to new information. Fama later provided the following summary :

“Market efficiency means that deviations from equilibrium expected returns are unpredictable based on currently available information. But equilibrium expected returns can vary through time in a predictable way. (Fama and Litterman, 2012)”

One can view the efficient market hypothesis (EMH) as presenting markets as a multi-agent system comprised of intelligent agents acting with local rules, whose actions have as an effect that all information is reflected in market prices. This is important, since if at least the semi-strong form of the efficient market hypothesis holds, investors cannot forecast future returns and one might believe they can do no better than to hold the market portfolio. Nevertheless, Fama himself states “Of all the potential embarrassments to market efficiency, momentum is the primary one. (Fama and Litterman, 2012)” suggesting that the hypothesis may not hold in all cases. There are several reasons one may wish to attempt to forecast future returns, in spite of evidence in support of the EMF.

First, some have found evidence that the efficient market hypothesis does not hold in all cases. Rosenberg, Reid, and Lanstein (1985) found that stocks with a low price/book ratio tend to outperform the market.

Second, mean-variance optimization allows investors to find optimal portfolios. However, it requires both an estimate of future returns and future covariances for all assets in the portfolio.

Third, the weak form of the EMH allows for the possibility that markets react gradually to new information. Computational intelligence and machine learning techniques may, by aggregating large amounts of data quickly and efficiently, establish new relationships between apparently unrelated data which, in effect, creates new information.

In this sense, this thesis contributes to the body of knowledge testing the efficient market hypothesis.

2.3.6 Summary

This section discussed asset allocation models including mean-variance optimization and the Black-Litterman asset allocation model. It presented various approaches which have been explored to enhance the models, demonstrating the existing broad interest in improving the way the Black-Litterman model is implemented.

The Black-Litterman model requires that the user provides views for each asset in the portfolio. Views are defined as a prediction of expected returns, and an estimate of the uncertainty of this prediction. As discussed in Section 2.2, Bayesian networks are well-suited to this type of requirement. This thesis therefore contributes to existing research on the use of the Black-Litterman model by exploring how Bayesian networks may be useful in generating views for a Black-Litterman model.

This section also discussed the objectives of this thesis in the context of the efficient market hypothesis.

2.4 Risk

Risk measurement, a related field, is also the subject of a number of research papers, and may provide guidance on potential areas to investigate in algorithmic asset management. Solutions which are currently being investigated in this area include building upon standard approaches (Di Bernardino et al., 2015), agent-based models (LeBaron, 2006), and probabilistic graphical models (Denev, 2011; Kwiatkowski and Rebonato, 2011; Rebonato and Denev, 2014; Rebonato and Denev, 2012).

2.5 Algorithmic Asset Management

Algorithmic asset management can be defined as the process of automating and enhancing the entire asset management value chain, from suitability testing to portfolio construction and client reporting. Algorithmic trading, defined as a method of automatically executing a large trade in such a way as to limit cost, market impact, and risk, can be considered a part of algorithmic asset management. This thesis focuses on one key part of the algorithmic asset management process: intelligent portfolio construction.

2.5.1 Robo-advisors

Robo-advisors can be defined as "algorithms to automatically allocate, manage and optimize clients'" portfolios.[2] According to the investment magazine Barron's, Betterment pioneered robo-advisory in 2010. Vanguard PAS and Schwabb Intelligent Portfolios, which started in 2015, manage \$83 billion and \$19 billion repectively. Goldmans Sachs, JP Morgan, and Morgan Stanley are all reported to be working on their own versions of robo-advisory[3].

Currently, the process generally involves investing client portfolios in low-cost exchange traded funds (ETF's) in order to provide exposure to asset classes while ensuring diversification. Robo-advisory services are therefore engaged at a relatively low level in the value-adding process of asset management.

One example of a robo-advisory service is the one offered by KeyTrade Bank in Belgium and Luxembourg[4]. This robo-advisor first generates a client risk profile using an online questionnaire. Based upon the answers the clients provide, the client is assigned to one of 10 risk profiles, from "very defensive" (low risk) to "very aggressive" (high risk), and assigned to one of 10 corresponding portfolios. Twelve ETF's were selected to represent 12 asset classes: Euro-Zone equities, US Equities, Japanese Equities, Emerging Market Equities, Pacific ex-Japan Equities, Euro-Zone Government Bonds, Bonds issued by companies in the Euro-Zone, High Yield Bonds, Emerging Market Bonds, Index-Linked Bonds, Industrial Metals, Gold. Each of the 10 portfolios can be invested in any or all of these ETF's, as well as Cash. The portfolio weights are determined on a monthly basis, based upon the risk-aversion coefficient associated with the risk profile, in a Black-Litterman framework which takes into consideration the views of the "Investment Committee". This "Investment Committee" oversees the investment process, and reviews the weights assigned to each portfolio to ensure they are reasonable. Trades are then performed automatically, on all client accounts. The process therefore combines minimal human intervention with a basic algorithm to perform the asset management process. The main advantages of this robo-advisor are said to be lower cost, and lower minimum requirement on assets under management, as an account can be opened with assets as low as 15,000 EUR.

[2]https://en.wikipedia.org/wiki/Robo-advisor
[3]July 31, 2017 issue of Barron's magazine available at https://www.barrons.com/magazine?archives=2017
[4]http://www.keytradebank.lu

Chapter 3

Experimental Data and Scientific Platform

This chapter presents the experimental data used in the experiments, including equity market data and macroeconomic data. It also describes the scientific analysis platform developed, and discusses design choices that have been made.

3.1 Introduction

Two types of data are used in this research. Equity market data, including the prices of securities, as well as the trading volume and market capitalization, are the first type. Macroeconomic data, including fundamental indicators of economic activity, inflation, interest rates, and oil prices, are the second. Data for Experiments 1 and 2 covered the period from 31/12/1994 to 30/09/2017. Data for Experiment 3 covered the period from 31/7/2016 to 1/8/2019.

The development of a scientific analysis platform for research on asset management is motivated first by the need to investigate different models at different steps of the asset management process, second by the need to compare the results of the use of these different models, and third by the desire to communicate these results to the investment community. The scientific analysis platform is composed of a framework which breaks down the asset management process into stages each of which can be enhanced by computational methods, and an audit trail at each stage that provides insight into the decision making process at that stage. The platform is described in Section 3.3.

3.2 Experimental Data

3.2.1 Equity Data

Work is based on the NASDAQ-100 (the Index), which represents the 100[1] largest companies listed on the NASDAQ weighted by market capitalization, excluding companies in the

[1]At 31/10/2017, the index is comprised of 107 companies.

financial sector. It is the basis for one of the most actively traded exchange traded funds (the ETF) in the world, the Powershares QQQ Trust, Series 1. This index was formed on January 31, 1985, is reviewed on a quarterly basis, and rebalanced on an annual basis, based on market values at end of October and shares outstanding at the end of November of each year. Rebalancing is published beginning December. The ETF was launched on March 10, 1999. Monthly data starts on March 31, 1999.

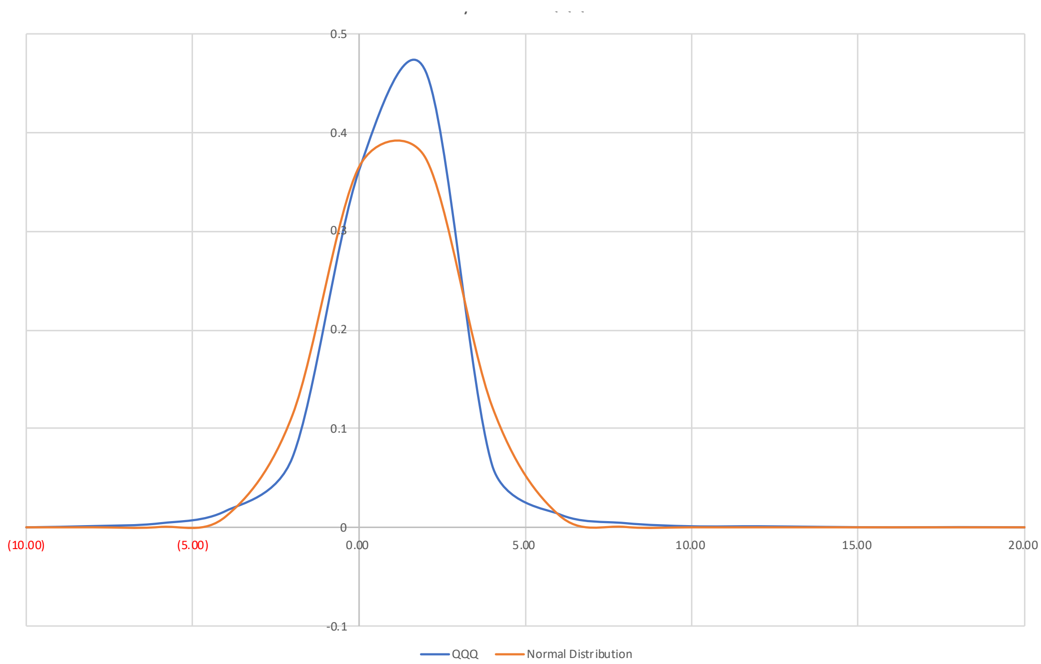
One of the criteria for inclusion in the index is market liquidity: the average daily trading volume of equities considered for inclusion must be at least 200,000 shares[2]. This provides some assurance that the closing prices quoted by market data providers are based on recent trades. Using only securities which are included in the Index therefore helps avoid erroneous results based on securities prices which cannot, in practice, be realized.

Data regarding the composition of the index at each quarter from 31/03/1995 to 30/09/2017 as well as shares outstanding over the same period were obtained from Siblis Research[3]. A total of 350 companies were part of the index at some point during this period. Of these, daily price data is available for 211 companies. For the remaining 139, only quarterly price data is available for the period from 31/03/1995 to 31/12/2004. Daily price data is available thereafter. Details are provided in Appendix C. This research focuses on the 211 companies for which daily price data is available for the entire period under investigation.

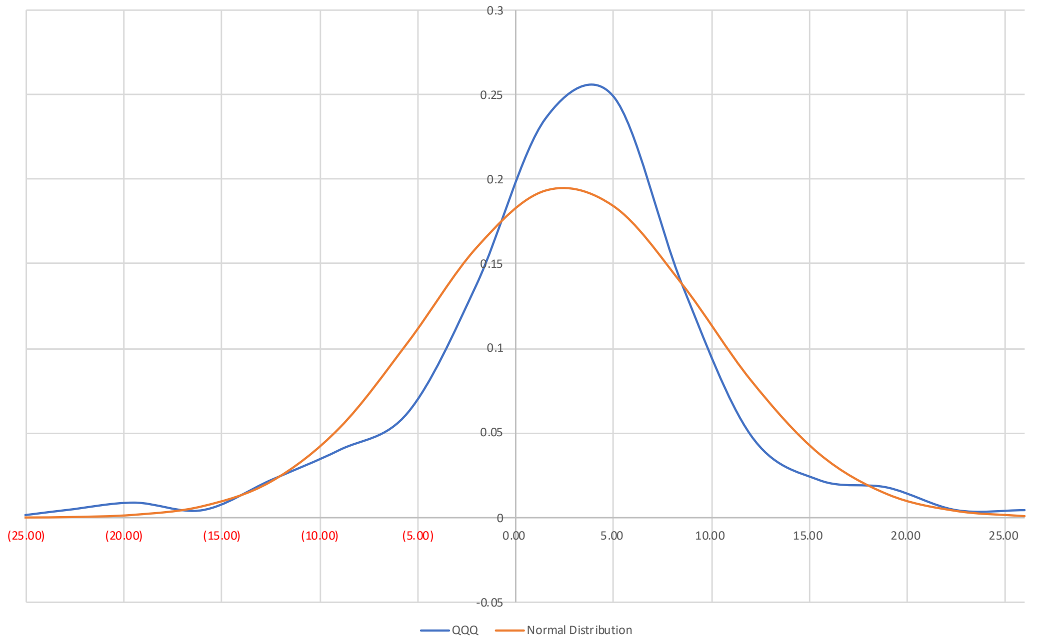
In the interest of understanding the data, Figure 3.1 presents the distribution of daily and monthly historical returns of the QQQ ETF compared to a normal distribution with the same mean and standard deviation as the daily or monthly series. Table 3.1 presents the mean, standard deviation, excess kurtosis, and skew of the two series. The Jarque-Bera test statistic is used to determine whether sample data come from a normal distribution. Comparing this statistic to the χ^2 distribution with two degrees of freedom, the assumption that either the daily or monthly series of returns is normally distributed can be rejected. In particular the positive kurtosis in both cases indicates the distribution is leptokurtic: the probability of extreme gains and losses are higher than would be expected if returns were normally distributed. This corresponds to observed “fat tails” in Figure 3.1. The positive skewness of daily returns indicates the likelihood of extreme losses is correspondingly higher than the likelihood of extreme gains. This agrees with market wisdom that gains are slow and losses sudden. On the other hand, the negative skewness of monthly returns would indicate the opposite, which is perhaps unexpected. This is not investigated further. The conclusion of this analysis is that methods which assume the returns series is normally distributed should be used with caution.

Jensen (2000) presents reasons historical data must be used with care. Of most interest to

[2]https://www.nasdaq.com/markets/indices/nasdaq-100.aspx retrieved 10/05/2018.
[3]www.siblisresearch.com



(a) The distribution of daily returns compared to a normal distribution.



(b) Distribution of monthly returns compared to a normal distribution.

Figure 3.1: The distribution of daily and monthly returns of the QQQ ETF compared to normal distributions. Daily returns exhibit positive skew and significant excess kurtosis while monthly returns actually exhibit slight negative skew over the period, and a relatively lower positive kurtosis. Neither daily nor monthly returns appear normally distributed.

this research is survivorship bias (Garcia and Gould, 1993; Malkiel, 1995). In this case, survivorship bias refers to the fact that the historical data used may exclude companies

Data	Daily	Std Error	Monthly	Std Error
Mean	0.04%	0.025%	0.8%	0.47%
Standard Deviation	1.8%		7.1%	
Skew	0.34	0.035	-0.38	0.16
Excess Kurtosis	7.01	0.071	1.89	0.32
Jarque-Bera Test	9,793		38	
Sample Size	4,734		225	
$\chi^2(2)$	~0		~10^{-9}	

Table 3.1: Statistics for Daily and Monthly Returns QQQ ETF. Neither daily nor monthly returns for this ETF are normally distributed.

which are no longer quoted on the NASDAQ today. One of reasons a company may no longer be quoted is bankruptcy. Simulations using a dataset which is biased in this way may be expected to produce higher returns than one which includes companies which are now bankrupt. The QQQ ETF may have included these companies in the past, and therefore the performance of portfolios constructed in this research cannot be directly compared to the performance of this ETF. Since the benchmarks developed in the first experiment use the same (potentially biased) dataset as the later experiments, the impact of this bias is partly eliminated. There remains the risk that models which are developed work well with healthy companies but would fail with the type of companies which go bankrupt. This risk is addressed in the third experiment, working on live data.

The initial data included some errors. In some cases, the prices were provided as strings rather than floating point numbers. All price and market capitalization data were reviewed manually to remove these spurious errors.

All securities used in this research are quoted in US dollars (USD). This is therefore the reference currency used.

The third experiment, working on daily data in actual trading conditions, will obtain prices from InteractiveBrokers. This data was of a higher quality than the data from Syblis research. Although problems linked to missing data due to market closures existed, all data was correctly represented and required no manual processing.

All market data being publicly available, no data privacy issues exist.

3.2.2 Macroeconomic Data

Macroeconomic time series data was obtained from the FRED database of the Federal Reserve Bank of St-Louis [4]. This data is also publicly available so that no data privacy issues exist.

[4]https://fred.stlouisfed.org

Data	Daily	Weekly	Monthly
Industrial Production			X
Consumer Price Index			X
Premium	X		
Term	X		
Oil	X		
Stress		X	

Table 3.2: This table presents the frequency at which factor data was available. Although Consumer Price Index and Industrial Production data are available on a monthly basis, there is a delay of several weeks after the end of the month to which they relate, before they become available. All other data is either available on the FRED database without delay, or represents data which is widely available in real time.

Other data sources were considered, including the data set used in (Beach and Orlov, 2007), the IFC Database which is now the Global Financial Development Database of the World Bank. However, as at 10/10/2018, this database was last been updated in July 2018. Information is therefore clearly not updated sufficiently frequently to be used for trading. In addition, back testing on the assumption this information was available at the time of trading will be unreliable at best as it assumes access to information which was not available to other market participants.

The following data series were obtained from FRED:

- Industrial Production: Industrial Production Index (INDPRO[5])
- CPI: Consumer Price Index for All Urban Consumers: All Items Less Food and Energy (CPILFESL)
- Premium: Difference between
  Moody's Seasoned BAA Corporate Bond Yield (BAA, Daily: DBAA)
  Moody's Seasoned AAA Corporate Bond Yield (AAA, Daily: DAAA)
- Term: 10-Year Treasury Constant Maturity Minus 2-Year Treasury Constant Maturity (T10Y2Y)
- Oil: Crude Oil Prices: West Texas Intermediate (WTI) - Cushing, Oklahoma (DCOILWTICO)
- Financial Stress: St. Louis Fed Financial Stress Index (STLFSI)

These data series were selected based on Beach and Orlov (2007).

The frequency of the data is presented in Table 3.2.

It should be emphasized that Industrial Production and Consumer Price Index data is available with considerable delay. CPI data for July 2018, for example, was available on Septem-

[5]The code in parenthesis is the FRED identifier for the series.

Description	Production	Inflation	Premium	Term	Oil
Mean	0.13%	0.18%	0.977	1.1909	0.81%
Maximum	2.05%	1.38%	3.430	2.8400	35.34%
Minimum	-4.31%	-1.77%	0.530	-0.4700	-32.37%
SD	0.65%	0.28%	0.435	0.9113	9.22%

Table 3.3: Macroeconomic factors, summary statistics.

ber 13 2018. This data is subject to embargo before that date, in order not to give unfair advantage to any market participants. Use of this data in back testing must be done with care, so that data is not used before it would have become available.

Other data such as corporate bond yields, treasury rates, and oil prices are included in the Fred database with some delay, however, as these prices are widely available on a near real time basis, there is less of an issue with their use.

In some cases, manual preprocessing of this data was required. For example, data for T10Y2Y on the 30/03/2018, the last business day in March of that year, is blank. There is, however, equity data on that day. In similar cases, data was replaced by data from the previous day.

Monthly changes in inflation were calculated based on the CPI. Monthly changes in Production were calculated based on the Industrial Production Index. The price of oil was not adjusted for inflation.

3.2.3 ETF's as Predictors of the Performance of Economic Factors

It would be useful to use equity prices to predict the performance of economic factors since, contrary to economic data, equity prices are easily available on a daily or real time basis. Breeden (1979) showed how portfolios can be used to replace state variables in an asset pricing model. Huberman, Kandel, and Stambaugh (1987) investigates the characteristics of portfolios with returns which mimick the performance of factors. Andersson et al. (2011) and Lamont (2001) report there is some evidence that equity portfolios can be predictors of economic factors such as consumption, inflation, and GDP. Melas, Suryanarayanan, and Cavaglia (2010) discusses practical issues such as trading volume.

Using ETF's which focus on companies in defined economic sectors may be one way of summarizing the market's view of the future performance of economic factors relevant to that sector. The future performance of key economic factors relevant to sectors such as raw materials may be relevant to the future returns of companies which depend indirectly on these sectors - for example because they are suppliers to companies in that sector.

3.2.4 Risk-Free Rate

The “risk-free” interest rate is sometimes required in certain algorithms. This rate is return available on a theoretical security which is free of default risk. A proxy for the risk-free rate for the US Dollar is normally the return on a security issued by the US Government, of an appropriate maturity (Grabowski, Nunes, and Harrington, 2014). As much of the focus of this research is on a one-month period between rebalancing, a logical risk-free rate would be the annualized 1-month Treasury constant maturity rate[6], however data on this series is only available from the FRED database as of 31.07.2001.

There are other interest rates which could be used to replace the 1-month treasury rate, including the 3-Month Treasury Constant Maturity Rate[7], for which data is available from the FRED database as of 04.01.1982, and the effective federal funds rate, which is a weighted average of overnight interest rates negotiated between financial institutions for lending balances held at a Federal Reserve Bank[8]. Data for this rate is readily available as of at least 1955.

Figure 3.2 compares the federal funds rate with the two treasury rates for the period from 31.07.2001. Over the period for which data is available for all series, both the federal funds and the 3-month treasury rate are close to the 1-month rate, although the federal funds rate exhibits more volatility than either the 1-month or 3-month rates. The choice of reference risk-free rate could be expected to impact results. Visually, the 1-month and 3-month rates may appear to behave most similarly. In fact, the correlation between the federal funds rate and the 1-month treasury rate is .98 over this period, while the correlation between the 3-month and the 1-month treasury rates is .99. However, correlations between changes in the federal funds rate and the 1-month treasury rate is much lower than between changes in the 3-month and 1-month treasury rates: .08 versus .67. As the change in interest rates could be expected to be relevant to changes in equity prices, for the purposes of Experiment 1 and 2, the risk-free rate is therefore taken to be the annualized 3-Month Treasury Constant Maturity Rate from the FRED database. A graph of the entire series is presented in Figure 3.3. The federal funds rate will be used for Experiment 3, as this experiment focuses on daily returns.

3.3 Scientific Analysis Platform

This research explores the use of Bayesian networks to develop views for a Black-Litterman asset allocation model, and determine whether they can help in the creation of better investment portfolios. It involves the development and comparison of a number of asset management models. In time, the number of models may be increased and may include

[6]https://fred.stlouisfed.org/series/DGS1MO
[7]https://fred.stlouisfed.org/series/DGS3MO
[8]https://fred.stlouisfed.org/series/FEDFUNDS

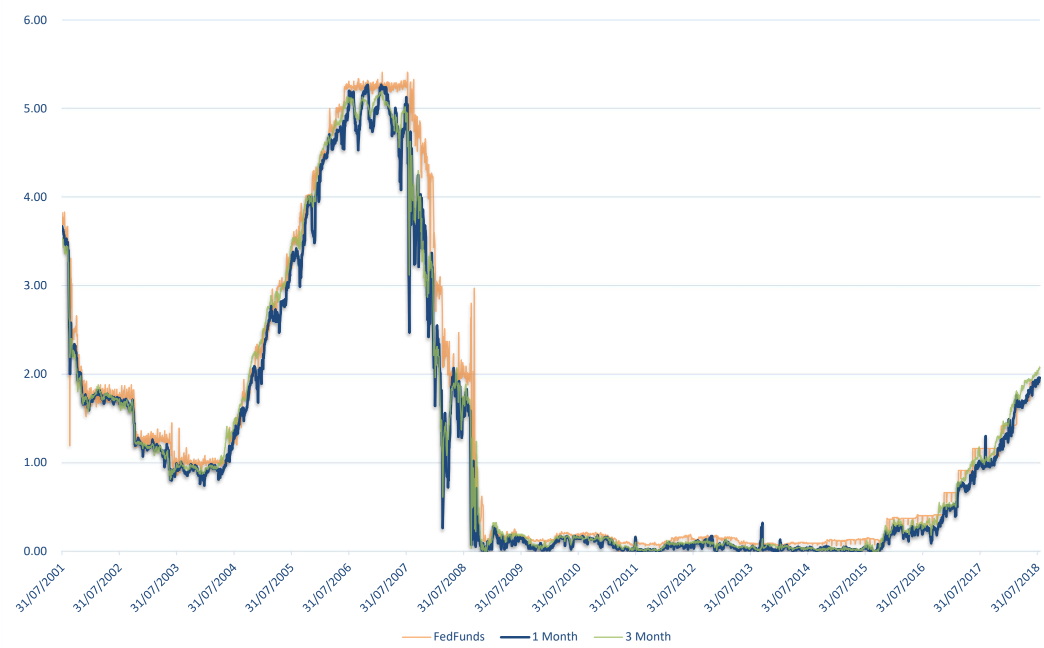


Figure 3.2: Federal funds, 1-month, and 3-month Treasury rates are compared. The federal funds rate exhibits more volatility than either 1-month or 3-month Treasury rates. The choice of reference risk-free rate could be expected to impact results. The 3-month Treasury rate is the reference rate for Experiments 1 and 2. The federal funds rate is the reference rate for Experiment 3.

some not specifically designed with this research in mind. It is therefore desirable to create a platform to facilitate the processes of model development, adaptation, and comparison.

Ultimately, this platform may be of interest to final users outside of academia. Where possible, the needs of these users should be anticipated in designing the platform in order to facilitate further enhancements to be brought at a later stage.

It may, at some point, be desirable to include two additional layers in the final platform: one addressing communication with users in the form of an intelligent conversational agent, another providing a live audit trail through a decentralized ledger of all transactions proposed by the various models over time.

3.3.1 Motivation

The development of the scientific analysis platform for research on asset management is motivated by the need to first investigate the use of different models at each stage of the asset management process, second compare the results of these models, and third communicate these results to third parties.

3.3.2 Design Objectives

One objective of the platform is to help determine whether asset management models based on Bayesian networks and a Black-Litterman asset allocation model are better, in some

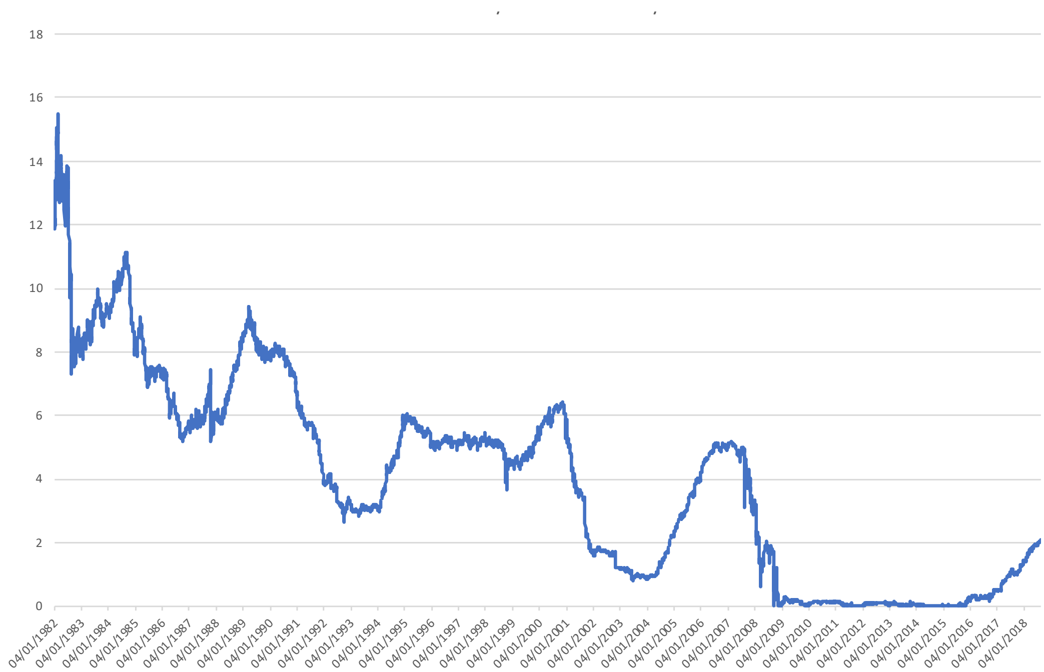


Figure 3.3: Risk-Free Rate: 3-month Treasury constant maturity rate. This is the reference rate for Experiments 1 and 2. Although the 3-month rate has been relatively low and stable in the past few years, this has not always been the case.

sense, than a set of benchmark portfolios.

In order to accomplish this, the platform should be designed to facilitate asset management model creation as well as model comparison. For example, it should be possible to use different asset models or comparison engines with minimal impact on the other parts of the system.

The platform should be designed to further the objectives of both this research and future final users if their needs can be anticipated. In particular, as one objective of this research is the comparison of model performance, it should help ensure that differences in model performance are due to the model decisions rather than external factors.

An overview of the proposed scientific platform is presented in Figure 3.4. It should be mentioned that other approaches to asset management exist. This approach was selected and this platform was designed in order to permit the use of different machine learning models at critical points in an asset management process.

From the point of view of data flows, the platform begins with the universe of equities and factors. Some means of selecting a subset of all available equities and all possible factors must be defined. This is termed “dimensionality reduction”. The data is fed to a model to predict the risk and return of individual investments. The predictions are fed into a portfolio construction model. The performance of the portfolios are analysed and reports are prepared.

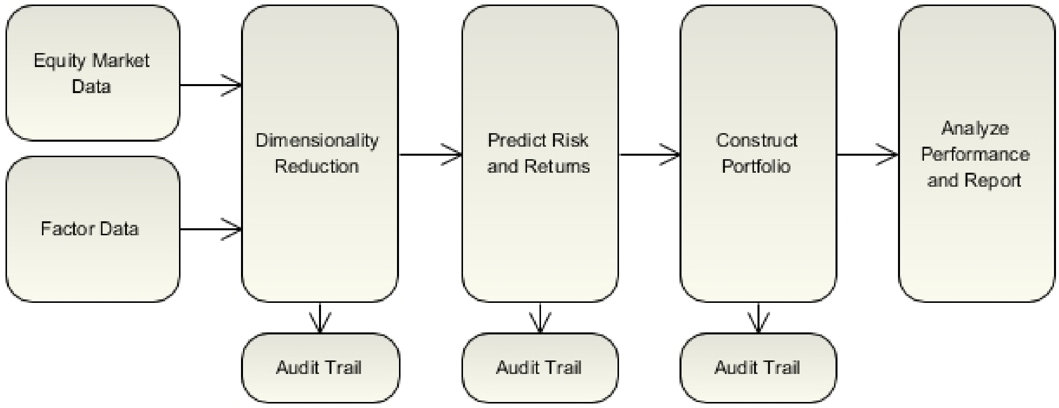


Figure 3.4: This is an overview of the scientific platform. A more detailed schema of one possible implementation is given in Figure 6.1.

At each step, an appropriate audit trail is automatically created. The audit trail ensures that the decisions taken by the model in that step can be understood and explained (Samek et al., 2019).

Each step of the process can be replaced by another model. For example, while this research uses Bayesian networks for return prediction, other models can be used for this step without modifying the rest of the platform.

At this stage, the scientific analysis platform is general. One possible implementation is presented in Section 6.5.

3.3.2.1 Data acquisition

Allowing models the freedom to select any existing security, in any market, may have certain advantages. Some models may be very well adapted to building portfolios of emerging market debt securities. Other models may be better at working with developed markets. If the first portfolio outperforms the second, one would need to know if the first model is better than the second, or whether emerging market debt, in general, outperforms developed markets. One would need to compare both models and markets.

On the other hand, it may be desirable to constrain the universe of securities that models may include in their portfolio. If all asset management models build their portfolio from the same list of securities, one source of uncertainty is removed which helps ensure that differences in performance are not due to external factors. In addition, as this research focuses on securities which were part of the NASDAQ 100, it is necessary that models be able to work with these securities. Finally, in practice, the final user may wish to select the securities or asset classes from which models build their portfolios. Users may, for example, wish to invest only in socially responsible companies. Models must be able to accept this constraint.

For similar reasons, models must be constructed to accept securities market data. The final user may have a preferred data source including for example Bloomberg, Refinitiv (formerly Thomson Reuters), Interactive Brokers, Telekurs, FactSet, and Quandl. Model performance may not be "improved" simply by using more favourable market prices. However, some models may use additional data. For example, some may use news or macro-economic indicators. Models must be permitted to obtain data - other than market data - on their own, as otherwise crucial flexibility may be lost.

Designing the market data acquisition stage to be separate from the asset management models, while allowing models to obtain any additional data they require, helps meet these objectives.

In practice, market data validation would need to be performed. First, market data should be valid in the obvious sense of being positive floating point numbers, perhaps in an acceptable range. In addition, all securities used must have sufficient trading volume to ensure that trades could actually be performed at the given market prices. For the purpose of this research, the data used have been verified manually (Chapter 3).

3.3.2.2 Model creation

In a mean-variance framework as used throughout this research, portfolio construction has two parts: estimating future risk and return, and creating an optimized portfolio using these estimated parameters. In this research, the portfolio is created using a Black-Litterman algorithm, as the focus is on comparing different models for parameter estimation. At a later stage, other approaches to portfolio construction could be considered, using a broader array of parameters, and portfolio construction could ultimately be separate from parameter estimation.

Some models will be developed specifically for this platform. These models must have a common interface. Other models may be provided, for example by the final user. It must be simple to include (or exclude) a new model from the comparison.

3.3.2.3 Model comparison

The comparison of highly dynamic models is a complex field of research. Industry standard mean-variance metrics are implemented in this research, however many others may be envisaged. For example, other metrics may be meaningful if it is desired that risk should be measured taking skew or kurtosis into consideration. It would be very interesting to be able, at a later stage, to implement more advanced comparison engines. The comparison stage must therefore be separate from the other stages in order to facilitate its replacement.

3.3.2.4 User interface

At a later stage, the results of this research would be provided in the form of software as a service (SAAS). In preparation for this, all results and interactions with users are through a web interface.

3.3.3 Design Overview

The platform will be organized around a mediator design pattern (Gamma et al., 1995). In this design pattern, only one class has direct knowledge of the interface of other classes. These other classes may therefore easily be modified and work independently each other. This approach helps ensure loose coupling and tight cohesion.

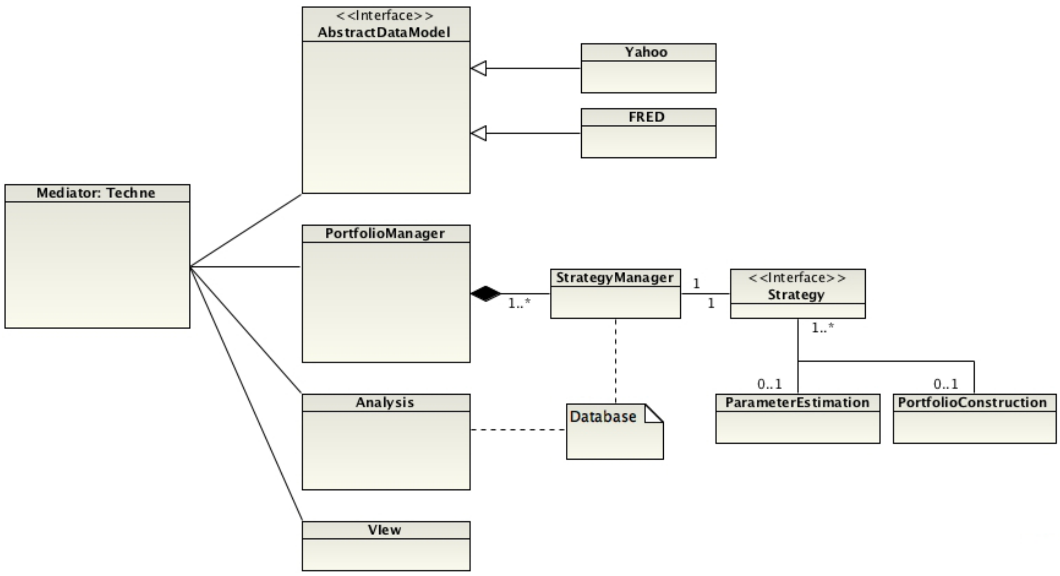


Figure 3.5: The scientific platform was developed in such a way as to allow the independent development of the component parts. This approach helps ensure loose coupling and tight cohesion.

The main classes presented in Figure 3.5 are:

- Techne: Techne is the mediator class. All interactions with the web interface, data, and model updates are through this class.
- AbstractDataModel: Defines the interface to download data.
- Strategy: Defines the specifics of the portfolio construction process, including the asset management model:

  ParameterEstimation: e.g. a Bayesian network for risk and return estimation

  PortfolioConstruction: a Black-Litterman algorithm, where relevant
- StrategyManager: Manages one Strategy object. Defines the structure of backtesting. Stores the result in a database.
- PortfolioManager: Manages a set of Strategy objects.
- Analysis: Defines metrics to compare asset management models. Accesses a database

for the results of backtesting previously stored by the StrategyManager.

- View: Django web interface.

When developing parameter estimation models, they may be accessed directly by the mediator for testing purposes. This is done without the creation of a Strategy object which uses the model. This practical dependency of the mediator class is not discussed further and is not included in Figure 3.5.

3.3.4 Tools

Practical decisions regarding tools to use when developing a SAAS solution include choice of the programming language, web framework, and database.

- Programming language
  Python 3[9] was selected as the programming language as many high quality, optimized python packages are available to handle large data sets, including pandas[10].
- Web framework
  A web framework is useful as it automatically handles many aspects of web development, such as web page templates, database access, forms, email, and user authentication. It also helps avoid security issues such as SQL injection, cross-site scripting, forgery and clickjacking. From the number of web frameworks which can be used with python, Django[11] was selected based on experience with this package. One useful aspect of Django is that it naturally implements a type of MVA by separating web page templates from the core business logic.
- Database
  A database management system may be used for data storage to deal with concurrent data access, eliminate redundancy, and help ensure data security. Possible choices include SQLite, MySQL, and PostgreSQL. Django recommends either MySQL or PostgreSQL[12].
  For practical reasons, the website is hosted by DreamHost. The simplest database solution on DreamHost is MySQL.
  In many cases, the advantages of a database were not required. Rather than converting database objects to pandas DataFrames (and vice versa), a simpler solution was often saving the pandas DataFrame as a pickle[13] object file.

3.3.5 Testing

Django supports creation of an automated test suite using an integrated unittest module. Tests may be written for all stages of the web-based application, including the HTTP-level

[9] https://www.python.org, accessed 10/05/2018
[10] https://pandas.pydata.org, accessed 10/05/2018
[11] https://www.djangoproject.com, accessed 10/05/2018
[12] https://docs.djangoproject.com/en/2.0/topics/migrations/
[13] https://docs.python.org/3/library/pickle.html, accessed 10/05/2018

requests.

3.3.6 Data Visualization

This research makes use of large data sets, elaborate data flows, and complex algorithms. These may contain errors which are susceptible to being caught by data validation and automated testing procedures, if these are designed to catch them. However, some of these errors may not be easily identifiable a priori. For example, the algorithm may call for trading a security which is not liquid, or holding a position in a security which exceeds a reasonable percentage of the outstanding equity of a company. Visualization of the data sets and independent reproduction of key metrics would be useful in identifying new or unexpected types of errors. It could also be useful in prototyping new metrics.

This additional layer of testing can be performed by using pandas and xlsxwriter[14] to output information to Microsoft Excel files. This technique can be used to create a full audit trail of program logic and relevant data.

Many of the charts and tables presented in this research have been produced using this technique. For example, a chart or table was defined once, for one Strategy, and was prepared automatically for all other Strategies.

Ultimately, this process could be adapted to provide final users with Excel files containing a level of detail appropriate to their needs.

3.3.7 Summary

This section has presented the scientific analysis platform developed as part of this research. It has considered the motivation for the development of the platform and the objectives which drive its design. The stages of data acquisition, model creation, and model comparison have been considered, as well as the requirement that each stage of the framework is loosely coupled in order to facilitate the use of different computational models.

[14]http://xlsxwriter.readthedocs.io, accessed 10/05/2018

Chapter 4

A Benchmark Portfolio Using Traditional Algorithms

This experiment establishes benchmarks against which the performance of other models are compared. The benchmarks are formed from a subset of the shares underlying the NASDAQ-100 equity index. The selected portfolios will be formed using two strategies: one which allocates equal weights to each security in the portfolio, and a canonical Black-Litterman model without additional information in the form of views on the performance of individual assets. The performance of these portfolios will be analyzed and compared to the performance of the QQQ Exchange Traded Fund which replicates the performance of the NASDAQ-100 equity index.

4.1 Introduction

This research explores the use of Bayesian networks to develop views for a Black-Litterman asset allocation model, to determine whether they can help in the creation of better investment portfolios. The objectives of this experiment are to develop benchmarks against which models implementing Bayesian networks can be compared, and to begin defining what is meant by “better” portfolios. These objectives are met in three steps. The first involves the creation of a simple portfolio based on buying and holding the QQQ ETF. The second involves the creation of a simple model which allocates an equal percentage of the total portfolio value to each security in the portfolio. The third step involves the implementation of a canonical Black-Litterman model. Each step is useful in building and validating the securities data set which will be used throughout this research, and implements a first version of the scientific platform, including backtesting and comparison metrics, which will be enhanced in later experiments.

4.2 Aims and Objectives

This experiment is divided into three parts. The first uses the QQQ ETF. The other two create portfolios which include the 211 companies for which daily data is available for the entire period under investigation (see Section 3.2.1).

First, a simple portfolio based on buying and holding the QQQ ETF is developed and tested. This ETF is comprised of the securities of interest to this research, it is low cost, publicly traded, and easily accessible. It therefore sets an objective benchmark of the risk and return profile for any portfolio invested in this asset class. However, it is not sufficient in this case as the data set used in this research may be biased (see Section 3.2.1).

Second, a portfolio based on allocating an equal weight to each security in the portfolio is developed and tested. In an equal-weighted portfolio, the same USD amount is invested in each security in the portfolio, at the beginning of each month. DeMiguel, Garlappi, and Uppal (2009) found that equal-weighted portfolios outperformed value-weighted portfolios in which the portfolio weight of a security is calculated based on it's market capitalization. This portfolio may therefore be a challenging benchmark. The backtesting procedure is developed and tested. The returns of this portfolio are compared to the first benchmark.

Third, a portfolio based on a canonical Black-Litterman algorithm, in the absence of views is developed and tested. If the implementation functions correctly, it should, in the absence of views, return portfolio weights which are similar to *market weights*: weights based on market capitalization. Due to errors introduced in particular by the matrix inversion process, the two sets of weights are not expected to be exactly the same, however they should not be systematically very different. This portfolio would then behave as a value-weighted portfolio.

4.3 Implementation

4.3.1 General

In this experiment, portfolios are rebalanced on a monthly basis, at the end of each month, starting from 1995. Except for the first and final year, 12 portfolios are constructed, per year. Metrics are based on the value of these monthly portfolios.

In practice, to ensure efficient execution, equity positions normally should and sometimes must be multiples of 100 shares. This limitation is ignored.

Transaction fees are charged on both purchases and sales of securities. Transaction fees are ignored for all portfolios, including the benchmark.

4.3.2 QQQ ETF

Developing and testing the QQQ ETF benchmark involves implementing part of the scientific platform including download of data, data validation, testing, export to Microsoft Excel, and presentation via a web-based interface. It also involves developing the backtesting procedure and basic return comparison metrics.

Data for the QQQ ETF are used in two ways. First, comparison metrics are run on the simple historical price series. Second, a Strategy (see Figure 3.5) is implemented to create a portfolio containing only units of the QQQ ETF. Metrics run on this Strategy as part of the backtesting procedure would be expected to equal those calculated based on the simple price series. This provides some assurance that the backtesting procedure functions as expected.

4.3.3 Equal Weights

Developing and testing the equal weights benchmark involves identifying all securities which were part of the NASDAQ-100, defining an AbstractDataModel interface and implementing it to download adjusted price data, manually validating the data, and defining and implementing a Strategy interface which uses an EqualWeights implementation of the PortfolioConstruction interface (See Figure 3.5).

The details of the calculation of the equal weights portfolio are illustrated in Figure 4.1 which presents the positions invested in American Airlines shares for the period from 27/02/2015 to 30/06/2017. The opening market value of the position is equal to the opening portfolio market value, divided by the number of securities in the portfolio. As at 31/10/2017, the Index was comprised of 107 companies. The number of securities included in the portfolios created in this research may vary depending upon the availability of daily market data, as discussed in Section 3.2.1. The quantity invested in the security is modified based on the portfolio's closing market value, so as to maintain an equal weight for all securities in the portfolio.

4.3.4 Canonical Black-Litterman

Developing and testing the Black-Litterman algorithm, in the absence of views involves downloading and validating market capitalization data, and creating an implementation of the Strategy interface which uses a Black-Litterman implementation of the PortfolioConstruction interface (See Figure 3.5).

One contribution of the Black-Litterman algorithm is the fact that it begins with expected market returns, calculated assuming the market is in equilibrium. It then combines these market returns with user provided views to generate the portfolio. One might expect that supplying the Black-Litterman algorithm with "neutral" views, which equal what the Black-Litterman algorithm calculates are market risk and returns, would yield the same portfolio as if no views had been supplied. If this is the case, the posterior returns calculated by the algorithm would be close to the priors calculated as equilibrium market weights, and the portfolio weights returned by a Black-Litterman algorithm with neutral views would also be very close to weights based on market capitalization. Implementing the third benchmark in this way has the advantage of providing some assurance that the implementation

value_date	Adj Close	Weight	Quantity	Opening Market Value	Closing Market Value	Opening Portfolio Market Value	Number of Securities in Portfolio
2015-02-27	46.75	1.02%	3,091	147,754	144,498	14,479,935	98
2015-03-31	51.52	1.02%	3,397	158,814	174,994	15,563,809	98
2015-04-30	47.23	1.02%	3,018	155,484	142,536	15,237,401	98
2015-05-29	41.44	1.02%	3,325	157,032	137,795	15,389,161	98
2015-06-30	39.06	1.02%	3,859	159,920	150,729	15,672,156	98
2015-07-31	39.22	1.02%	3,992	155,924	156,568	15,280,571	98
2015-08-31	38.22	1.02%	3,983	156,230	152,227	15,310,574	98
2015-09-30	38.07	1.01%	3,780	144,443	143,887	14,299,897	99
2015-10-30	45.32	1.01%	3,661	139,384	165,911	13,798,989	99
2015-11-30	40.54	1.01%	3,357	152,106	136,075	15,058,535	99
2015-12-31	41.61	1.01%	3,780	153,228	157,275	15,169,527	99
2016-01-29	38.31	1.00%	3,624	150,797	138,833	15,079,726	100
2016-02-29	40.40	1.00%	3,620	138,673	146,232	13,867,343	100
2016-03-31	40.41	1.00%	3,440	138,985	139,019	13,898,474	100
2016-04-29	34.18	1.00%	3,645	147,263	124,569	14,726,349	100
2016-05-31	31.53	1.00%	4,197	143,446	132,334	14,344,600	100
2016-06-30	27.97	1.00%	4,735	149,318	132,472	14,931,804	100
2016-07-29	35.08	1.00%	5,239	146,551	183,771	14,655,089	100
2016-08-31	35.98	1.00%	4,457	156,355	160,356	15,635,464	100
2016-09-30	36.28	1.00%	4,379	157,552	158,897	15,755,154	100
2016-10-31	40.24	1.00%	4,389	159,260	176,617	15,926,001	100
2016-11-30	46.14	1.00%	3,895	156,728	179,728	15,672,802	100
2016-12-30	46.39	1.00%	3,500	161,520	162,390	16,152,005	100
2017-01-31	43.97	1.00%	3,483	161,566	153,123	16,156,647	100
2017-02-28	46.16	1.00%	3,892	171,107	179,654	17,110,734	100
2017-03-31	42.12	1.00%	3,839	177,211	161,692	17,721,116	100
2017-04-28	42.44	1.01%	4,278	180,192	181,556	17,839,050	99
2017-05-31	48.31	1.01%	4,325	183,539	208,928	18,170,357	99
2017-06-30	50.22	1.01%	3,908	188,778	196,226	18,689,014	99

Figure 4.1: Equal weights strategy sample: AAL (American Airlines).

correctly takes views into consideration. This avoids having to modify the Black-Litterman implementation in the future.

The implementation first calculates the prior expected returns, $\boldsymbol{\Pi}$, using Eq. (2.21). This requires the calculation of the risk aversion parameter δ, the covariance matrix $\boldsymbol{\Sigma}$, and the market weights $\boldsymbol{w}$.

The value of the risk aversion parameter of the market, δ can be calculated using Eq. (2.23). Walters (2008) states that the value of δ can be assumed if one does not know the expected return and standard deviation of the market portfolio. In this research, the value of δ was assumed to be constant and equal to 2.5, following the value used by He and Litterman (1999).

Care must be taken when calculating the value of the prior covariance matrix $\boldsymbol{\Sigma}$ to ensure that the number of data points used is higher than the number of equities in the portfolio, otherwise the resulting matrix will be singular and it cannot be inverted. In this research, the covariance matrix $\boldsymbol{\Sigma}$ is calculated using the covariance of the last 180 monthly returns $\boldsymbol{r}$ (Satchell and Scowcroft, 2000)

$$\boldsymbol{\Sigma} = 12 \times \operatorname{cov}(r, r). \tag{4.1}$$

Market weights $\boldsymbol{w}$ were determined from the number of outstanding shares multiplied by the share price at the end of the month.

The Black-Litterman model determines a combined return distribution of returns from a prior equilibrium distribution and a view distribution. These are distributed as follows (Idzorek, 2007)

$$\boldsymbol{\pi} \sim N(\Pi, \tau\boldsymbol{\Sigma}) \tag{4.2}$$

$$\boldsymbol{\omega} \sim N(Q, \boldsymbol{\Omega}) \tag{4.3}$$

$$\boldsymbol{\rho} \sim N(E[\boldsymbol{R}], [\tau\boldsymbol{\Sigma}^{-1} + (\boldsymbol{P}^{\intercal}\boldsymbol{\Omega}P)]^{-1}), \tag{4.4}$$

where $\boldsymbol{\pi}$ is the prior, $\boldsymbol{\omega}$ the view, and $\boldsymbol{\rho}$ the combined returns. The scalar τ is a measure of the uncertainty of the covariance matrix. Idzorek (2007) discusses the role of this parameter and its value in literature. It is set to 0.05 in this research.

The posterior expected return can be calculated using Eq. (2.28).

In future experiments, the only elements which will change between the third portfolio and future implementations is the estimation of the parameters which comprise views and the frequency of rebalancing: the same market data is used, and the same fundamental Black-Litterman algorithm is used. Although backtesting procedures and comparison metrics evolve throughout this research, they are applied consistently to all portfolios, including the benchmarks.

4.4 Results

4.4.1 Overview

An overview of the initial web page presenting the returns based on backtesting is provided in Figure 4.2. The figure presents the returns an investor would have had if invested in each of the strategies - with monthly rebalancing - starting one year, three years, five years, and ten years ago. The figure also presents the annual returns of each strategy. Returns for 2017 are until 30.09.2017.

4.4.2 QQQ ETF

Data for the QQQ ETF were used in two ways in order to test the Strategy interface. A first portfolio was created manually using the price of one unit of the ETF. A second portfolio was created using the Strategy interface, with an initial investment of $1M representing 22,040.30 shares of the ETF. If the Strategy and Strategy Manager classes are correctly implemented, the monthly and daily returns of portfolios produced using these two methods should be identical.

Model Analysis	Black Litterman	Equal Weights	QQQ ETF	QQQ_Strategy
One year	0.218176	0.174943	0.237387	0.237387
Three years	0.491623	0.351510	0.519971	0.519971
Five years	1.061317	1.101411	1.253564	1.253564
Ten years	1.705271	1.496258	2.114606	2.114606

Model	Black Litterman	Equal Weights	QQQ ETF	QQQ_Strategy
2017	0.218512	0.158170	0.236238	0.236238
2016	0.071755	0.071415	0.070979	0.070979
2015	0.089421	0.008323	0.094359	0.094359
2014	0.109386	0.202594	0.191814	0.191814
2013	0.374169	0.381451	0.366342	0.366342
2012	0.200370	0.150572	0.181125	0.181125

Figure 4.2: Web page: backtesting of returns.

The yearly returns of these portfolios appear in the two columns with labels "QQQ" in Figure 4.2. Daily returns are presented in Figure 4.3, in which "Price Return" represents the daily return of the ETF and "Return Strategy" represents the daily return of the portfolio produced using the implementation of the Strategy interface. As expected, the two "QQQ" portfolios have identical returns over all periods, which provides evidence that the Strategy and Strategy Manager classes were correctly implemented.

value_date	Adj Close	Weight	Quantity	Market Value	Price Return	Return Strategy
1999-03-11	45.59	1.00	22,040.30	1,004,896.32		
1999-03-12	44.48	1.00	22,040.30	980,416.36	(2.44%)	(2.44%)
1999-03-15	45.76	1.00	22,040.30	1,008,568.21	2.87%	2.87%
1999-03-16	46.15	1.00	22,040.30	1,017,136.00	0.85%	0.85%
1999-03-17	45.82	1.00	22,040.30	1,009,792.37	(0.72%)	(0.72%)
1999-03-18	46.70	1.00	22,040.30	1,029,375.69	1.94%	1.94%
1999-03-19	45.51	1.00	22,040.30	1,003,060.16	(2.56%)	(2.56%)
1999-03-22	44.95	1.00	22,040.30	990,820.39	(1.22%)	(1.22%)
1999-03-23	43.37	1.00	22,040.30	955,936.64	(3.52%)	(3.52%)

Figure 4.3: QQQ ETF and strategy performance calculation.

4.4.3 Equal Weights

Yearly returns of a portfolio constructed using the equal weights Strategy with monthly rebalancing are presented in Figure 4.2. End of month market values of this portfolio are presented in Figure 4.4. The market value calculated using the model is presented in the "Close" column. The closing market values *prior* to rebalancing are recalculated in an Excel spreadsheet using securities values and quantities which are also recalculated in the spreadsheet using equal weights. The results are presented in the "Close (Recalculated)" column. The "Opening (Recalculated)" column recalculates the previous close *after* rebalancing, which is effectively the portfolio's opening market value. The purpose of Figure 4.4 is twofold. First, it tests that the Closing value of the portfolio calculated by the Strategy Manager are correct. Second, it verifies that the act of rebalancing does not change the total portfolio value V, i.e. $V = Q_t \times P_t = Q_{t+1} \times P_t$, where P_t are the prices of assets at time t and Q_t and Q_{t+1} are the number of shares invested in each asset before and after rebalancing. The monthly performance of the portfolio is presented in the last column.

value_date	Close	Close (Recalculated)	Opening (Recalculated)	Performance
1999-03-31	3,562,391.25	3,562,391.25	3,275,536.00	8.76%
1999-04-30	3,801,494.34	3,801,494.34	3,562,391.25	6.71%
1999-05-28	3,791,225.48	3,791,225.48	3,801,494.34	-0.27%
1999-06-30	4,215,324.46	4,215,324.46	3,791,225.48	11.19%
1999-07-30	4,218,304.70	4,218,304.70	4,215,324.46	0.07%
1999-08-31	4,290,330.31	4,290,330.31	4,218,304.70	1.71%
1999-09-30	4,370,139.70	4,370,139.70	4,290,330.31	1.86%
1999-10-29	4,677,436.82	4,677,436.82	4,370,139.70	7.03%
1999-11-30	5,219,463.42	5,219,463.42	4,677,436.82	11.59%

Figure 4.4: Equal weights strategy performance calculation.

4.4.4 Canonical Black-Litterman

Yearly returns of a portfolio constructed using the Black-Litterman Strategy with monthly rebalancing are presented in Figure 4.2.

End of month market values of an early portfolio calculated using an algorithm which contained an error, are presented in Figure 4.5. In this Figure, the market value of the "Opening" portfolio value was not equal to the previous "Close". This difference was caused by rounding errors in calculation of the model weights. In principle, the total value of all weights should have been equal to 1. This was not the case in this early version of the algorithm. The corrected end of month market values are presented in Figure 4.6.

value_date	Close	Close (Recalculated)	Opening (Recalculated)	Performance
1999-03-31	5,030,033.57	5,030,033.57	4,569,750.01	9.74%
1999-04-30	4,932,282.93	4,932,282.93	4,948,806.57	-1.94%
1999-05-28	4,553,962.65	4,553,962.65	4,816,079.80	-7.67%
1999-06-30	5,069,286.16	5,069,286.16	4,484,371.79	11.32%
1999-07-30	5,084,272.78	5,084,272.78	5,054,324.68	0.30%
1999-08-31	5,529,209.31	5,529,209.31	5,083,013.40	8.75%
1999-09-30	5,462,539.27	5,462,539.27	5,520,730.61	-1.21%
1999-10-29	5,756,871.34	5,756,871.34	5,440,687.42	5.39%
1999-11-30	6,279,387.74	6,279,387.74	5,763,751.07	9.08%
1999-12-31	7,779,750.25	7,779,750.25	6,260,628.59	23.89%

Figure 4.5: Error in Black-Litterman strategy performance calculation.

As discussed in 4.3.4, the prior and posterior returns calculated by the model are expected to be close. Figure 4.7 presents the prior and posterior returns for a selection of dates and

value_date	Close	Close (Recalculated)	Opening (Recalculated)	Performance
1999-03-31	4,683,278.79	4,683,278.79	4,254,725.73	10.07%
1999-04-30	4,637,360.38	4,637,360.38	4,683,278.79	-0.98%
1999-05-28	4,384,970.11	4,384,970.11	4,637,360.38	-5.44%
1999-06-30	4,956,919.13	4,956,919.13	4,384,970.11	13.04%
1999-07-30	4,971,782.22	4,971,782.22	4,956,919.13	0.30%
1999-08-31	5,408,214.05	5,408,214.05	4,971,782.22	8.78%
1999-09-30	5,351,208.69	5,351,208.69	5,408,214.05	-1.05%
1999-10-29	5,648,287.01	5,648,287.01	5,351,208.69	5.55%
1999-11-30	6,153,594.03	6,153,594.03	5,648,287.01	8.95%
1999-12-31	7,646,744.10	7,646,744.10	6,153,594.03	24.26%

Figure 4.6: Correct Black-Litterman strategy performance calculation.

securities. Prior and posterior returns were exactly the same, to 10 significant figures, for all dates and all securities.

	30/06/1995		31/07/1995		31/08/1995	
Ticker	Prior	Posterior	Prior	Posterior	Prior	Posterior
AAPL	0.21728	0.21728	0.21413	0.21413	0.21293	0.21293
ADBE	0.21841	0.21841	0.21421	0.21421	0.21618	0.21618
ADSK	0.16896	0.16896	0.16632	0.16632	0.16442	0.16442
AMAT	0.28305	0.28305	0.27674	0.27674	0.27705	0.27705
AMGN	0.15681	0.15681	0.15492	0.15492	0.15186	0.15186
ATML	0.04956	0.04956	0.01996	0.01996	0.08692	0.08692
BOBE	(0.01500)	(0.01500)	(0.00365)	(0.00365)	0.02121	0.02121
CBRL	0.10175	0.10175	0.10514	0.10514	0.10789	0.10789
COST	0.14940	0.14940	0.14684	0.14684	0.14592	0.14592
CPWR	(0.07273)	(0.07273)	(0.03004)	(0.03004)	(0.00756)	(0.00756)
CRUS	0.18039	0.18039	0.17885	0.17885	0.17377	0.17377
CSCO	0.16645	0.16645	0.16244	0.16244	0.15610	0.15610
CTAS	0.10484	0.10484	0.10352	0.10352	0.10220	0.10220
DELL	0.08466	0.08466	0.06728	0.06728	0.02122	0.02122
EA	0.16023	0.16023	0.16066	0.16066	0.15940	0.15940
INTC	0.24333	0.24333	0.23768	0.23768	0.23850	0.23850
JBHT	0.07013	0.07013	0.06913	0.06913	0.07009	0.07009
KELYA	0.08042	0.08042	0.07727	0.07727	0.07687	0.07687
KLAC	0.21806	0.21806	0.21700	0.21700	0.21672	0.21672
LLTC	0.03401	0.03401	0.01554	0.01554	0.01883	0.01883
LRCX	0.28156	0.28156	0.27829	0.27829	0.27863	0.27863
MLHR	0.10019	0.10019	0.09388	0.09388	0.09048	0.09048
MOLX	0.04658	0.04658	0.03091	0.03091	0.03152	0.03152
MSFT	0.24456	0.24456	0.24090	0.24090	0.23876	0.23876

Figure 4.7: Comparison of prior and posterior returns on selected dates.

The model weights assigned by this implementation of the Black-Litterman algorithm are very similar to market weights. Figure 4.8 presents a graph of the market weights and the model weights assigned to each asset for the 30/06/2017. For each equity, the graph presents it's relative market weight and the weight the model assigns to it. For example, AAPL is the last point on the right in Figure 4.8, with market and model weights of 0.106. The securities are sorted so that market weights are in ascending order. The correlation coefficient for the two series on this date is 0.99. The average correlation coefficient for the two series for each of the 267 months from 31/07/1995 to 29/09/2017 was 0.99. As can be seen in Figure 4.1, market and model weights were virtually identical for almost all dates on which the model was run but there were exceptions: two weight series had correlation coefficients of 0.74 (on the 27/04/2004) and 0.82 (on the 31.03.2009). The weights calculated by the algorithm include negative weights representing short positions, while market weights are naturally only positive weights representing long positions. Nevertheless, they are valid solutions to the optimization problem posed to the Black-Litterman algorithm. There is therefore no evidence that the Black-Litterman implementation does not function correctly.

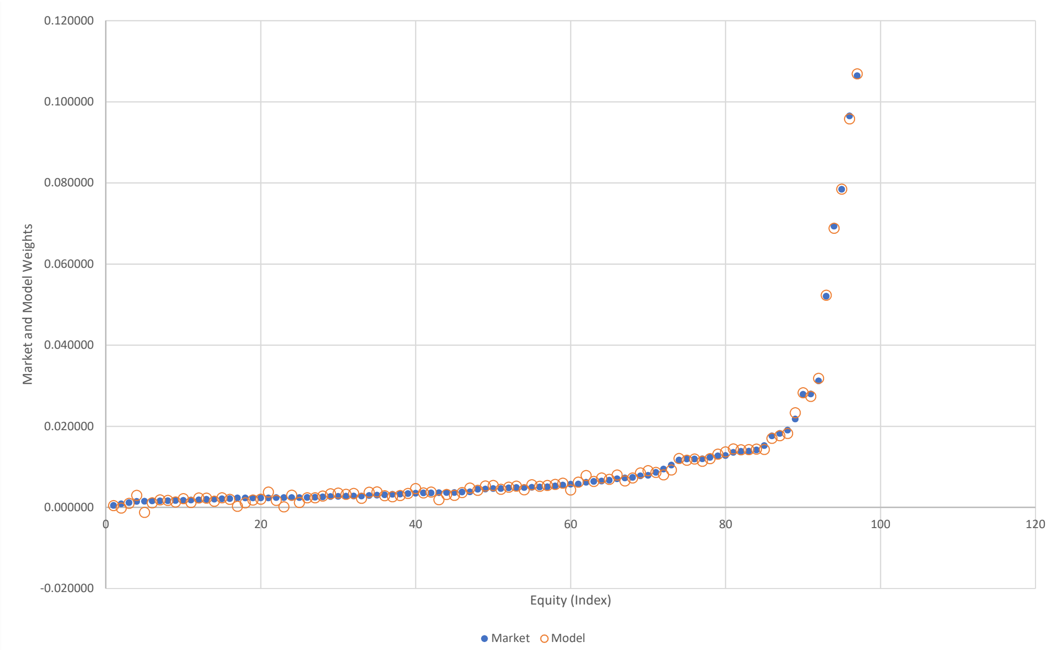


Figure 4.8: Comparison of market and model weights assigned to assets on 30/06/2017.

4.5 Discussion

This experiment developed the infrastructure for the second and third experiments.

A Django framework was developed which enables the deployment of the results of the research in the form of SAAS.

Two benchmarks were developed, one based on equal weights, and one based on a Black-Litterman algorithm without views. In the process, the various classes making up the scientific platform - in particular the Strategy and Strategy Manager classes - were implemented and tested. The testing process was performed in part through unit tests, and in part through the automatic production of Excel spreadsheets which will serve as audit trails of

Correlations	Count
0.74 - 0.76	1
0.76 - 0.78	0
0.78 - 0.80	0
0.80 - 0.82	1
0.82 - 0.84	0
0.84 - 0.86	0
0.86 - 0.88	0
0.88 - 0.90	2
0.90 - 0.92	0
0.92 - 0.94	0
0.94 - 0.96	1
0.96 - 0.98	8
0.98 - 1.00	254

Table 4.1: Correlations between model and market weights over entire period.

the complex data processes.

The tests which were performed did not uncover errors in the Black-Litterman algorithm. The model weights calculated were close to market weights, as expected, for almost all periods. The equilibrium market returns calculated by the model as priors were very close to the posterior returns, as expected as no views were provided to modify the posterior.

One operational problem was encountered in the implementation of the Black-Litterman algorithm: if there are more equities than data points, the covariance matrix is singular and cannot be inverted. This is a however a known result, and care must simply be taken to ensure there are more than 100 data points, or 9 years, for the calculation of the covariance matrix.

Chapter 5

Generating Views Using a Bayesian Network

This experiment investigates the use of Bayesian networks for asset management as input to a Black-Litterman model. Different approaches are compared, including the use of two different packages, discrete versus continuous distributions for random variables, and different functional forms for the relationship between factors and returns.

5.1 Introduction

The core aim of this research is to determine whether using Bayesian networks to develop views for a Black-Litterman asset allocation model may result in better investment portfolios. The objectives of this experiment are to develop Bayesian networks to generate views in a flexible way so that the networks may be modified as required.

5.2 Aims and Objectives

The overall aim of this experiment is to explore the use of Bayesian networks to generate views for a Black-Litterman asset allocation model.

The first objective will be to build and explore different functional forms for the relation between factors and returns for a Bayesian network in order to generate views for a Black-Litterman asset allocation model.

The second objective will be to develop tools to understand and evaluate the output of the Bayesian network.

5.3 Background

5.3.1 Bayesian Statistics

One of the main motivations for using Bayesian statistics is to obtain a measure of the uncertainty of predictions. Another is to enable the use of priors which can be used to express expert knowledge regarding the factor of interest.

Bayesian networks present other advantages as well. Whereas a model such as a neural network is opaque and works as a black box, the relative importance of factors in a Bayesian network can be identified. This lends the model a degree of transparency. Bayesian networks may also be extended without recalculating the entire graph if it is desirable to add independent factors to the graph.

5.3.2 Tools

Several options exist to model Bayesian networks using Python. Different packages focus on different aspects of Bayesian networks, including learning model structure and inference using either discrete or continuous distributions. The main task with which this research is concerned is inference, and three packages appear to be appropriate for this task on Python: PyStan, Edward, and PyMC3 . Either of these packages would appear to be suitable for the purposes of this research[1]. PyStan (Carpenter et al., 2017) is a python interface to Stan, a probabilistic programming language written in C++. It requires a C++ compiler at runtime, which slows execution initially but helps ensure faster calculations. Stan is stable and benefits from strong academic support. Edward (Tran et al., 2016) is a more recent effort. It has recently been included in TensorFlow Probability[2]. This would tend to suggest that it is future proof, however documentation is uneven at this point. PyMC3 is a more established package, with complete documentation and examples available on the web. It based on Theano (Bergstra et al., 2010), which appears to be discontinued[3], however the developers of PyMC3 have declared the future version, PYMC4, will be based on TensorFlow. Development work in this research is based on PyMC3 , mainly due to the impression that, as it is written in Python, it would be easier to learn and prototype with this module than it would be with PyStan, and it is more mature than TensorFlow Probability with better documentation available for it at this point in time.

PyMC3 version 3.5 on Windows 10 was used initially. This version appears to have a bug which inhibits working with several cores simultaneously. The same version of PyMC3 on linux run inside a virtual machine did not have this limitation and was therefore used for this experiment. For Experiment 3, PyMC3 was run on linux, on the UCL cluster.

A package that was also considered is pgmpy, based on (Koller and Friedman, 2009). Although this package is easy to use, inference is prohibitively slow beyond very basic networks, and customization of the functional form between factors and returns also did not appear possible. Initial work performed using pgmpy is presented in Appendix D.

[1]Pyro is a more recent alternative which appears promising (Bingham et al., 2019).

[2]https://medium.com/tensorflow/introducing-tensorflow-probability-dca4c304e245 last accessed 26.11.2019

[3]https://en.wikipedia.org/wiki/Theano_(software) last accessed 26.11.2019

5.3.3 Priors

5.3.3.1 Introduction

Prior probability distributions for the parameters of the Bayesian networks must be selected. In practice, investors may wish to define priors using their expert knowledge. It is also possible to define priors based on historical data. In this thesis, uninformative priors are selected, subject to constraints such as the avoidance of overflow errors.

5.3.3.2 Maximimum Entropy

The entropy of a continuous distribution p is defined as (Jaynes, 1968)

$$H(p) = -\int p(x)\log p(x)dx. \tag{5.1}$$

According to the maximum entropy principle, the prior distribution which maximizes Eq. (5.1) should be selected.

5.3.3.3 Uniform Distribution

On a finite interval $[a, b]$, the uniform distribution is the maximum entropy distribution (Park and Bera, 2009). The probability density function of the continuous uniform distribution for a variable x is given as (Hogg and Craig, 1978)

$$p(x; a, b) = \begin{cases} \frac{1}{b-a} & \text{for } a \leq x \leq b \\ 0 & \text{for } x < a \text{ or } x > b. \end{cases} \tag{5.2}$$

When a variable x is uniformly distributed on the interval $[a, b]$ the notation used is

$$x \sim U(a, b). \tag{5.3}$$

5.3.3.4 Normal Distribution

On the open interval the normal distribution is the maximum entropy distribution (Park and Bera, 2009). The probability density function of the continuous normal distribution for a variable x is given as (Hogg and Craig, 1978)

$$p(x; \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right). \tag{5.4}$$

When a variable x is normally distributed with mean μ and variance σ^2, the notation used is

$$x \sim N(\mu, \sigma^2). \tag{5.5}$$

5.3.3.5 Gamma Distribution

The gamma distribution is the maximum entropy distribution on a semi-infinite support with given expectation (Park and Bera, 2009). The probability density function of the gamma distribution for a variable x is given as (Hogg and Craig, 1978)

$$p(x;\alpha,\beta) = \begin{cases} \frac{\beta^\alpha x^{\alpha-1}\exp(-\beta x)}{(\alpha-1)!} & \text{for } x > 0 \\ 0 & \text{for } x \leq 0, \end{cases} \tag{5.6}$$

where $\alpha, \beta > 0$ are selected such that $E(x) = \frac{\alpha}{\beta}$. When a variable x is gamma-distributed, the notation used is

$$x \sim \Gamma(\alpha, \beta). \tag{5.7}$$

5.3.3.6 Folded Normal Distribution

In certain cases, all that is known is that the variable of interest does not take on negative values, although it may be zero. In this case, the folded or half-normal distribution was used. The probability density function in this case is given as (Leone, Nelson, and Nottingham, 1961)

$$p(x;\sigma) = \begin{cases} \frac{\sqrt{2}}{\sigma\sqrt{\pi}} \exp\left(-\frac{x^2}{2\sigma^2}\right) & \text{for } x \geq 0 \\ 0 & \text{for } x < 0. \end{cases} \tag{5.8}$$

When a variable x is distributed according to the folded normal distribution with mean $\mu = 0$ and variance σ^2, the notation used in this thesis is

$$x \sim |N(0, \sigma^2)|. \tag{5.9}$$

5.3.3.7 Bounded Normal Distribution

In certain cases, the normal distribution would have been used, but it was necessary to truncate it in order to avoid overflow or underflow errors. The probability density function of the bounded normal distribution used in this case is given as (Burkardt, 2014)

$$p(x;\mu,\sigma,a,b) = \begin{cases} \frac{1}{\sigma}\frac{\phi\left(\frac{x-\mu}{\sigma}\right)}{\Phi\left(\frac{b-\mu}{\sigma}\right)-\Phi\left(\frac{a-\mu}{\sigma}\right)} & \text{for } a \leq x \leq b \\ 0 & \text{for } x < a \text{ or } x > b \end{cases} \tag{5.10}$$

$$\phi(\nu) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}\nu^2\right) \tag{5.11}$$

$$\Phi(\nu) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\nu} \exp\left(-t^2/2\right) dt, \tag{5.12}$$

where $\phi(\nu)$ is the standard normal distribution, and $\Phi(\nu)$ its cumulative distribution function. When a variable x is distributed according to the bound normal distribution on the interval $[a, b]$ with mean μ and variance σ^2, the notation used in this thesis is

$$x \sim \Psi(\mu, \sigma^2, a, b). \tag{5.13}$$

5.3.4 Modelling returns in PyMC3

5.3.4.1 Introduction

Expected returns are modelled in a Bayesian network as a function of factors, which may include past returns and macro-economic factors. Various models exist to combine factors in order to reflect stylized statistical properties of asset returns (Cont, 2001). Some properties it may be desirable to model include heavy tails, time-varying volatility which has a degree of autocorrelation which leads to volatility clustering, as well as (negative) correlation between returns and volatility.

One interesting aspect of Bayesian networks is the ability to freely modify the models used at each node. Fabozzi, Focardi, and Kolm (2006) and Kita et al. (2012) present the following among many other potentially useful models. All these models may be adapted to include external regressors. In a Bayesian network, all variables and parameters are considered continuous random variables.

PyMC3 makes use of the MCMC methods presented in Section 2.2.6.

5.3.4.2 AR(p)

An AR(p) (auto regressive) model defines a time-varying random process as a linear combination of its own past values and a stochastic error term. This is useful to model time series which display autocorrelation, including equity returns over short periods (Cont, 2001), and inflation (Fuhrer, 2009)

$$r_t = \alpha + \sum_{i=1}^{p} \beta_i r_{t-i} + \epsilon_t \tag{5.14}$$

$$\epsilon_t \sim N(0, \sigma^2), \tag{5.15}$$

where σ is constant and $N(\mu, \sigma^2)$ is the normal distribution defined in Section 5.3.3.4. When $p = 1$, this reduces to AR(1)

$$r_t = \alpha + \beta r_{t-1} + \epsilon_t. \tag{5.16}$$

5.3.4.3 MA(q)

In the moving average regression model, returns are modelled based upon past error terms. This model is useful if random shocks, which have affected past returns, are assumed to affect current returns as well, for example through mean reversion if the parameters are negative

$$r_t = \alpha + \sum_{i=1}^{q} \beta_i \epsilon_{t-i} + \epsilon_t \tag{5.17}$$

$$\epsilon_t \sim N(0, \sigma^2), \tag{5.18}$$

where and $N(\mu, \sigma^2)$ is the normal distribution defined in Section 5.3.3.4.

5.3.4.4 ARMA(*p*,*q*)

The ARMA model combines both an AR(p) and MA(q) models. It can be useful if both returns and error terms are assumed to be autocorrelated

$$r_t = \alpha + \sum_{i=1}^{p} \beta_{1i} r_{t-i} + \sum_{j=1}^{q} \beta_{2j} \epsilon_{t-j} + \epsilon_t \tag{5.19}$$

$$\epsilon_t \sim N(0, \sigma^2), \tag{5.20}$$

where and $N(\mu, \sigma^2)$ is the normal distribution defined in Section 5.3.3.4.

5.3.4.5 ARCH(*q*)

ARCH models introduce time-varying volatility and allow volatility clustering (Engle, 1982). The model was first applied to estimates of the variance of UK inflation. It is specified as

$$r_t = \varepsilon_t \tag{5.21}$$

$$\sigma_t^2 = \omega + \sum_{i=1}^{q} \lambda_i \varepsilon_{t-i}^2 \tag{5.22}$$

$$\varepsilon_t = \sigma_t \epsilon_t \tag{5.23}$$

$$\epsilon_t \sim N(0, 1), \tag{5.24}$$

where $\lambda_i >= 0$ for all i and $N(\mu, \sigma^2)$ is the normal distribution defined in Section 5.3.3.4. Engle (1982) distinguishes between a constant unconditional variance, ω in Eq. (5.22) and conditional variance σ^2 which is conditional on past variance. Engle (1982) focusses discussion on the variance, Eq. (5.22).

For simplicity, this thesis focusses on $q = 1$, referred to here as an ARCH(1) model, where returns and variance depends only on the most recent value.

It may be useful to note that ϵ as defined in Eq. (5.20) and ε as defined in Eq. (5.23) have identical probability density functions due to the fact that the mean in both cases is zero, since $X \sim N(\mu, \sigma^2) \implies cX \sim N(c\mu, c^2\sigma^2)$. This can be seen in Figure 5.1. This figure is a histogram created using 100,000 random draws from two normal distributions, one $\sim N(0, \sigma^2)$ and the other standard normal, $\sim N(0, 1)$. The second was then scaled by σ. The value of σ used for illustration was 0.5, but the conclusion is valid for all values of σ.

5.3.4.6 ARCH-M(*p*,*q*)

ARCH in mean (Engle, Lilien, and Robins, 1987) reflects the expectation higher volatility results in higher returns. The model was originally applied to estimating risk premia in the

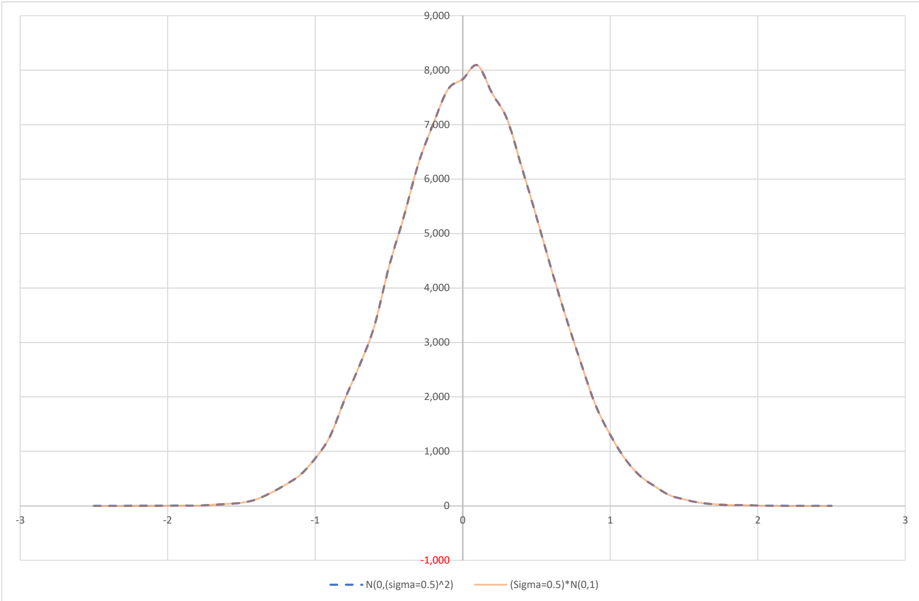


Figure 5.1: PDF of variables defined in Eqs. (5.20) and (5.23) for $\sigma = 0.5$.

term structure of interest rates. ARCH-M(p,q) is defined as

$$r_t = \alpha + \sum_{i=1}^{p} \beta_i \sigma_{t-i}^2 + \varepsilon_t \tag{5.25}$$

$$\sigma_t^2 = \omega + \sum_{i=1}^{q} \lambda_i \varepsilon_{t-i}^2. \tag{5.26}$$

5.3.4.7 GARCH(p,q)

The GARCH model (Bollerslev, 1986) is an extension to the ARCH model which is more flexible. The equation for variance in this model bears a passing resemblance to the ARMA model as it includes both an auto regressive term as a function of past variance, and a “moving average” term depending upon past (return) error terms

$$r_t = \varepsilon_t \tag{5.27}$$

$$\sigma_t^2 = \omega + \sum_{i=1}^{p} \lambda_{1i} \sigma_{t-i}^2 + \sum_{j=1}^{q} \lambda_{2j} \varepsilon_{t-j}^2 \tag{5.28}$$

$$\varepsilon_t = \sigma_t \epsilon_t, \tag{5.29}$$

where all λ’s are restricted to be non-negative. This research focusses on a GARCH(1,1) model.

5.3.4.8 GARCH-M(1,1)

GARCH in mean models (Engle and Bollerslev, 1986) add a variance term in the mean equation which again helps reflect the expectation that higher volatility will result in higher returns. This model was originally applied to estimating risk premiums in the foreign exchange markets

$$r_t = \alpha + \beta_1 r_{t-1} + \beta_2 \sigma_t^2 + \varepsilon_t \tag{5.30}$$

$$\sigma_t^2 = \omega + \lambda_1 \sigma_{t-1}^2 + \lambda_2 \varepsilon_{t-1}^2 \tag{5.31}$$

$$\varepsilon_t = \sigma_t \epsilon_t. \tag{5.32}$$

5.3.4.9 EGARCH(1,1)

GARCH models imply that past error terms and variance have the same impact, whether they arise from increasing or decreasing returns, which may not be the case (Nelson, 1991). EGARCH models (Nelson, 1991) were introduced to address this limitation

$$r_t = \alpha + \beta r_{t-1} + \varepsilon_t \tag{5.33}$$

$$\log(\sigma_t^2) = \omega + \lambda_1 \log(\sigma_{t-1}^2) + \lambda_2 \left[\left| \frac{\varepsilon_{t-1}}{\sigma_{t-1}} \right| - E\left(\left| \frac{\varepsilon_{t-1}}{\sigma_{t-1}} \right| \right) \right] + \lambda_3 \frac{\varepsilon_{t-1}}{\sigma_{t-1}} \tag{5.34}$$

$$\varepsilon_t = \sigma_t \epsilon_t, \tag{5.35}$$

where ϵ_t is IID with $\mathrm{E}(\epsilon_t) = 0$ and $\mathrm{Var}(\epsilon_t) = 1$. It is not necessarily normally distributed: Nelson (1991) discussed the Generalized Error distribution and the Student t distribution. If ϵ follows a Student t distribution with $\nu > 1$ degrees of freedom, the expectation is given by (Ahsanullah, Shakil, and Kibria, 2015)

$$E(|\epsilon_t|) = 2\sqrt{\frac{\nu}{\pi}} \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\Gamma\left(\frac{\nu}{2}\right)(\nu - 1)}. \tag{5.36}$$

5.3.4.10 EGARCH-M(1,1)

EGARCH-M as defined in Nelson (1991) includes the variance in the return equation to help reflect the expectation that higher risk could lead to higher returns. The full form used in this research, including external regressors is (Beach and Orlov, 2007)

$$r_t = \alpha + \beta_1 r_{t-1} + \beta_2 \sigma_t^2 + \boldsymbol{\beta}_3^\intercal \boldsymbol{Z}_{1t} + \varepsilon_t \tag{5.37}$$

$$\log(\sigma_t^2) = \omega + \lambda_1 \log(\sigma_{t-1}^2) + \lambda_2 \left[\left| \frac{\varepsilon_{t-1}}{\sigma_{t-1}} \right| - E\left(\left| \frac{\varepsilon_{t-1}}{\sigma_{t-1}} \right| \right) \right] + \lambda_3 \frac{\varepsilon_{t-1}}{\sigma_{t-1}} + \boldsymbol{\lambda}_4^\intercal \boldsymbol{Z}_{2t} \tag{5.38}$$

$$\varepsilon_t = \sigma_t \epsilon_t, \tag{5.39}$$

where ϵ_t is distributed as for EGARCH. $\boldsymbol{Z}_n$ are (arrays of) exogenous regressors. $\boldsymbol{\beta}_3$ and $\boldsymbol{\lambda}_4$ may also be arrays.

5.3.4.11 Theano

Theano[4] is a Python library used to optimize and evaluate multi-dimensional mathematical expressions at speeds similar to compiled C code. It is currently maintained by the PyMC3 team. Implementing Bayesian EGARCH-M using PyMC3 utilizes advanced functionality of Theano.

5.4 Design

5.4.1 Network

Figure 5.2 represents a Naive Bayesian network which forms the basis of this implementation. The model implies that the change in price of an asset in the next period, $N + 1$, is related to the change in price of the asset in period N and changes in macroeconomic data in period N. All factors are assumed to be conditionally independent. This assumption can be taken because data is available for all factors. This first model therefore excludes Industrial Production and Inflation, for which this independence assumption could be expected not to hold. Interest rates, for example, would not be expected to be independent of inflation or industrial production. The model also excludes the use of ETF's as factors, as data was only available from earliest 2004.

Table 3.2 presents the frequency of macroeconomic data. The holding period assumed in this experiment is one month, and all data would appear to be available on a monthly basis. As discussed in 3.2.2, however, data is sometimes released mid-month, and with considerable delay in relation to the period for which the data is calculated. If data is required on a more timely basis than it is available, it may be estimated. This estimation process may be modelled as additional dependencies within the Bayesian Net, by identifying parent or child factors related to the data to be estimated.

5.4.2 Models

It is expected that a complex model such as EGARCH-M will be necessary to provide a reasonable prediction of future returns. The development of this model will be done on an incremental basis, for two reasons. First, a complex model may, in fact, not be better than a simpler model. It is interesting and relevant to explore this. Second, it is desirable to test and debug the development of the model in an organized manner by introducing relevant technical difficulties gradually. Linear regression is straightforward to implement in PyMC3 . AR(1) (Eq. (5.16)) adds the problem of considering the history of returns in predicting future returns. GARCH (Eq. (5.28)) adds the complexity of including past standard deviation and past errors in returns, in the calculation of predicted standard deviation. EGARCH (Eq. (5.34)) adds the logarithm in the calculation of expected standard deviation, and EGARCH-M (Eq. (5.38)) is the final objective.

[4]http://deeplearning.net/software/theano/introduction.html

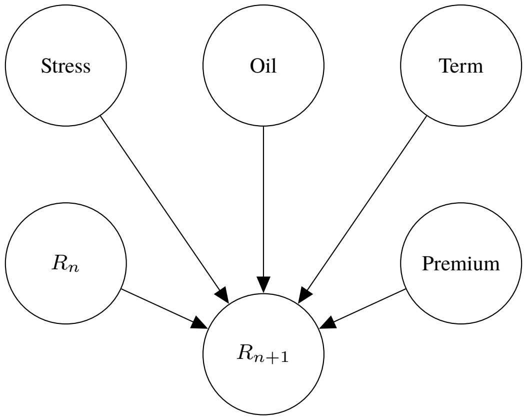


Figure 5.2: A Bayesian network is a directed acyclic graph where an absence of a connection between two nodes expresses an assumption of conditional independence. In a naive Bayesian network, all factors are assumed to be conditionally independent. This approach may be relevant if the values of all factors are expected to be known. Conditional dependence between factors may be introduced later if assumptions change or if scenario analysis is desired.

The following models will be implemented as Bayesian networks and the results compared.

- Linear regression
- AR(1)
- GARCH(1,1)
- EGARCH(1,1)
- EGARCH-M(1,1) assuming ε follows Normal distribution

PyMC3 includes basic implementations of AR(p) and GARCH(1,1). These models will nevertheless be implemented separately for three reasons. First, the PyMC3 implementations cannot be used with external regressors, as required in this research. Second, they are not structured in such a way that prediction is easy. Third, test data can be used to compare the output of new implementations with the standard ones PyMC3 provides in order to ensure that they work as expected.

5.5 Implementation

5.5.1 Introduction

The following models will attempt to find a relation between changes in oil prices, changes in credit spreads, changes in the term structure of interest rates, and changes in a stress index calculated by the FRED.

There is some evidence any relationships which may exist are not stable, and depend upon recent conditions (Yuhn, Kim, and Mccown, 2018). For illustration, Figure 5.3 graphs changes in factors to the returns on AAPL shares. Outliers are excluded in these graphs. Visually, it would appear plausible that, if there is a relationship between changes in the term structure of interest rates and returns on AAPL stocks for example, the two are positively correlated in the first period and negatively correlated in the second. Monthly data over a rolling period of 54 months is considered in this experiment. Over the period from 30/12/1994 to 31/05/1999, the relationship between the $n+1$ monthly returns of AAPL and factor returns for month n is presented in Figure 5.4. The period presented was a particularly interesting one, including the Asian financial crisis in 1997, the breakdown of Long-Term Capital Management (LTCM) in 1998, the continued buildup of the dot-com bubble which collapsed in 2000, and the financial crisis of 2008. In many cases, as can be seen for example in Figure 5.4, there are outliers which may need to be taken into consideration.

Implementing models in PyMC3 is complicated by several factors.

PyMC3 , using Theano, first creates a computational graph compiled to C code, and then uses this graph with the data. In this process, it is not straightforward to trace the value of variables during computation. It is not possible to print the values of variables during computation. It is not possible to insert break points in this process. The following approaches were used to work around this situation[5].

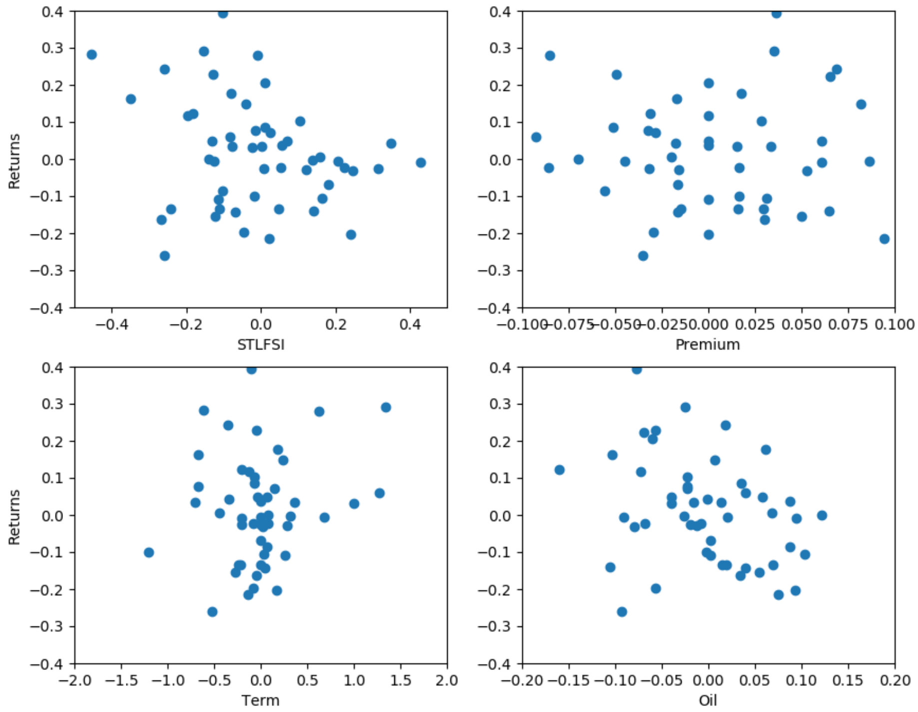
When computation is not halted, but the model is simply not converging - or not converging to known parameter values - then it is possible to create a type of audit trail by making the output of the computational graph include an array of *all* intermediate values. In this way, it is possible to trace the values of variables during computation.

When computation is halted, however, this approach cannot be used. In some cases, PyMC3 fails with a cryptic message relating to *Bad Initial Energy* or *Maxx matrix contains zeros on the diagonal*. Simulations in Excel using data which seems to be related to the errors is useful in identifying specific causes. It was found that these errors are generally related to two different issues: overflow (or underflow) and problems with data.

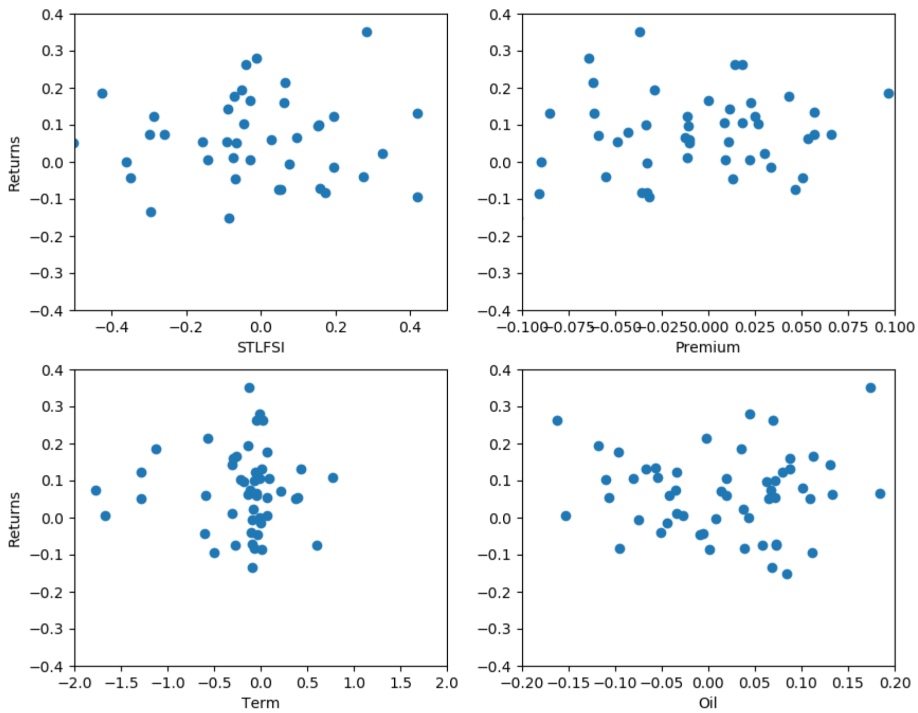
Overflow errors were found to occur in models which require logs or exponentials. Experimentation in Excel can help identify the range to which prior values must be constrained to avoid overflow or underflow errors. Priors can then be adjusted accordingly, for example to a Uniform distribution over the range of acceptable values. This generally did not improve inference, but it did avoid overflow errors. Priors are discussed in more detail in Secton 5.3.3

Problems related to data were found to occur when data was missing. These data points are converted to "NaN", or "Not a number", which causes PyMC3 to fail. Once these data points have been identified, they can be adjusted to useable values. In this experiment, missing data for changes in the value of factors or stock prices was converted to 0.

[5]docs.pymc.io/notebooks/variational_api_quickstart.html#Tracking-parameters presents an alternative approach to tracing parameter values.



(a) Period 31/12/1994 to 30/06/1999.



(b) Period 30/04/2003 to 31/10/2007.

Figure 5.3: Changes in factors against returns of AAPL, for two different periods. Any simple linear relationships which may appear in the first set of graphs is not observed in the second set. This suggests that relationships change over time. Some outliers have been removed for clarity.

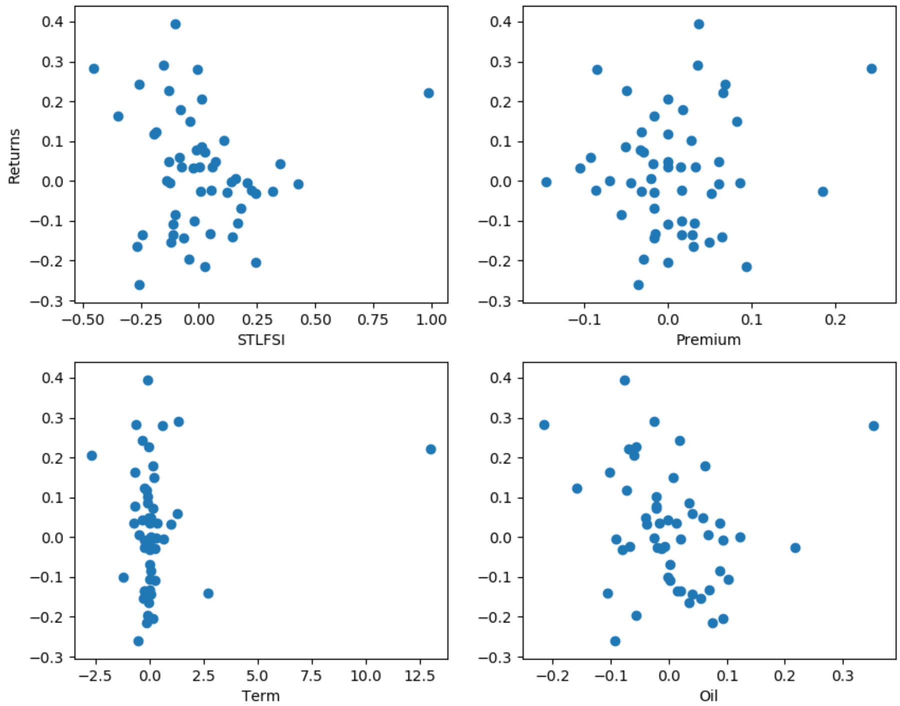


Figure 5.4: Changes in factors vs next period returns of AAPL stock for the period 31/12/1994 to 31/05/1999.

Computation times are relatively long, taking from around 15 to 75 minutes for one data point. For this reason, creating test cases is not always a practical approach. It was, however, found that initial testing and model construction could be performed using smaller sampling sizes. It was found, for example, that if using 4 chains of 1000 samples + 500 samples for tuning did not yield somewhat reasonable results, larger sampling sizes of 4 chains of 20,000 samples each generally did not either.

Initialization of models using ADVI rather than training using NUTS was found to provide results which were closer to known true values when testing models using synthetic data. One drawback of this approach is that it requires more time. ADVI also sometimes results in *Bad Energy* errors while training with NUTS did not result in errors.

Finally it is in the end time saving to work very progressively, using models which are well understood and implemented in Excel to create synthetic data, and introducing new technical difficulties one at a time.

5.5.2 Testing

One way of testing whether a model has been implemented correctly is to construct a series using a predefined process with known parameter values, and see whether the model can return the parameters and make predictions which are not unreasonable. It is straightforward to create an artificial time series which follows, for example, an AR(1) or an EGARCH-M process with external regressors in Microsoft Excel.

If the model is implemented correctly and for it to be useful, it should return the parameter values similar to those used in Excel to create the time series, and it should make reasonable predictions. The new model implementations are therefore tested by producing appropriate time series in Microsoft Excel using known parameters, and ensuring that the new models can recover the known model parameters.

The models are trained on 99 data points, and the resulting parameter estimates used to predict the 100th data point. The parameter values as well as the prediction are compared to the known values.

5.5.3 Linear Regression

Implementation of linear regression is the first model on which others are built. The relationship between the exogenous regressors, X, and asset returns, r, is modelled as

$$r \sim N(\mu, \sigma^2) \tag{5.40}$$

$$\mu_t = \alpha + \sum_i \beta_{it} X_{it} \tag{5.41}$$

$$\alpha \sim N(0, 0.2) \tag{5.42}$$

$$\beta_i \sim N(0, 0.2) \tag{5.43}$$

$$\sigma \sim |N(0,1)|, \tag{5.44}$$

where $|N(0,1)|$ is a folded normal distribution defined in Section 5.3.3.6, $N(\mu, \sigma^2)$ is the normal distribution defined in Section 5.3.3.4, $i = \{0, 1, 2, 3\}$, and the given values (0 and 0.2 for example) are the priors provided to the model. The normal distribution is used for simplicity and could easily be replaced by, for example, a t-distribution to better represent the heavier tails observed in the distribution of actual returns.

The model was implemented using monthly data from 30.11.1994 to 30.09.2017. Predictions were performed for each month from 30.07.1999 to 30.09.2017. Each prediction was performed using a model which was trained on the latest 54 months. 6 chains of 5000 samples each were generated using MCMC, for a total of 30,000 samples. An additional 1000 samples per chain were generated for tuning. These samples were discarded.

Training in PyMC3 results in a "trace" variable. It contains an array of all values of the parameters α, β_i, and σ which are the result of the sampling in PyMC3 . 500 sets of parameter values are drawn from the 30,000 useful samples. Posterior prediction is performed by using the latest factor values applied to each parameter value set, to provide 500 predictions of future returns. The mean and standard deviation of these predictions is taken as the final prediction. Results are presented in Section 5.6.1.

5.5.4 AR(1)

Implementation of an autoregressive model (AR) is interesting for two reasons. First, AR introduces dependence of expected returns on past returns. Second PyMC3 contains a basic implementation of an AR model which does not use external regressors, but which provides a benchmark for a first implementation of an AR model which can be extended to include external regressors.

A basic custom implementation of AR(1) was first developed and compared to the default AR(1) implementation which is included in PyMC3 . This custom implementation was further developed to include external regressors and validated using synthetic data. This second implementation forms the basis for GARCH and EGARCH-M implementations in the following sections.

The relationship between asset returns was first modelled as

$$r_t = \beta r_{t-1} + \epsilon_t \tag{5.45}$$

$$\beta \sim N(0, 2) \tag{5.46}$$

$$\sigma \sim |N(0, 1)| \tag{5.47}$$

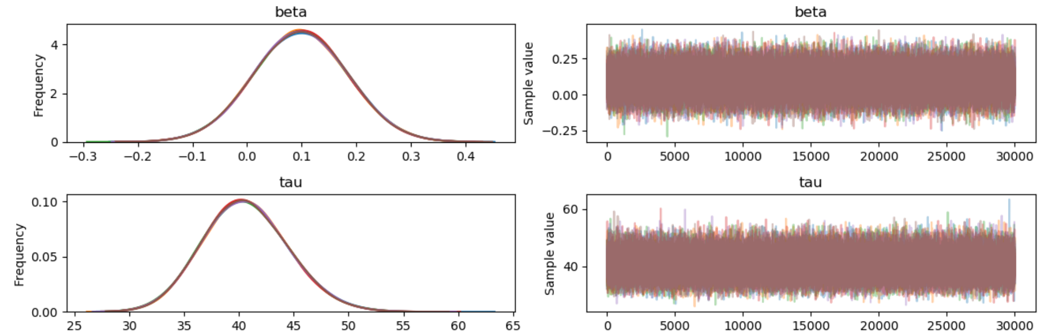
$$\epsilon_t \sim N(0, \sigma^2), \tag{5.48}$$

where $|N(0, 1)|$ is a folded normal distribution defined in Section 5.3.3.6 and $N(\mu, \sigma^2)$ is the normal distribution defined in Section 5.3.3.4. A uniform distribution could have been used for β, as the value of this parameter could be expected to lie in the open interval $(-1, 1)$ however it is not inconceivable that β would lie outside this interval for a short period. The consequence may be a rapidly increasing or decreasing geometric series which may be counter intuitive but not impossible. Use of the normal distribution was actually found to make no difference in practice and was retained for later, purely technical, convenience.

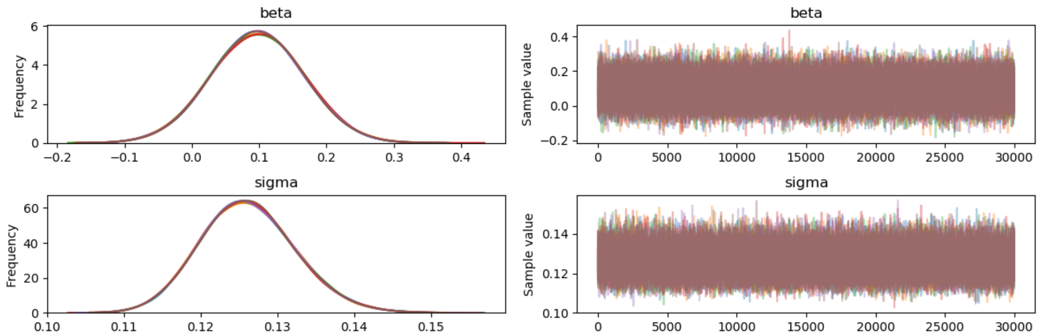
PyMC3 defines AR(1) in terms of $\tau \sim \Gamma(1, 1)$ where $\tau = \frac{1}{\sigma^2}$. The two models were run once in six chains of 30,000 samples each on monthly data from 31.01.1995 to 31.05.1999. The two models both converge, mix well, and were found to agree on a value for β as presented in Figure 5.5 and summarized in Table 5.1. The model was further enhanced to include exogenous regressors. The relationship between asset returns and exogenous

Model	Beta	Sigma	Rhat
PyMC3	0.096030	0.084716	0.999986
Custom implementation	0.096196	0.068088	0.999994

Table 5.1: PyMC3 AR(1) compared to custom implementation: key statistics for the value of β. The custom implementation of AR(1) estimates the same value of β and a similar value of σ as the implementation of AR1 which is supplied as part of PyMC3 .



(a) Default PyMC3 version of AR(1)



(b) Custom implementation of AR(1)

Figure 5.5: The left hand side of the graphs shows that all chains of both models converge to the same distributions. The second graphs in (a) and (b) are different because PyMC3 defines AR(1) in terms of $\tau \sim \Gamma(1, 1)$ where $\tau = \dfrac{1}{\sigma^2}$. The right hand side of the graphs shows that the two models mixed well and that the parameter space was evenly searched. The two models were found to agree on a value for β.

regressors was modelled as

$$r_t = \beta_0 r_{t-1} + \sum_{i=1} \beta_i f_i + \epsilon_t \tag{5.49}$$

$$\beta_i \sim N(0, 2) \tag{5.50}$$

$$\sigma \sim |N(0, 1)| \tag{5.51}$$

$$\epsilon_t \sim N(0, \sigma^2), \tag{5.52}$$

where $\boldsymbol{f}$ are exogenous regressors, $|N(0, 1)|$ is a folded normal distribution defined in

Parameter	Actual Value	Estimate Mean	Estimate Std Dev.
β_0	0.750	0.782	0.023
β_1	2.000	1.984	0.066
β_2	-0.500	-0.511	0.028
σ^2	0.150	0.173	0.013
x_{i+1}	1.763	1.816	0.172

Table 5.2: AR(1) training and prediction using synthetic data. The estimated mean value of all parameters are very close to their true known values.

```
with pm.Model() as ar_model:
    beta0 = pymc3.Normal('beta0', 0, 10)
    beta1 = pymc3.Normal('beta1', 0, 10)
    beta2 = pymc3.Normal('beta2', 0, 10)
    stdev = pymc3.HalfNormal('stdev',sd=10)
    X_im1 = theano.shared(x_im1.values)
    X_i = theano.shared(x_i.values)
    R1_im1 = theano.shared(r1_im1.values)
    R2_im1 = theano.shared(r2_im1.values)
    mu = X_im1 \times beta + R1_im1 \times beta1 + R2_im1 \times beta2
    obs = pymc3.Normal('obs', mu=mu, sd=stdev, observed=X_i)
    ar_trace=pymc3.sample(draws=20000,tune=5000,cores=4,chains=4)
```

Listing 5.1: Implementation of AR1 with regressors

Section 5.3.3.6, and $N(\mu, \sigma^2)$ is the normal distribution defined in Section 5.3.3.4. A simplified implementation of AR1 with regressors is presented in Listing 5.1. Following the procedure discussed in Section 5.5.2, this implementation was tested on synthetic data created in Excel. The real process for the synthetic data is

$$x_t = \beta_0 x_{t-1} + \beta_1 f_{1t-1} + \beta_2 f_{2t-1} + \epsilon_t \tag{5.53}$$

$$\epsilon_t \sim N(0, \sigma^2) \tag{5.54}$$

$$f_1 \sim N(0.5, 0.25) \tag{5.55}$$

$$f_2 \sim N(1.5, 0.25), \tag{5.56}$$

where $\boldsymbol{f}$ are exogenous regressors, $N(\mu, \sigma^2)$ is the normal distribution defined in Section 5.3.3.4, and the parameters are as defined in Table 5.2. Results of the training and prediction are presented in Table 5.2 which shows that the actual values for all parameters as well as the 100th data point x_{i+1} are within two standard deviations of the model estimates.

The model was implemented using monthly data from 30.11.1994 to 30.09.2017. Predictions were performed for each month from 30.07.1999 to 30.09.2017. Each prediction was performed using a model which was trained on the latest 54 months. 4 chains of 20000 samples each were generated using MCMC, for a total of 80,000 samples. An additional

Model	ω	λ_1	λ_2
PyMC3	0.003	0.098	0.131
Custom implementation	0.003	0.098	0.131
True value	0.05	0.1	0.15

Table 5.3: A flexible custom implementation of GARCH(1,1) recovered the same parameter mean values as PyMC3 when presented with synthetic data series. These parameter values are different from the known true values due to the random innovation term in the synthetic data.

5000 samples per chain were generated for tuning. These samples were discarded. One single month for one stock, including prediction, took approximately 1.5 minutes on the computer used for testing.

Training in PyMC3 results in a "trace" variable. It contains an array of all values of the parameters α, β_i, and σ which are the result of the sampling in PyMC3 . 5000 sets of parameter values are drawn from the 80,000 useful samples. Posterior prediction is performed by using the latest factor values applied to each parameter value set, to provide 5000 predictions of future returns. The mean and standard deviation of these predictions is taken as the final prediction. Results are presented in Section 5.6.2.

5.5.5 GARCH(1,1)

Implementing a GARCH model with regressors in PyMC3 is interesting for two reasons. First, it introduces a technical difficulty as the calculation of current volatility depends upon past volatility calculations as well as past errors in returns. Second, PyMC3 includes a basic implementation of GARCH(1,1) which does not allow external regressors, but which can provide a benchmark for a first implementation of a GARCH(1,1) model which can be extended to include external regressors.

A basic custom implementation of GARCH(1,1) was developed and tested as described in Section 5.5.2. The model described in Section 5.3.4.7 was specified as

$$\lambda_1, \lambda_2 \sim |N(0,1)| \tag{5.57}$$

$$\sigma \sim |N(0,1)| \tag{5.58}$$

$$\epsilon_t \sim N(0,\sigma^2), \tag{5.59}$$

where $|N(0,1)|$ is a folded normal distribution defined in Section 5.3.3.6, and $N(0,\sigma^2)$ is the normal distribution defined in Section 5.3.3.4. Results are summarized in Table 5.3. As the custom implementation can be considered a more flexible reformulation of the PyMC3 implementation of GARCH(1,1), it is not unexpected that the results are very close for these two implementations. The model described in Section 5.3.4.7 was further enhanced

Parameter	True Value	Estimated Mean	Standard Deviation	Number of SD From True Value
α	0.020	0.053	0.027	-1.194
β_0	0.050	-0.094	0.110	1.305
β_1	0.700	0.756	0.191	-0.293
ω	0.020	0.035	0.014	-1.022
λ_1	0.500	0.333	0.162	1.030
λ_2	0.150	0.268	0.164	-0.720
x_{i+1}	-0.088	0.011	0.108	0.813

Table 5.4: The values of all parameters to a synthetic data series generated using exogenous regressors were estimated using Bayesian GARCH(1,1). The mean estimates were within 1.5 standard deviations of the known true values. The predicted value of x_{i+1} was within one standard deviation of the true value.

to include external regressors in the returns formula, modelled as

$$r_t = \alpha + \beta_0 r_{t-1} + \beta_1 f_1 + \epsilon_t \tag{5.60}$$

$$\alpha \sim N(0.5, 1) \tag{5.61}$$

$$\beta_i \sim N(0.5, 1) \tag{5.62}$$

$$\omega \sim N(0.5, 0.2) \tag{5.63}$$

$$\lambda_i \sim N(0.5, 0.2) \tag{5.64}$$

$$\varepsilon_t \sim N(0, \sigma^2), \tag{5.65}$$

where f_1 is an exogenous regressor and $N(\mu, \sigma^2)$ is the normal distribution defined in Section 5.3.3.4. This implementation was tested following the procedure described in Section 5.5.2. Results of the training and prediction are presented in Table 5.4 which shows that the actual values for all parameters as well as the 100^{th} data point are within two standard deviations of the model estimates. The model was trained on historical stock data. The results are presented in Section 5.6.3.

5.5.6 EGARCH(1,1)

EGARCH is not implemented as part of PyMC3 . Implementation of the model is therefore interesting in its own right. It introduces the use of logarithms, which gave rise to issues discussed in Section 5.5.1.

An implementation of EGARCH(1,1) described in Section 5.3.4.9 was developed and tested

Parameter	True Value	Estimated Mean	Standard Deviation	Number of SD From True Value
α	0.020	0.010	0.016	0.620
β_0	0.003	-0.006	0.028	0.318
β_1	0.100	-0.013	0.098	1.153
β_2	0.150	0.134	0.111	0.148
β_3	0.200	0.178	0.097	0.225
β_4	0.400	0.454	0.107	-0.507
ω	4.000	3.486	1.159	0.443
λ_1	-0.040	0.001	0.283	-0.146
λ_2	0.002	0.364	0.809	-0.447
λ_3	0.500	0.104	0.777	0.509
x_{i+1}	-0.278	-0.058	0.126	-1.744

Table 5.5: The values of all parameters to a synthetic data series generated using exogenous regressors were estimated using Bayesian EGARCH(1,1). The mean estimates were within 1.5 standard deviations of the known true values. The predicted value of x_{i+1} was within two standard deviations of the true value.

as described in Section 5.5.2. The model was specified as

$$\alpha \sim \Psi(0, 1, -0.5, 0.5) \tag{5.66}$$

$$\beta_i \sim \Psi(0, 1, -1.0, 1.0) \tag{5.67}$$

$$-\omega \sim \Gamma(\alpha = 2, \beta = 0.5) \tag{5.68}$$

$$\lambda_1 \sim N(0, 0.01) \tag{5.69}$$

$$\lambda_i \sim N(0, 1) \text{ (for i = \{2, 3\})} \tag{5.70}$$

$$-\log(\sigma^2) \sim \Gamma(\alpha = 2, \beta = 0.5) \tag{5.71}$$

$$\epsilon_t \sim N(0, \sigma^2), \tag{5.72}$$

where $\Psi(\mu, \sigma^2, a, b)$ is a bounded normal distribution defined in Section 5.3.3.7, $N(\mu, \sigma^2)$ is the normal distribution defined in Section 5.3.3.4, and $\Gamma(\alpha, \beta)$ is the Gamma distribution defined in Section 5.3.3.5. The Gamma distribution is strictly positive. As $0 < \sigma < 1$ however, $\log(\sigma^2) < 0$. Therefore the values of $-\omega$ and $-\log(\sigma^2)$ are those modelled using the Gamma distribution.

Results are summarized in Table 5.5. The actual values for all parameters as well as the 100^{th} data point are within 1.5 standard deviations of the model estimates.

The model was trained on historical stock data. The results are presented in Section 5.6.4.

Parameter	True Value	Estimated Mean	Standard Deviation	Number of SD From True Value
α	0.020	0.011	0.015	0.604
$\beta_0 0$	0.100	0.076	0.060	0.404
$\beta_0 1$	0.750	0.895	0.811	-0.179
β_1	0.010	0.072	0.092	-0.673
β_2	0.200	0.082	0.084	1.405
β_3	0.500	0.459	0.083	0.488
β_4	0.700	0.784	0.096	-0.868
ω	3.000	3.076	0.423	-0.180
λ_1	0.300	0.244	0.095	0.591
λ_2	-0.250	0.003	0.218	-1.158
λ_3	2.000	1.625	0.681	0.551
x_{i+1}	-0.242	-0.013	0.179	-1.281

Table 5.6: The values of all parameters to a synthetic data series generated using exogenous regressors were estimated using Bayesian EGARCH-M(1,1). The mean estimates were within 1.5 standard deviations of the known true values. The predicted value of x_{i+1} was also within 1.5 standard deviations of the true value.

5.5.7 EGARCH-M(1,1)

An implementation of EGARCH-M(1,1) described in Section 5.3.4.10 was developed and tested as described in Section 5.5.2. The model was specified as

$$\alpha \sim N(0, 0.02) \tag{5.73}$$

$$\beta_i \sim N(0, 1) \tag{5.74}$$

$$-\omega \sim \Gamma(\alpha = 2, \beta = 0.5) \tag{5.75}$$

$$\lambda_1 \sim U(0, 0.9) \tag{5.76}$$

$$\lambda_2 \sim U(-0.5, 0.5) \tag{5.77}$$

$$\lambda_3 \sim \Gamma(\alpha = 2, \beta = 0.5) \tag{5.78}$$

$$-\log(\sigma^2) \sim \Gamma(\alpha = 2, \beta = 0.5) \tag{5.79}$$

$$\epsilon_t \sim N(0, \sigma^2), \tag{5.80}$$

where $U(a, b)$ is a uniform distribution defined in Section 5.3.3.3, $N(\mu, \sigma^2)$ is the normal distribution defined in Section 5.3.3.4, and $\Gamma(\alpha, \beta)$ is the Gamma distribution defined in Section 5.3.3.5. As $0 < \sigma < 1$, $\log(\sigma^2) < 0$, the values of $-\omega$ and $-\log(\sigma^2)$ are those modelled using the Gamma distribution.

The priors were selected in order to allow for a range of values of σ which correspond to observed market values, very roughly between 2% and 30%.

Results are summarized in Table 5.6. The actual values for all parameters as well as the 100^{th} data point are within two standard deviations of the model estimates.

Data	Real Data	Prediction
Mean	0.030	0.034
Standard Distribution	0.122	0.049

Table 5.7: This table presents the first two moments of the distributions of real and predicted returns of the linear regression model over the same period. While both distributions have similar means, the standard deviation of real returns is over 12%, while the standard deviation of predicted returns is only 4.9%.

The model was trained on historical stock data. The results are presented in Section 5.6.5.

5.6 Results

5.6.1 Linear regression

The model specified in Section 5.5.3 was implemented using monthly data covering the period 31.11.1994 to 30.09.2017.

The traceplot for the first prediction is presented in Figure 5.6. The traceplot has one row for each parameter α, β_i, and σ. The left side of the traceplot presents the distribution of the parameter. For each of these graphs, each line represents one chain. The right hand side of the traceplot is used to evaluate whether the MCMC model has explored the parameter space adequately. In this case, all chains converge to the same distribution, and all seem to have explored the parameter space evenly.

Figure 5.7 compares all predicted returns to actual realized returns. In most cases, actual realized returns are within one standard deviation of predicted returns, however realized returns clearly show much more variation than predicted returns. For example, several consecutive months in 2008 have very similar and stable predictions, while actual realized returns for those months are either very positive or very negative.

Table 5.7 presents the first two moments of the distributions of real and predicted returns over the same period. While both distributions have similar means, the standard deviation of real returns is over 12%, while the standard deviation of predicted returns is only 4.9%.

The Kolmogorov-Smirnov test (Simard and L'Ecuyer, 2011) is used to test whether two samples come from the same distribution. The result of this test yields a p-value of the order of 10^{-6}, resulting in the rejection the null hypothesis, at 5% significance level, that the two samples do in fact come from the same distribution.

Even if individual predictions different from realized returns, perhaps the errors cancel each other out over longer periods, and predictions could be useful. Comparing the performance of a notional portfolio having returns equal to the predictions, to a portfolio with returns equal to realized returns may help investigate this question.

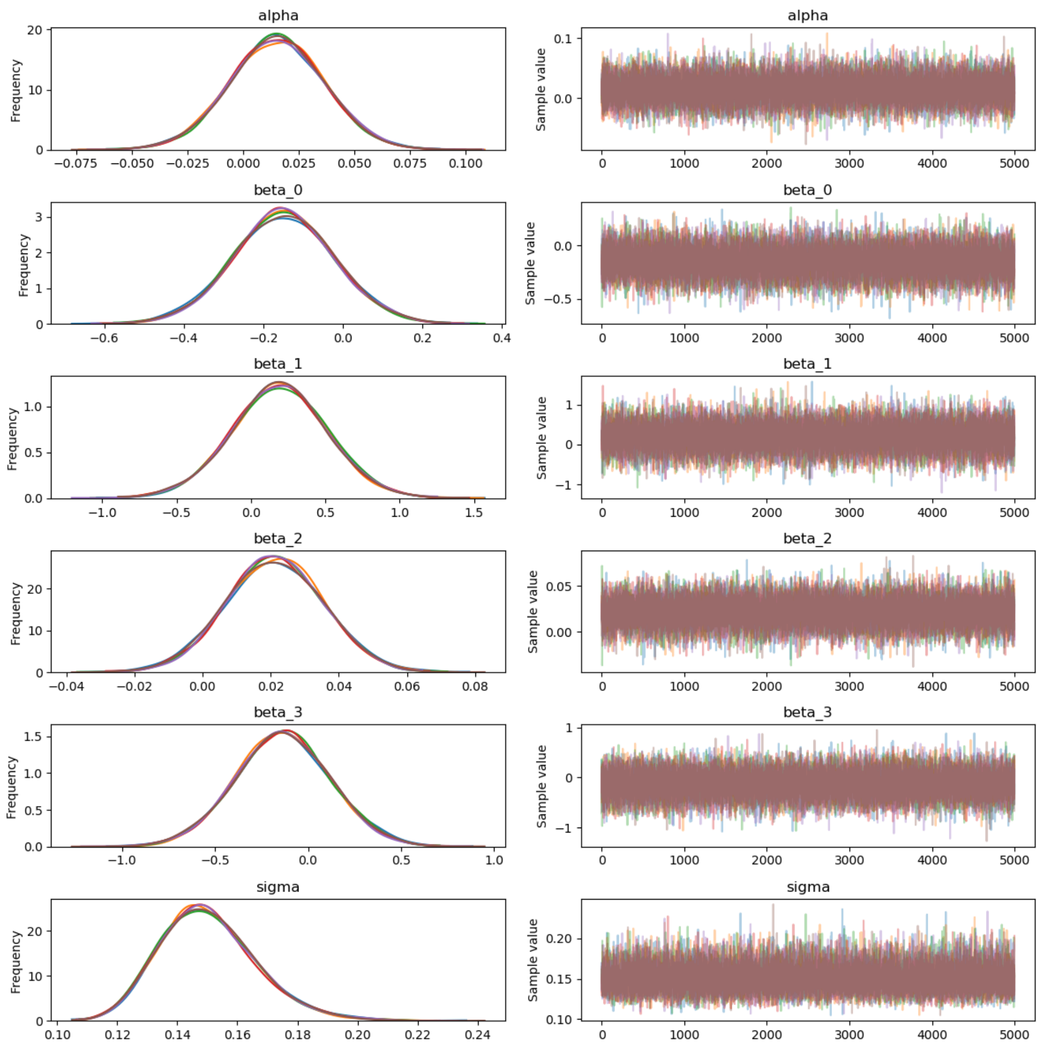


Figure 5.6: The traceplot related to the training of the linear regression model for the first data point, on 30.07.1999. The left hand side of the figure shows that all chains converged to the same distribution for the parameter values. The right hand side of the traceplot shows the MCMC model has explored the parameter space evenly.

As the period used in this research included at least two unusual sub-periods: the dot-com bubble which ended in 2000, and the financial crisis of 2008, Figure 5.8 compares the relative performance of a real and a notional portfolio, breaking down the investment period by resetting the value of the portfolio to 100 USD on 31.12.2002, 31.12.2007, and 31.12.2009 in order to explore the relative performance during these periods. This breakdown would suggest that there are in fact no periods during which the simple linear regression model based on the factors used in this section would provide predicted returns which would be useful in a Black-Litterman model.

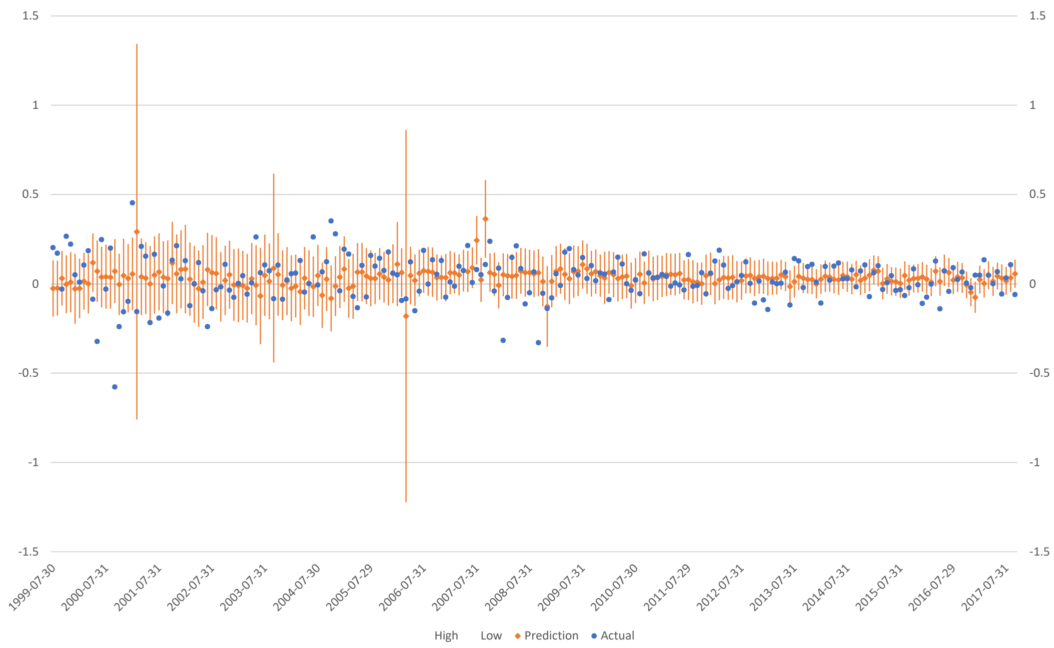


Figure 5.7: This chart compares all predicted returns to actual realized returns for the linear regression model from 30.07.1999 to 30.09.2017, with bars indicating +/- 1 standard deviation. In most cases, actual realized returns are within one standard deviation of predicted returns, however realized returns clearly show much more variation than predicted returns. For example, several consecutive months in 2008 have very similar and stable predictions, while actual realized returns for those months are either very positive or very negative.

Data	Real Data	Prediction
Mean	.030	.035
Standard Distribution	0.122	0.046

Table 5.8: This table presents the first two moments of the distributions of real and predicted returns of the AR(1) model over the same period. While both distributions have similar means, the standard deviation of real returns is over 12%, while the standard deviation of predicted returns is only 4.6%.

5.6.2 AR1

The model specified in Section 5.5.4 was implemented using monthly data covering the period 31.11.1994 to 30.09.2017.

The traceplot for the first prediction is presented in Figure 5.9. For all parameters, the traceplot shows that the parameter space has been search evenly, and that all chains converge to similar distributions.

Figure 5.10 compares all predicted returns to actual realized returns. In most cases, actual realized returns are within one standard deviation of predicted returns, however realized returns again show much more variation than predicted returns.

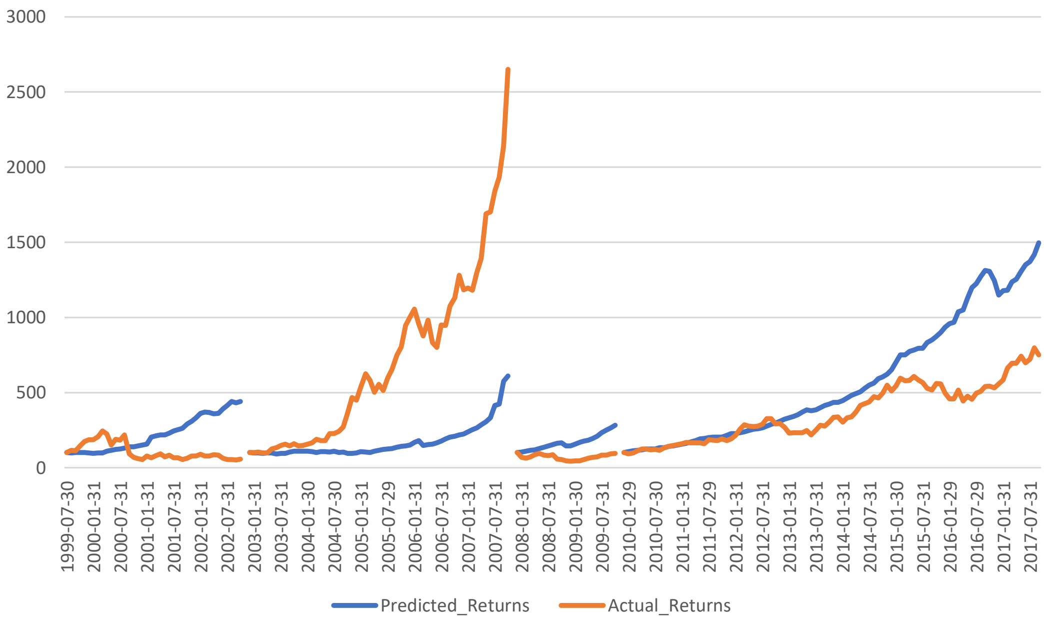


Figure 5.8: This graph compares the relative performance of a real and a notional portfolio, breaking down the investment period by resetting the value of the portfolio to 100 USD on 31.12.2002, 31.12.2007, and 31.12.2009 in order to explore the relative performance during these periods. This breakdown would suggest that there are in fact no periods during which the simple linear regression model based on the factors used in this section would provide predicted returns which would be useful in a Black-Litterman model.

Table 5.8 presents the first two moments of the distributions of real and predicted returns over the same period. While both distributions have similar means, the standard deviation of real returns is over 12%, while the standard deviation of predicted returns is only 4.6%.

The result of the Kolmogorov-Smirnov test again yields a p-value of the order of 10^{-6}, resulting in the rejection the null hypothesis, at 5% significance level, that the two samples do in fact come from the same distribution.

Figure 5.11 compares the relative performance of a real and a notional portfolio, breaking down the investment period by setting the value of the portfolio to 100 USD on 31.12.2002, 31.12.2007, and 31.12.2009 in order to explore the relative performance during these periods. This breakdown would suggest that there are in fact no periods during which the AR(1) model based on the factors used in this section would provide predicted returns which would be useful in a Black-Litterman model.

5.6.3 GARCH(1,1)

The model specified in Section 5.5.5 was implemented and trained on monthly data from 30.11.1994 to 30.09.2017. Predictions were performed for each month from 30.07.1999 to 30.09.2017. Each prediction was performed using a model which was trained on the latest 54 months. 4 chains of 6,000 samples each were generated, for a total of 24,000 samples.

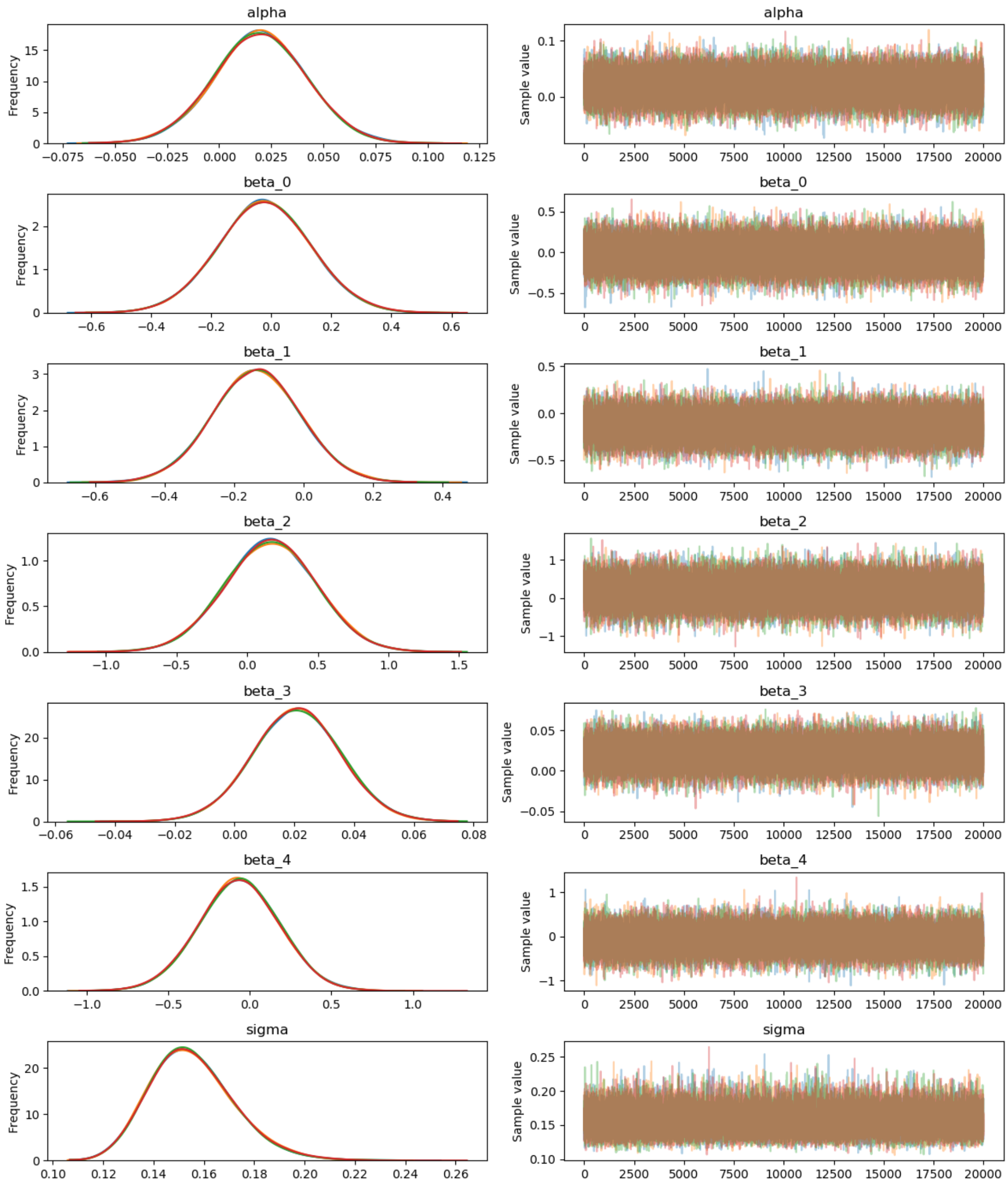


Figure 5.9: The traceplot related to the training of the AR(1) model for the first data point, on 30.07.1999. The left hand side of the figure shows that all chains converged to the same distribution for the parameter values. The right hand side of the traceplot shows the MCMC model has explored the parameter space evenly.

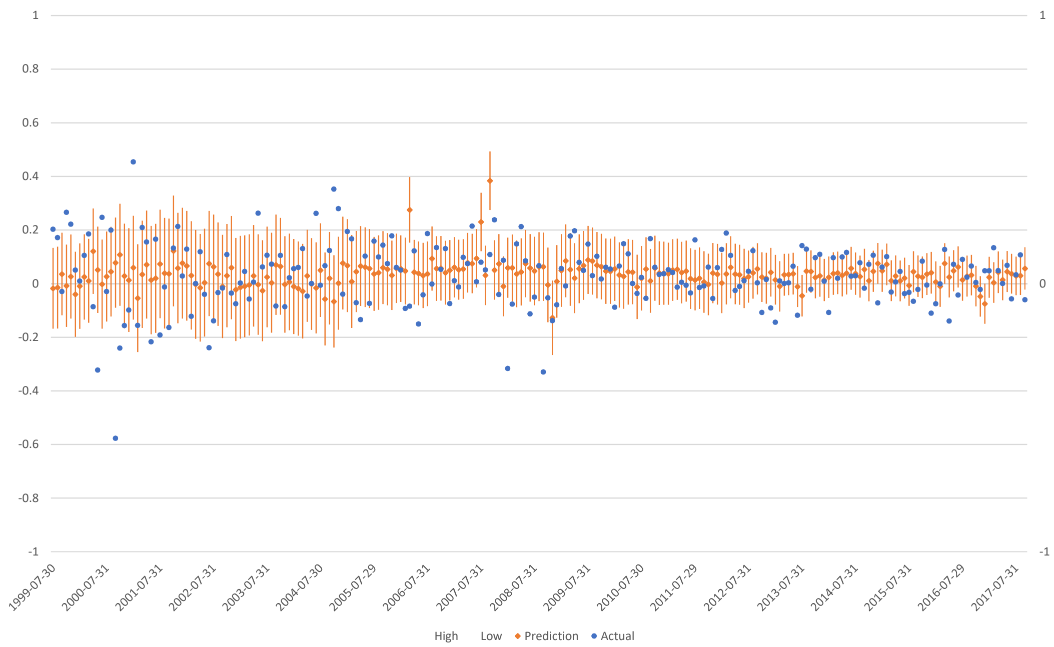


Figure 5.10: This chart compares all predicted returns to actual realized returns for the AR(1) model from 30.07.1999 to 30.09.2017, with bars indicating +/- 1 standard deviation. In most cases, actual realized returns are within one standard deviation of predicted returns, however realized returns again show more variation than predicted returns.

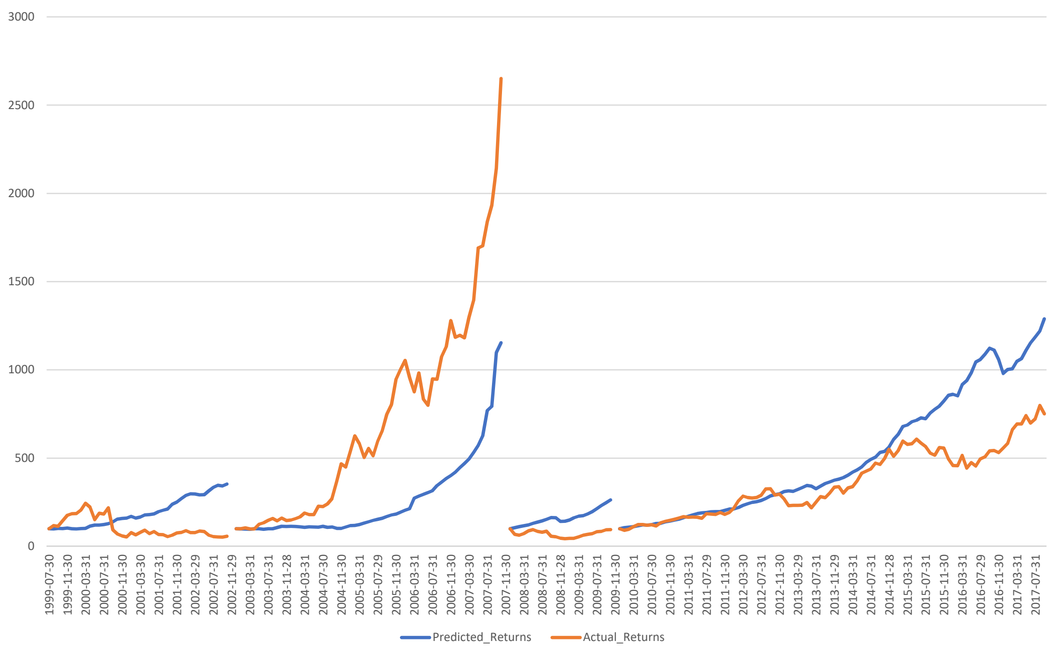


Figure 5.11: AR(1) - performance breakdown by sub-period. This graph compares the relative performance of a real and a notional portfolio, breaking down the investment period by resetting the value of the portfolio to 100 USD on 31.12.2002, 31.12.2007, and 31.12.2009 in order to explore the relative performance during these periods. This breakdown would suggest that there are in fact no periods during which the AR(1) model based on the factors used in this section would provide predicted returns which would be useful in a Black-Litterman model.

Data	Real Data	Prediction
Mean	0.030	0.033
Standard Distribution	0.122	0.046
Spearman Correlation		-0.076

Table 5.9: This table presents the first two moments of the distributions of real and predicted returns of the GARCH(1,1) model over the same period. While both distributions have similar means, the standard deviation of real returns is over 12%, while the standard deviation of predicted returns is only 4.6%.

1,000 samples per chain were generated for tuning and discarded.

The traceplot for the first prediction is presented in Figure 5.12. The left hand side of the traceplot shows that all chains converge to the same distributions for each parameter. The right hand side of the traceplot shows that the parameter spaces have been explored evenly.

Figure 5.13 compares all predicted returns to actual realized returns. In most cases, actual realized returns are outside of one standard deviation of predicted returns. GARCH(1,1) underestimates the uncertainty in the prediction.

Table 5.9 presents the first two moments of the distributions of real and predicted returns over the same period. While both distributions have similar means, the standard deviation of real returns is around 12%, while the standard deviation of predicted returns is much lower, at 4.6%.

The result of the Kolmogorov-Smirnov test yields a p-value of the order of 10^{-7}, resulting in the rejection of the null hypothesis, at 5% significance level, that the two samples do in fact come from the same distribution.

Figure 5.14 compares the relative performance of a real and a notional portfolio, breaking down the investment period by setting the value of the portfolio to 100 USD on 31.12.2002, 31.12.2007, and 31.12.2009 in order to explore the relative performance during these periods. This breakdown does not clearly provide evidence that the prediction returns obtained using the GARCH(1,1) model are not useful in predicting real returns, as the performance of the real and the notional portfolio bear a certain resemblance. Nevertheless, the correlation coefficient between the two series is very low, and in fact negative, at -0.076.

5.6.4 EGARCH(1,1)

The model specified in Section 5.5.6 was implemented and trained on monthly data from 30.11.1994 to 30.09.2017. Predictions were performed for each month from 30.07.1999 to 30.09.2017. Each prediction was performed using a model which was trained on the latest 54 months. 4 chains of 6,000 samples each were generated, for a total of 24,000 samples. 1,000 samples per chain were generated for tuning and discarded. Training one stock for

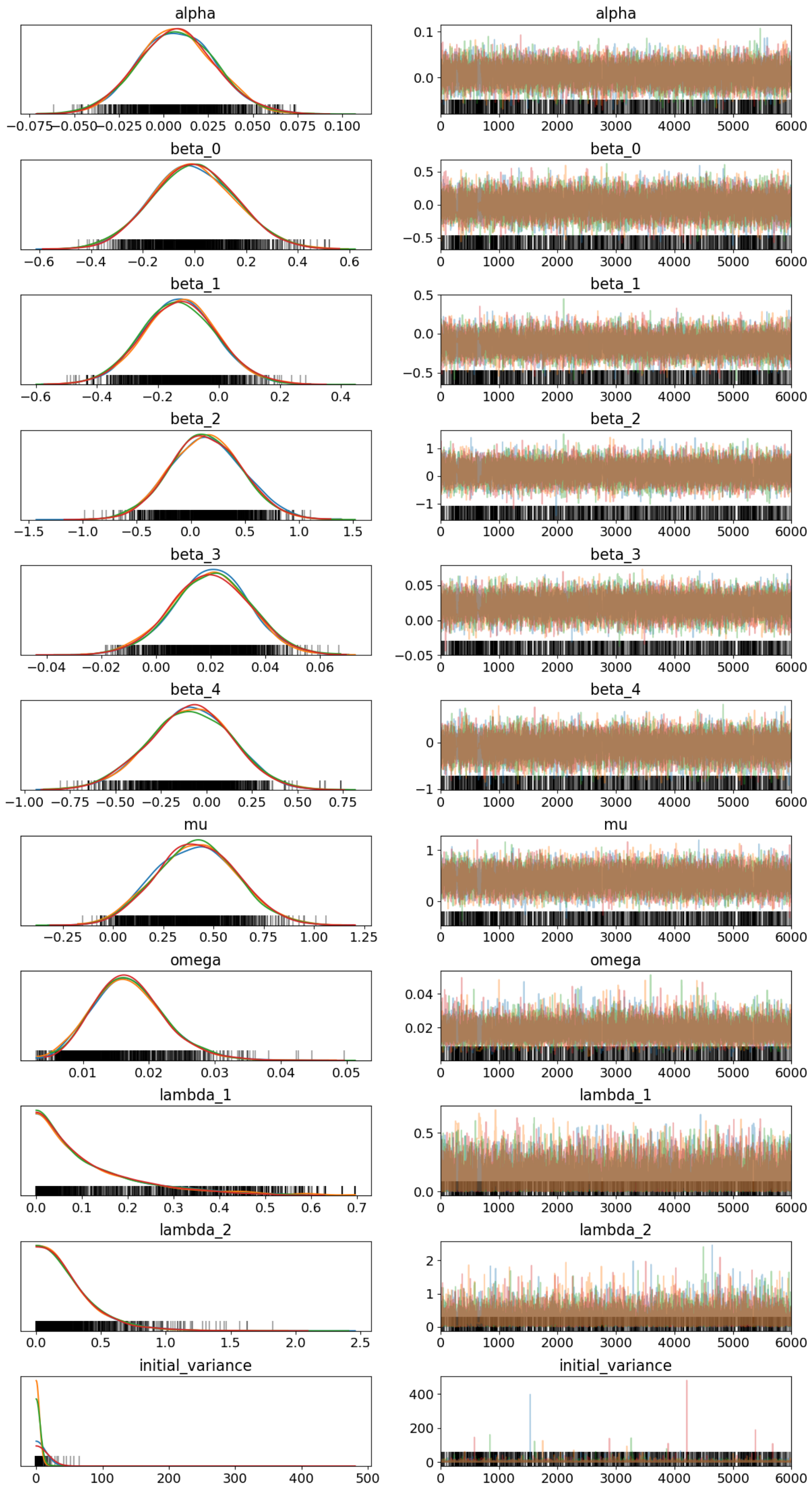


Figure 5.12: The traceplot related to the training of the GARCH(1,1) model for the first data point, on 30.07.1999. The left hand side of the figure shows that all chains converged to the same distribution for the parameter values. The right hand side of the traceplot shows the MCMC model has explored the parameter space evenly.

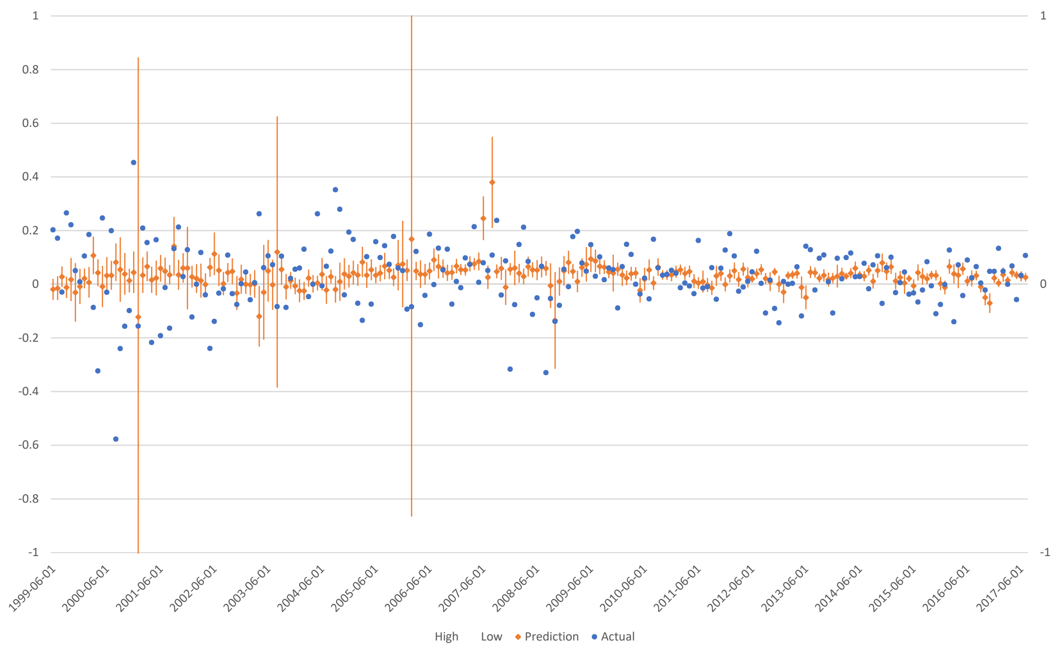


Figure 5.13: This chart compares all predicted returns to actual realized returns for the GARCH(1,1) model from 30.07.1999 to 30.09.2017, with bars indicating +/- 1 standard deviation. In most cases, actual realized returns are outside of one standard deviation of predicted returns. GARCH(1,1) underestimates the uncertainty in the prediction.

one prediction took approximately 20 minutes on average.

Data	Real Data	Prediction
Mean	0.030	0.033
Standard Distribution	0.122	0.041
Spearman Correlation		-0.026

Table 5.10: This table presents the first two moments of the distributions of real and predicted returns of the EGARCH(1,1) model over the same period. While both distributions have similar means, the standard deviation of real returns is over 12%, while the standard deviation of predicted returns is only 4.1%.

The traceplot for the first prediction is presented in Figure 5.15. The left hand side of the traceplot shows that some chains did not converge to the same distributions for each parameter. In addition, the right hand side of the traceplot shows that the parameter spaces have not been explored evenly. The use of a larger number of samples to tune the model was explored. Results are presented in Appendix E resolved these issues but required a very long time and resulted in similar predictions. For these reasons, results based on more limited tuning were retained.

Figure 5.16 compares all predicted returns to actual realized returns. Most realized returns lie outside of one standard deviation from predicted returns. EGARCH(1,1) underestimates the uncertainty of predictions.

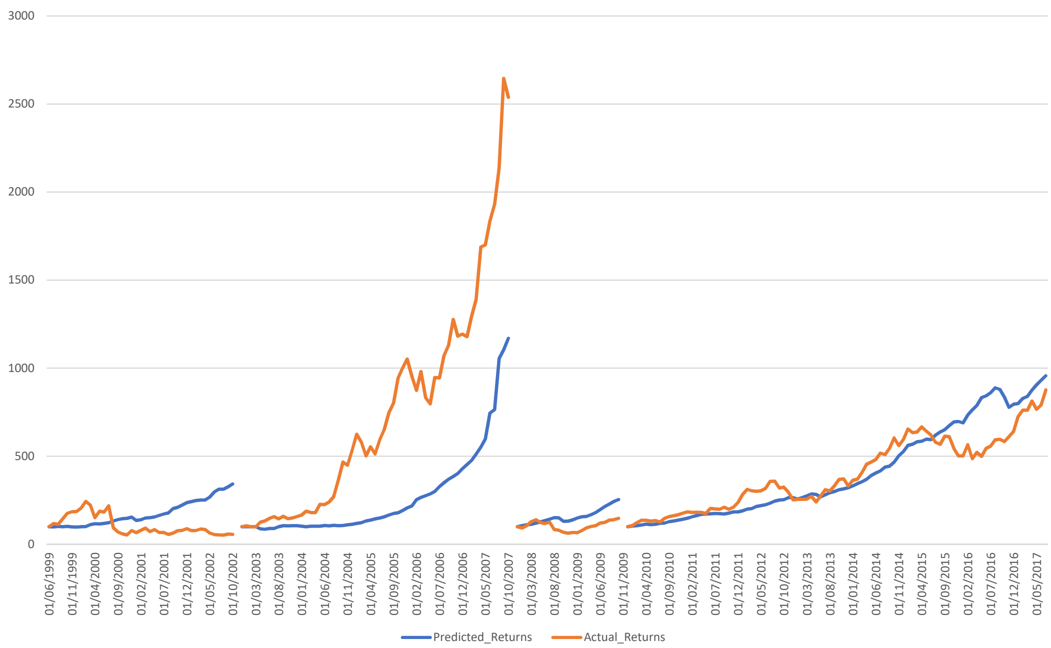


Figure 5.14: GARCH(1,1) - performance breakdown by sub-period. This graph compares the relative performance of a real and a notional portfolio, breaking down the investment period by setting the value of the portfolio to 100 USD on 31.12.2002, 31.12.2007, and 31.12.2009 in order to explore the relative performance during these periods. This breakdown does not clearly provide evidence that the prediction returns obtained using the GARCH(1,1) model are not useful in predicting real returns, as the performance of the real and the notional portfolio bear a certain resemblance. Nevertheless, the correlation coefficient between the two series is very low, and in fact negative, at -0.076.

Table 5.10 presents the first two moments of the distributions of real and predicted returns over the same period. While both distributions have similar means, the standard deviation of real returns is around 12%, while the standard deviation of predicted returns is much lower, at 4.1%.

The result of the Kolmogorov-Smirnov test yields a p-value of the order of 10^{-9}, resulting in the rejection of the null hypothesis, at 5% significance level, that the two samples do in fact come from the same distribution.

Figure 5.17 compares the relative performance of a real and a notional portfolio, breaking down the investment period by setting the value of the portfolio to 100 USD on 31.12.2002, 31.12.2007, and 31.12.2009 in order to explore the relative performance during these periods. This breakdown would suggest that there are in fact no periods during which the EGARCH(1,1) model based on the factors used in this section would provide predicted returns which would be useful in a Black-Litterman model. In fact the correlation coefficient of the two series is very small and negative, -0.026.

5.6.5 EGARCH-M(1,1)

The model specified in Section 5.5.7 was implemented and trained on monthly data from 30.11.1994 to 30.09.2017. Predictions were performed for each month from 30.07.1999 to 30.09.2017. Each prediction was performed using a model which was trained on the latest 54 months. 4 chains of 6,000 samples each were generated, for a total of 24,000 samples. 3,000 samples per chain were generated for tuning and discarded. Training was slow as there were frequent errors as described in Section 5.5.1. In these cases, training was launched once again until it completed without error. One single month for one stock took up to approximately 75 minutes on the computer used for testing.

The traceplot for the first prediction is presented in Figure 5.18. The left hand side of the traceplot shows that, even with the relatively limited samples size used to tune the model, all chains converge to the same distributions for each parameter. The right hand side of the traceplot shows that the parameter spaces have been explored reasonably evenly. Although, as discussed in Appendix E, increasing the number of samples used for tuning would improve the traceplots, the increase in training time required to do so would be prohibitive. As the predictions using fewer samples for tuning appear to be similar to those obtained with a larger number of samples, this approach was retained.

Figure 5.19 compares all predicted returns to actual realized returns. Many realized returns lie within one standard deviation from predicted returns. EGARCHM(1,1) better estimates the uncertainty of predictions than previous models. The bar chart does, however, identify four severe outliers: a negative return of 65% is predicted on the 31.05.2013, a negative return of 70% is predicted on the 31.01.2014, -59% on 31.05.2016, and -62% on 30.06.2016.

Data	Real Data	Prediction
Mean	0.030	0.02
Standard Distribution	0.122	0.112
Spearman Correlation		-.062

Table 5.11: This table presents the first two moments of the distributions of real and predicted returns of the EGARCH-M(1,1) model over the same period. Both distributions have similar means and standard deviations. Nevertheless the result of the Kolmogorov-Smirnov test yields a p-value of the order of 10^{-5}, resulting in the rejection of the null hypothesis, at 5% significance level, that the two samples do in fact come from the same distribution.

The size of these negative returns is not unrealistic, as there actually was a month with negative returns of 59% in early 2000. However the frequency with which these negative returns are predicted is unrealistic, in addition to being incorrect. In practice, a review of outlier predictions would be required.

Table 5.11 presents the first two moments of the distributions of real and predicted returns over the same period. Both distributions have similar means and standard deviations.

Nevertheless the result of the Kolmogorov-Smirnov test yields a p-value of the order of 10^{-5}, resulting in the rejection of the null hypothesis, at 5% significance level, that the two samples do in fact come from the same distribution.

Figure 5.20 compares the relative performance of a real and a notional portfolio, breaking down the investment period by setting the value of the portfolio to 100 USD on 31.12.2002, 31.12.2007, and 31.12.2009 in order to explore the relative performance during these periods. The graph in the last period is strongly affected by the outlier predictions discussed above. This breakdown would suggest that there are in fact no periods during which the EGARCH-M(1,1) model based on the factors used in this section would provide predicted returns which would be useful in a Black-Litterman model. In fact the correlation coefficient of the two series is small and negative, -0.062.

5.7 Summary

The main objective of this experiment was to build and explore different models for a Bayesian network to generate views for a Black-Litterman asset allocation model, and to develop tools to understand and evaluate the predictions.

Different Bayesian networks were built using AR(1), GARCH(1,1), EGARCH(1,1), and EGARCHM(1,1) models.

The models were tested against synthetic data created in Excel. In all cases, it was possible to recover the known values of model parameters in using the Bayesian networks, leading to the conclusion that the models were correctly implemented.

The models were then used to predict the returns on AAPL stocks. All Bayesian networks were constructed using the same factors. Results did not clearly indicate that any model would be useful in generating views for a Black-Litterman model.

In no case was it possible to provide evidence that predicted returns and realized returns were similarly distributed. Correlation coefficients between the predicted and realized returns were low and even negative.

In most cases, the models failed to correctly estimate the uncertainty in the prediction. This is important as the uncertainty of the prediction is a direct input into the Black-Litterman model. Exceptions were the linear regression, AR(1), and EGARCHM(1,1) models.

5.8 Discussion

In this experiment, Bayesian networks were implemented using a number of different functional forms for the relationship between factors and returns. Predictions of returns using these models may be improved through a better selection of factors and an increase in the number of samples generated to tune the model. This must be performed in such a way that inference remains possible in a reasonable time period.

Increasing the number of factors used could result in better predictions. The likelihood of finding factors on which the returns of a given stock are conditionally dependent increases when a broader range of factors are considered. The cost is an increase in training time.

Increasing the number of samples used to tune the model has been shown to result in a better search through the parameter space for an EGARCH model. This may result in better predictions. The cost is an additional increase in training time.

Training for one data point made use of four cores on the computer and took between 15 and 75 minutes. Training one stock for 218 dates took up to 12 days on a machine with 4 cores. Applying this same model to 100 the equities in the NASDAQ would be beyond the scope of this thesis. Increasing either the number of factors used or the number of samples for tuning would further increase training times. There are several possible solutions which, used together, may reduce this issue.

First, the number of equities considered for inclusion in the portfolio could be reduced to a subset of the 100 equities in the NASDAQ. One approach could be to include as many equities as possible, subject to time constraints. It should be noted that regulatory or other constraints, not considered further in this research, may dictate a lower limit on the number of equities in the portfolio to ensure sufficient diversification.

Second, training could be performed on a high performance cluster. This research will make use of the UCL Cluster.

Third, it may be possible to select better priors. One problem was that training produced errors which required the initiation of a new training sessions. One hypothesis is that training sometimes explored a parameter space resulting in over/underflows which were avoided in other training sessions. Selecting better priors which avoid this parameter space may be possible.

Fourth, the size of the Bayesian network could be reduced through dimensionality reduction applied to the factors. Algorithms exist which could help reduce a broader range of factors to a smaller number of relevant features. This is further considered in Section 6.3.1.3.

While the main objectives of this experiment were met, further research is required in order to build Bayesian networks which better predict returns as well as estimates of the uncertainty of the predictions.

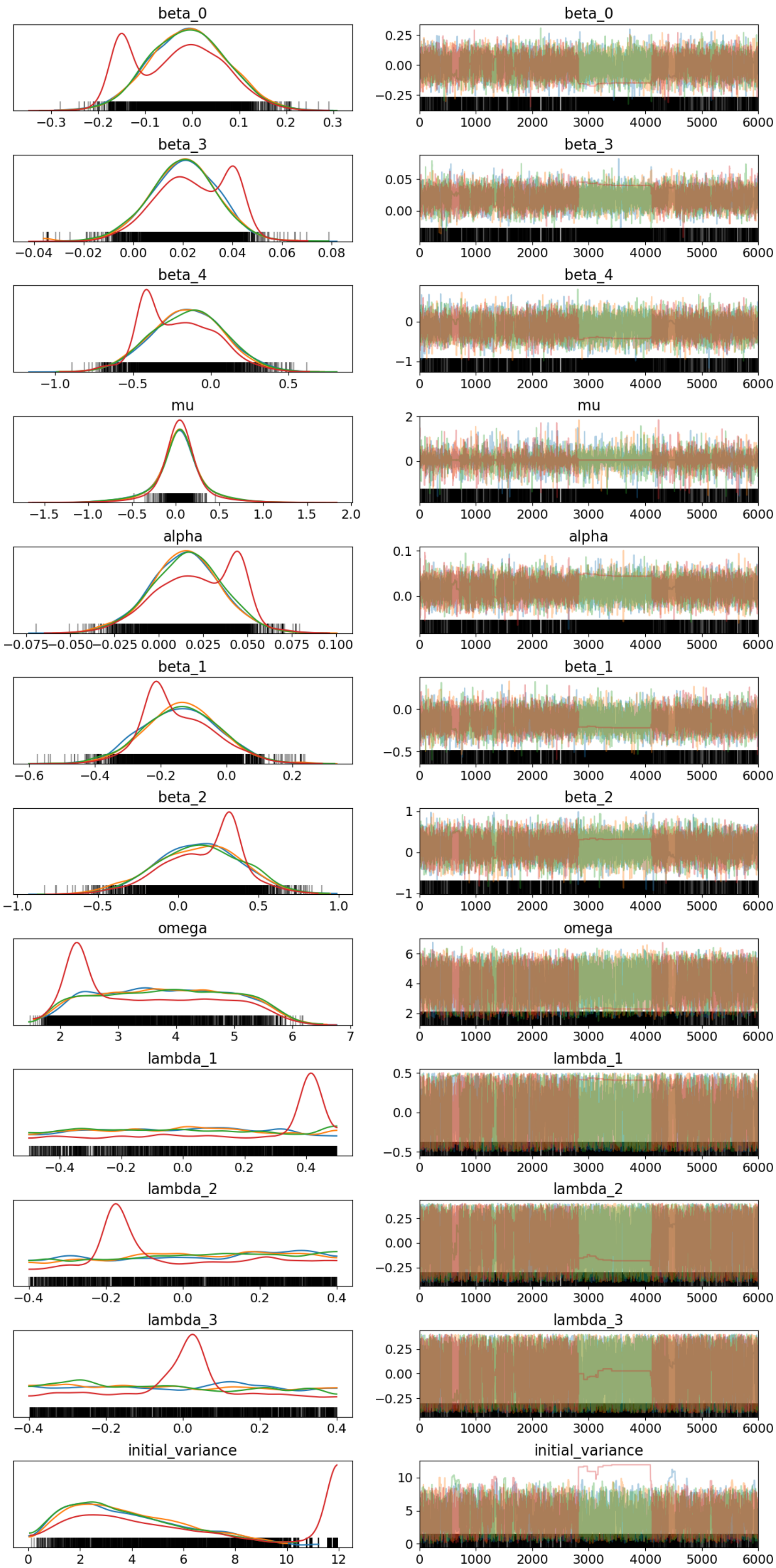


Figure 5.15: The traceplot related to the training of the EGARCH(1,1) model for the first data point, on 30.07.1999. The left hand side of the traceplot shows that some chains did not converge to the same distributions for each parameter. The right hand side of the traceplot shows that the parameter spaces have not been explored evenly. The use of a larger number of samples to tune the model resolved these issues but required a very long time and resulted in similar predictions. Results based on more limited tuning were therefore retained.

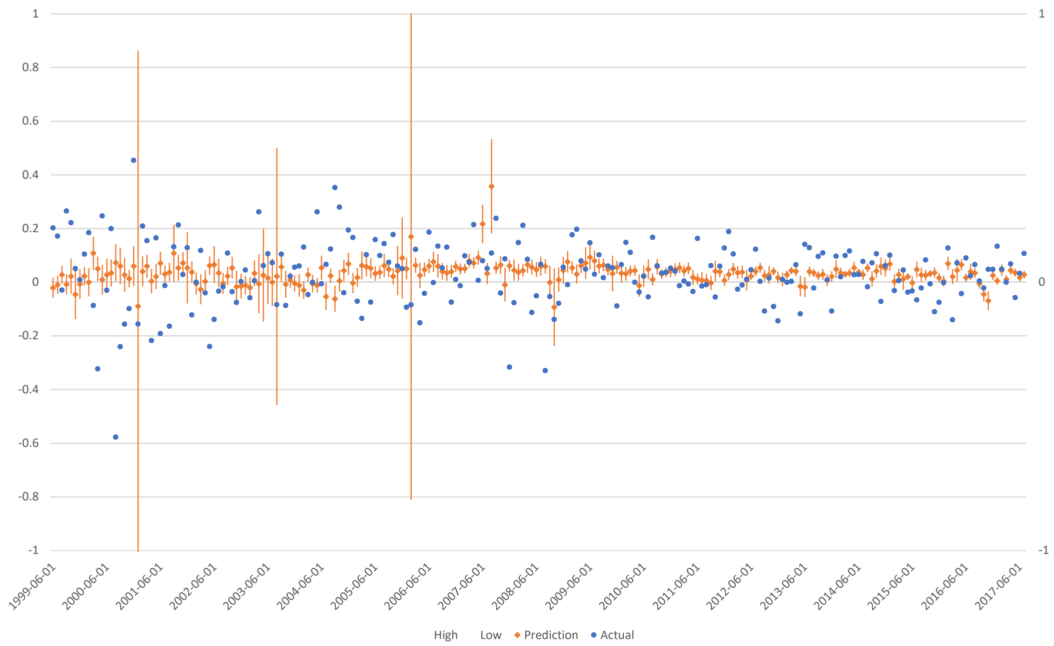


Figure 5.16: This chart compares all predicted returns to actual realized returns for the EGARCH(1,1) model from 30.07.1999 to 30.09.2017, with bars indicating +/- 1 standard deviation. In most cases, actual realized returns are outside of one standard deviation of predicted returns. EGARCH(1,1) underestimates the uncertainty in the prediction.

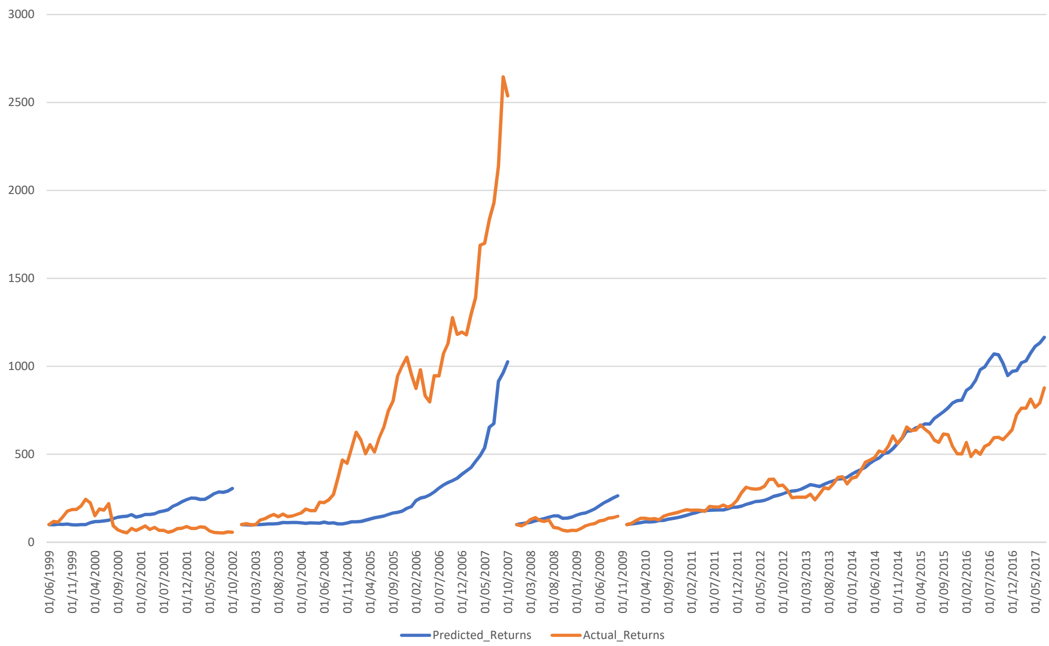


Figure 5.17: EGARCH(1,1) - performance breakdown by sub-period. This graph compares the relative performance of a real and a notional portfolio, breaking down the investment period by setting the value of the portfolio to 100 USD on 31.12.2002, 31.12.2007, and 31.12.2009 in order to explore the relative performance during these periods. This breakdown would suggest that there are in fact no periods during which the EGARCH(1,1) model based on the factors used in this section would provide predicted returns which would be useful in a Black-Litterman model. In fact the correlation coefficient of the two series is very small and negative, -0.026.

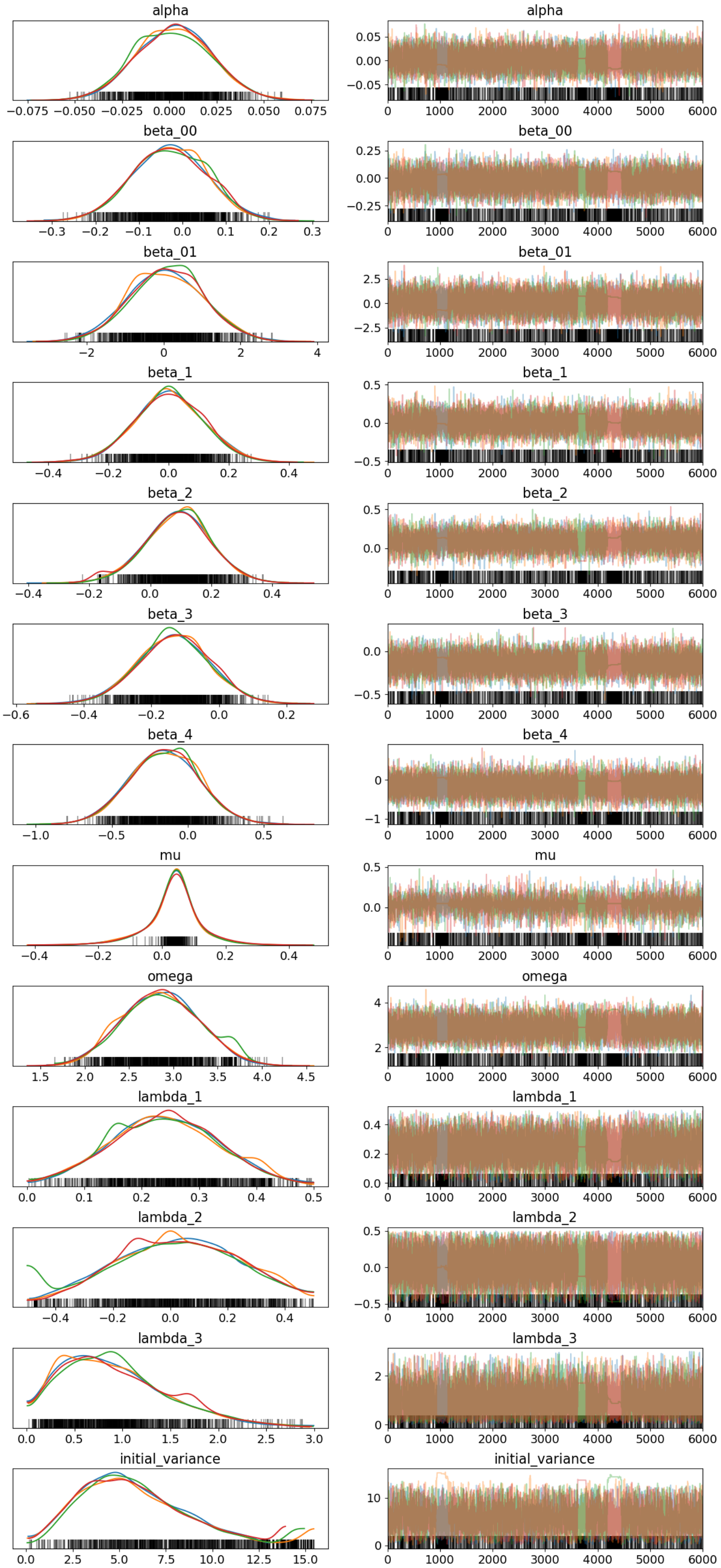


Figure 5.18: The traceplot related to the training of the EGARCH-M(1,1) model for the first data point, on 30.07.1999. The left hand side of the traceplot shows that, even with the relatively limited samples size used to tune the model, all chains converge to the same distributions for each parameter. The right hand side of the traceplot shows that the parameter spaces have been explored reasonably evenly.

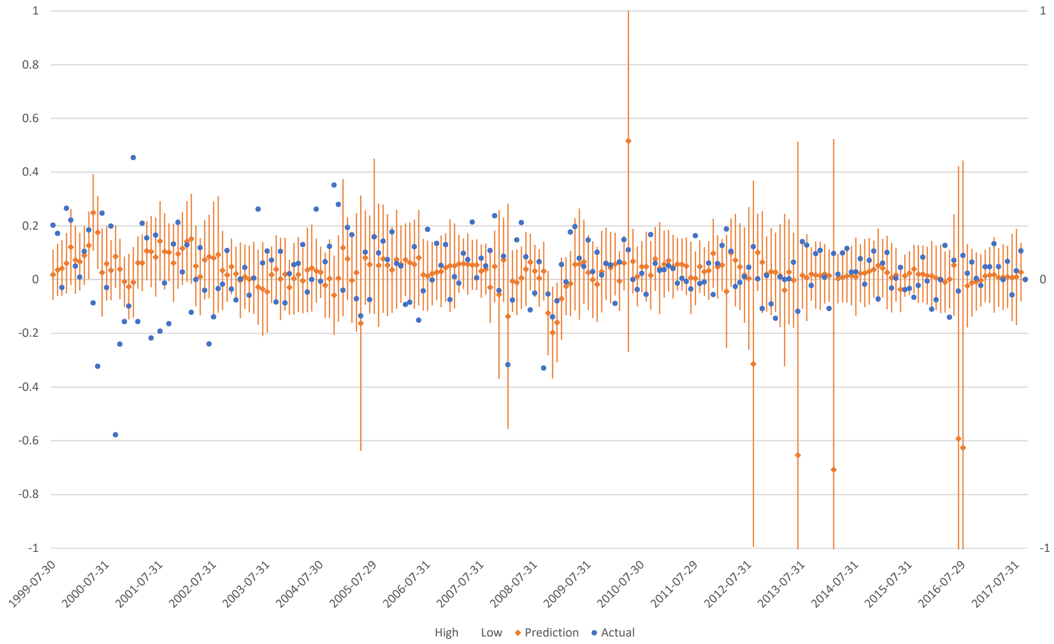


Figure 5.19: This chart compares all predicted returns to actual realized returns for the EGARCH-M(1,1) model from 30.07.1999 to 30.09.2017, with bars indicating +/- 1 standard deviation. Many realized returns lie within one standard deviation from predicted returns. EGARCHM(1,1) better estimates the uncertainty of predictions than previous models. The bar chart does, however, identify four severe outliers: a negative return of 65% is predicted on the 31.05.2013, a negative return of 70% is predicted on the 31.01.2014, -59% on 31.05.2016, and -62% on 30.06.2016.

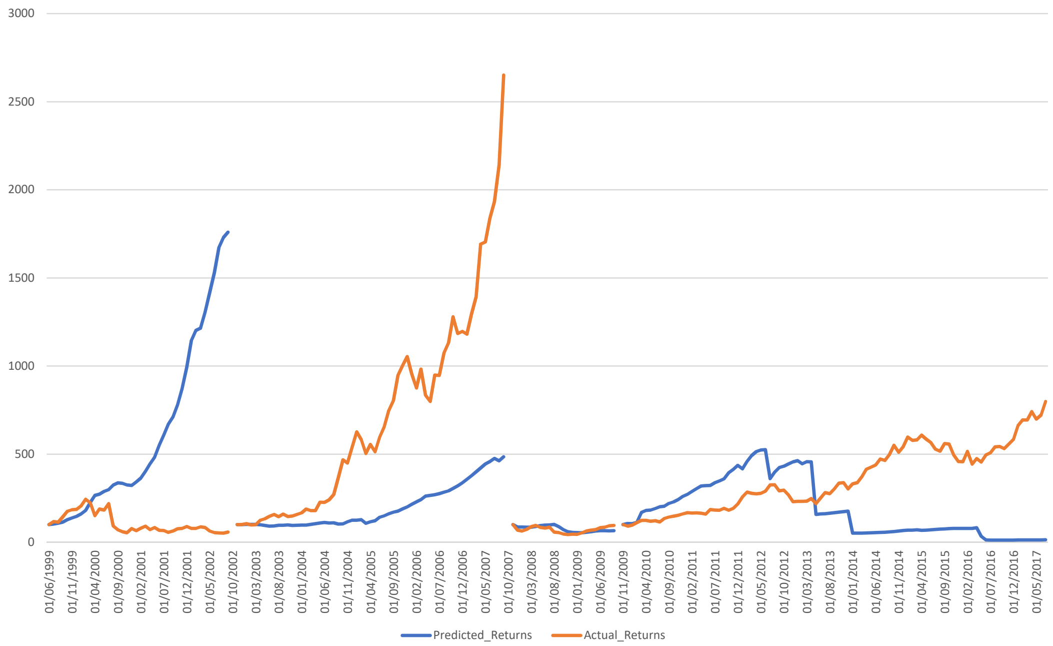


Figure 5.20: EGARCH-M(1,1) - performance breakdown by sub-period. This graph compares the relative performance of a real and a notional portfolio, breaking down the investment period by setting the value of the portfolio to 100 USD on 31.12.2002, 31.12.2007, and 31.12.2009 in order to explore the relative performance during these periods. The graph in the last period is strongly affected by the outlier predictions discussed above. This breakdown would suggest that there are in fact no periods during which the EGARCH-M(1,1) model based on the factors used in this section would provide predicted returns which would be useful in a Black-Litterman model. In fact the correlation coefficient of the two series is small and negative, -0.062.

Chapter 6

Algorithmic Asset Management System

This experiment compares the performance of portfolios generated based on views produced by different algorithms to the performance of a benchmark. The views are generated using six algorithms: standard GARCH, standard EGARCH, standard EGARCHM, Bayesian GARCH, Bayesian EGARCH, and Bayesian EGARCHM. The views are used in a standard Black-Litterman portfolio optimization model developed in Experiment 1. The benchmark is a neutral portfolio constructed without views. A portfolio with equal weights is also constructed. This experiment finalizes the scientific analysis platform begun in the first experiment.

6.1 Introduction

This experiment aims to determine whether views generated using a Bayesian network can be useful in portfolio construction using a final version of the scientific analysis platform used throughout this thesis. It will compare the performance of portfolios created using a Black-Litterman model with views generated using a Bayesian networks based on various models representing the relationship between factors and equity returns to the returns on a neutral portfolio. A portfolio with equal weights will also be constructed.

This experiment will consider portfolios invested over the three year period from 31 July 2016 to 1 August 2019 and rebalanced on a daily basis. Daily returns from earlier periods will be considered in training as required.

6.2 Aims and Objectives

The ultimate aim of this experiment is to backtest a Black-Litterman model using views generated from a Bayesian network using only data available at time of trade, and to evaluate the performance of the resulting portfolios using a final version of the scientific analysis platform developed throughout this thesis.

The first objective is to enhance and finalize the scientific analysis platform.

The second objective is to implement and backtest a Black-Litterman model to develop several portfolios, each based on a different Bayesian network using only information available at time of trade.

The third objective is to evaluate the performance of these portfolios within the platform.

6.3 Background

6.3.1 Dimensionality Reduction

The QQQ index is made up of around 100 equities[1]. Hundreds of different factors can be used to forecast equity returns. Creating a number of models which predict the returns of 100 equities using all potentially available factors would be intractable. Dimensionality reduction is the process of reducing the set of factors used in prediction. It will be used to inform on the core set of equities which can be used to model the QQQ index. It will also be used to select which factors are "best".

Generally, dimensionality reduction can be performed by selecting a subset of all factors, or by extracting a (smaller) set of relevant features from all factors .

6.3.1.1 Factor Selection

Selecting 10 equities to represent the NASDAQ-100 is not considered a critical process. Listing all possible sets of 10 equities and selecting the set which has, as a market-weighted portfolio, the highest correlation with the Index would in principle work, but would necessitate the calculation of correlations on around 10^{19} sets, which is intractable. A factor selection approach will be used instead.

Three approaches to factor selection include wrapper methods, filter methods, and embedded methods (Chandrashekar and Sahin, 2013). A filter method ranks available factors using an appropriate criterion.

Possible approaches to ranking the 100 equities include ordering by market capitalization or correlation with the Index. This experiment will rank each equity based upon the correlation of equity returns with the returns of the QQQ ETF, over the period being considered and retain 10 equities. This relatively simple approach is selected for several reasons. It is expected to give a reasonable proxy for the ETF, it is also simple, objective, and easily implemented.

6.3.1.2 Feature Extraction

Selecting a subset of factors is expected to be important to this experiment.

[1]The actual number of equities varies. As at 15.08.2019, there are 103 equities in the Index `https://en.wikipedia.org/wiki/NASDAQ-100`.

One approach could be to consider all possible subsets of factors and train each model for each equity on each set in order to select the subset which provides the best prediction, however this is intractable.

Feature extraction consists in transforming a large number of available factors into a smaller set of uncorrelated features which retain most of the information present in the initial factors. Many approaches can be considered (Ding et al., 2012). Principal Component Analysis (PCA) will be used in this experiment (Jolliffe, 2002).

6.3.1.3 Principal Component Analysis

PCA is an unsupervised learning algorithm to transform the space of factors into a feature space of fewer dimensions which retains as much of the initial variance as desired. It does not take into consideration the model used, and so will not select those factors most significant for a given model. It can in this sense be considered model neutral, which may be desirable at this stage in order to compare the performance of models.

For a set of n factors x, the algorithm looks for a succession of m uncorrelated linear functions $a_i' x$ having maximum variance, where $0 < i \leq m$

$$a_i' x = \sum_{j=1}^{n} a_{ij} x_j. \tag{6.1}$$

The linear functions $a_i' x$ are the principal components. The ideal outcome is that most of the variance of x will be explained by $m \ll n$ principal components.

The a's are found by looking at the covariance matrix Σ of the dataset. The i^{th} principal component is given by $z_i = a_i' x$ where a_i is an eigenvector of Σ corresponding to its i^{th} largest eigenvalue.

6.3.2 Performance Measurement

In comparing the performance of different portfolios, both the returns of the portfolios and their risk must be taken into consideration. For example, given portfolios with similar returns, investors would prefer the one with lower risk.

6.3.2.1 Sharpe Ratio

Sharpe (Sharpe, 1966, 1994) proposed the following Ex Post ratio S in order to compare a portfolio to a benchmark

$$S = \mu / \sigma \tag{6.2}$$

$$\mu = \frac{1}{N} \sum_{n=1}^{N} \mu_n \tag{6.3}$$

$$\mu_n = R_n - B_n, \tag{6.4}$$

where R_n and B_n are the realized and benchmark returns respectively, and σ is the standard deviation $\sigma^2 = \sum_n (\mu_n - \mu)^2/(N-1)$ over the period.

The best estimator of S, $\hat{S}$, is also of this form (Lo, 2002).

The Sharpe ratio is generally presented based on annual returns. If it is calculated based on daily IID returns, one approach would be to simply multiply it by $\sqrt{252}$, as there are 252 trading days in a year and returns, the numerator in Equation (6.3.2.1), increase linearly with t whereas σ, the denominator, increases as the $\sqrt{t}$. Lo (2002) argues that, given the fact that market returns are not IID, this approach is misleading. For this reason, the Sharpe Ratio will generally not be annualized. Results will be useful for this research but not comparable to data generally available on other funds or investments.

The Standard Error (Lo, 2002) in the estimation of the Sharpe Ratio will be estimated as

$$\text{SE} = \sqrt{\frac{1 + \frac{\hat{S}^2}{2}}{T}}. \tag{6.5}$$

The 95% confidence interval around $\hat{S}$ is given by

$$\hat{S} \pm 1.96\sqrt{\frac{1 + \frac{\hat{S}^2}{2}}{T}}. \tag{6.6}$$

6.3.2.2 Modigliani Risk-Adjusted Performance

Modigliani risk-adjusted performance (Modigliani and Leah, 1997), $M2$, is similar to the Sharpe Ratio, with the advantage of being more intuitive as it is expressed in terms of % returns. It is defined as

$$\text{M2} = S\sigma_B, \tag{6.7}$$

where S is calculated using the excess returns over R_f, the average risk-free rate for the period under consideration, and σ_B is the standard deviation of the benchmark portfolio.

6.3.2.3 Herfindahl-Hirschman Index

The Herfindahl-Hirschman Index (Rhoades, 1993) (HHI) is a measure of concentration. Typically defined based on market concentration of firms, it is used in this thesis to explore the extent to which portfolios are giving higher weights to a smaller subset of equities. For a portfolio of N equities, it is defined as

$$H = \sum_{n} \omega_n^2. \tag{6.8}$$

The HHI will range from $1/N$, if each equity has an equal weight, to 1 if only one equity is given a weight of 100%.

The Participation Ratio, defined as $1/H$, gives an intuition of the number of equities in which the portfolio is concentrated, and can range from 1 to N.

6.4 Design

In order to meet the aims and objectives defined in Section 6.2, this experiment will be designed as follows:

First, select a set of 10 equities as a proxy to the around 100 equities in the NASDAQ-100 Index using factor selection as described in Section 6.3.1.1.

Second, define and justify a relatively broad range of potential factors, and extract the most interesting features of this set as described in Section 6.3.1.2. Data relating to the factors must be available at time of trade.

Third, generate views based on:

1. a traditional statistical GARCH model (S-GARCH)
2. a traditional statistical EGARCH model (S-EGARCH)
3. a traditional statistical EGARCHM model (S-EGARCHM)
4. a Bayesian network using a GARCH model (B-GARCH)
5. a Bayesian network using an EGARCH model (B-EGARCH)
6. a Bayesian network using an EGARCHM model (B-EGARCHM)

The result is six time series of {expected return, variance of expected return} pairs.

Fourth, implement and backtest an equal-weighted portfolio, a Black-Litterman portfolio with no views, and six Black-Litterman portfolios based on the views generated previously. The result will be eight time series of portfolio returns.

Fifth, compare the portfolio returns. Include a comparison to the returns of the QQQ Index to provide an intuition that the overall returns are reasonable.

6.5 Implementation

This experiment implements the platform first presented in Section 3.3. Each component of the platform can be developed and modified separately. A detailed view of the scientific platform implemented in this experiment is presented in Figure 6.1. Each component is

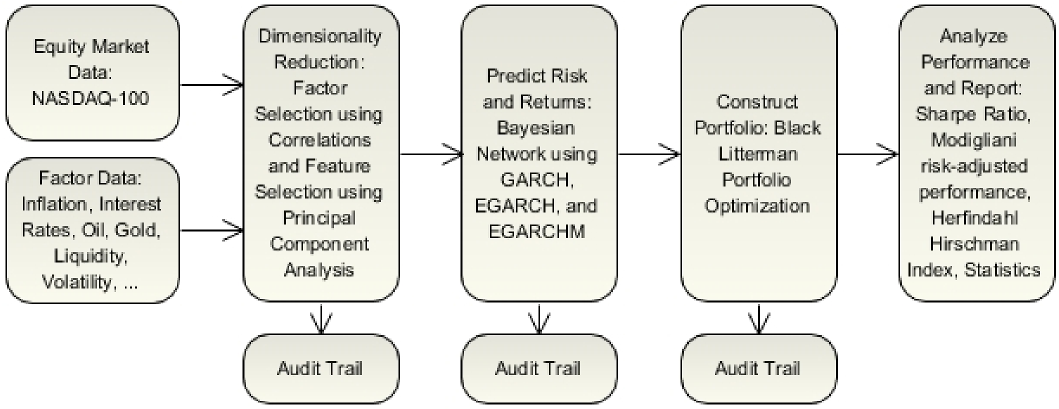


Figure 6.1: This is a detailed view of the platform, first presented in Section 3.3, which is implemented in this experiment. Each component of the platform can be developed and modified separately. Details are discussed in Section 6.5.

discussed in the following sections. All analysis is based on audit trails and reports which are automatically produced by the platform.

6.5.1 Selecting Subset of Equities

As discussed in Section 6.3.1, a small number of equities must be selected from those in the NASDAQ-100 Index, in order to keep the computation times reasonable. Equities will be ranked based upon the correlation of their daily returns to the daily returns of the QQQ ETF.

The objective is not to pre-select equities which outperform the Index, but to select a set of equities which is representative. The process may however introduce bias, for example if it systematically omits those securities with low correlations to the returns of the ETF due to the fact that they performed very poorly in the period under review while the ETF performed very well overall.

The correlation between the daily returns on the QQQ ETF and each equity in the Index was calculated once: over the period from 31/07/2015 to 31/07/2016. The 10 equities with the correlations above 0.7260 were selected. They are listed in Table 6.1. HSIC had the lowest correlation with the index over the period, 0.7260, and MSFT the highest, 0.8181. AMZN had the highest mean daily returns, 0.0017 and AAPL the lowest, with -0.0003. Portfolio construction and back testing was performed over the period from 31/07/2016 to 01/08/2019.

6.5.2 Selecting Features from Factors

6.5.2.1 Selected Factors

A large number of factors have been used in research on equity returns (Beach and Orlov, 2007; Harvey et al., 2014). These factors include those based on quantitative data such as oil prices, and those based on qualitative data such as news or other unstructured data.

Ticker	Mean Returns	Standard Deviation	Correlation to QQQ
QQQ	0.0003	0.0131	1.0000
AAPL	-0.0003	0.0189	0.7473
ADP	0.0006	0.0134	0.7640
AMGN	0.0002	0.0184	0.7789
AMZN	0.0017	0.0218	0.7424
FISV	0.0010	0.0128	0.8079
GOOG	0.0010	0.0178	0.7679
GOOGL	0.0009	0.0173	0.7919
HSIC	0.0009	0.0125	0.7260
MSFT	0.0010	0.0169	0.8181
SBUX	0.0002	0.0160	0.7722

Table 6.1: Portfolios were constructed with the ten equities with the highest correlations with the NASDAQ-100 Index over the period from 31/07/2015 to 31/07/2016. Portfolio construction and back-testing was performed over the period from 31/07/2016 to 01/08/2019.

Although news data could be useful in predicting equity returns, defining metrics based on news is a complex subject in its own right. This thesis will focus on quantitative data for practical reasons.

The time required for training Bayesian networks depends strongly upon the number of factors used. In order to reduce computation time to a reasonable time period, given the fact that training will be performed for 10 equities and 252 dates, a subset of the factors that have been used in research is selected, and a small set of features is extracted from these factors.Preliminary testing indicates that feature extraction using PCA is rapid - around 1 second per data point - and can be recalculated at each point prior to predictions.

The set of features may be different for each equity. The set of factors which are significant could be expected to change over time, as different factors become more relevant than others, and so the set of features to be extracted would also vary with time. For example, interest rates may be important at some point, and news on trade more significant at other times.

In many cases, daily data will not be available for a factor. One approach in dealing with missing data could be to recalculate factors using, for example, interpolation. This approach is rejected as it could not be used in actual trading. Another approach could be to predict the missing factors using historical data. In this thesis, factors are only considered if daily data can be obtained for that factor. In some cases, a proxy such as an ETF for the factor may be included. Preference will be given to factors for which data is available from the markets, as this is transparent and objective data. The list of selected factors is presented in Table 6.2.

Beach and Orlov (2007) used EGARCHM(1,1) to create views for Black-Litterman models using the growth in industrial production for industrial countries, inflation, the return on the US dollar index relative to major currencies, the difference in the yield on BAA and AAA bond indexes from Moody's, the difference in the three-month Eurodollar yield and the three-month treasury bill yield, the difference in the 10-year treasury bond yield and the three-month treasury bond yield, and the percentage change in the world spot price of oil.

Data for industrial production is not available on a daily basis. No proxy was found.

Data for inflation is not available on a daily basis. An ETF, RINF, which tracks the spread between US Treasury inflation-protected securities and US Treasurys of equal maturity, can in principle be used as a proxy for inflation. However, assets under management of the ETF are only around 50 MUSD, which leads to reduced liquidity and therefore potential mispricing relative to inflation. Fees are relatively high due to the fact that the strategy requires short positions in US Treasurys. In spite of these issues, the ETF will be included in the list of selected factors.

The return on the USD relative to other major currencies appears intuitively as an interesting factor. This could be modelled as the difference between interest rates USD Treasuries and the government bonds of other major currencies. As the feature extraction would model differences between factors if they are relevant, it is considered more useful to include both instruments as factors. As interest rates are inversely related to the prices of bonds, ETFs and indices tracking the prices of treasuries and international treasury bonds were included.

The world spot price of oil was considered using the BNO ETF. The price of Gold was also included.

Harvey et al. (2014) lists 316 different factors broken down into different risk types: financial, macro, microstructure, behavioural, accounting, and other. Volatility, liquidity, momentum, sentiment, short sale restrictions, media coverage, PE ratio, intangibles, and political campaign contributions are examples of factors which could be considered.

Market volatility can be included with the VIX Index. Volatility of the equity itself is available from market data suppliers as "Option Implied Volatility".

Many measures of liquidity exist, some simple such as the bid-ask spread, some more complex (Fong, Holden, and Trzcinka, 2017). A summary statistic for this measure is not readily available. Daily trading volume will be taken as a pragmatic and simple proxy for liquidity.

Momentum may be modelled by taking into consideration the six-month return on the equity.

Historical data for other data such as sentiment and financial ratios is not readily available at this time.

Identifier	Details	Proxy for
RINF	ProShares Inflation Expectations ETF	Inflation
USTTEN	ICE US 10 Year Treasury Futures Index	US 10 Year Treasury
IGOV	iShares International Treasury Bond	Return on major currencies
BNO	United States Brent Oil Fund	Oil ETF
IRX	CBOE Interest Rate Composite Index	13 Week Treasury Bill
TNX	CBOE Interest Rate 10 Year Note	10 Year Teasury Note
GLD	SPDR Gold Shares	Gold ETF
VIX	CBOE Volatility index	Market volatility
Volume	Trading volume × Market Price	Liquidity
OIV	Option Implied Volatility	Equity specific volatility
Momentum	Market Value Today / Market Value 6 months earlier - 1	Momentum

Table 6.2: These are the factors which were selected. A broad range of factors that have been used in research on equity returns in Harvey et al. (2014) and Beach and Orlov (2007) were considered. Selection was made by focusing on quantitative factors for which daily market data could be obtained either for the factor itself or for a proxy such as an ETF for the factor.

```
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA

#Following line is external code which returns a pandas DataFrame with the
    factors.
factors = get_factors_dataframe()
scaled_factors = StandardScaler().fit_transform(factors)
pca = PCA(0.95)
principal_components = pca.fit_transform(scaled_factors)
features = principal_components
```

Listing 6.1: PCA with scikit-learn

6.5.2.2 Feature Extraction

In order to extract a smaller number of relevant features from the factors listed in Table 6.2, a Principal Component Analysis is performed.

Using scikit-learn in python, the process is presented in Listing 6.1. The procedure requires that data be normalized with mean of 0 and variance of 1. This process in itself may speed up training of the Bayesian networks. The implementation requires that 95% of the variance in the original factors be explained by the Principal Components. These are returned in a Numpy array.

Preliminary testing indicates that between 6 and 7 features are extracted from the 11 factors in Table 6.2 when using 1 year of data.

6.5.3 Generate Views

Views, consisting of predicted returns and variance, are produced for every working day over the three year period from 31/07/2016 to 01/08/2019. Previous periods are used as required, for example for training or calculation of six-month momentum.

Traditional GARCH, EGARCH and EGARCHM models are created using the arch[2] python package.

For each prediction, PCA is run over the 3 months prior to the prediction. The resulting features are used as inputs to a Bayesian network. Training of the Bayesian network is over a one year period prior to prediction.

At times statistical EGARCH and EGARCHM did not converge. On those points, statistical GARCH was used for prediction.

In the statistical package, the volatility equations did not make use of features (external regressors). Only EGARCHM made use of features in predicting returns.

6.5.4 Generate Portfolios

The views generated in Section 6.5.3 were combined with equity market weights in a Black-Litterman model to generate portfolios for each day in the period from 31/07/2016 to 01/08/2019.

The Black-Litterman model described in Section 2.3.3 combines implicit market expected returns with predicted returns using a measure of uncertainty of the predicted returns. More importance is given to predicted returns if uncertainty is close to zero. More importance is given to market expected returns if uncertainty is high.

A very low level of uncertainty, in addition to reflecting a level of confidence in the predicted return which may be unrealistic, reduces the advantages of the Black-Litterman model over the simple Mean Variance model presented in Section 2.3.1. In particular, a very low level of uncertainty sometimes results in extremely large long and short positions being taken. It was found empirically that a minimum value of uncertainty of 0.1 was desirable. Uncertainty above 1.0 was found to have little impact. In these cases, market expected returns were given overwhelming importance. The uncertainty of the views derived in Section 6.5.3 was therefore cast to the interval [0.1, 1.0].

6.6 Results

6.6.1 Views

Correlations between the predicted and actual returns are presented in Table 6.3. None of the models predicted returns which had high correlations with actual returns for the equities considered. In some cases the correlations were negative. Nevertheless, the predictions may add value when being used to create portfolios, for example if they help identify those equities which are expected to over or under perform by, for example, correctly ranking

[2]Kevin Sheppard. (2018, October 3). bashtage/arch: Release 4.6.0 (Version 4.6.0)

securities in order of increasing performance or identifying outliers.

Details of the statistical properties of return predictions are presented in Table 6.4 and 6.5. Over the period, ADP and HSIC were the two securities with the highest actual mean return. With the exception of B-EGARCHM, which estimates a slightly higher return for HSIC, AAPL, and AMZN relative to most other securities, no other approach seems to rank the highest performers overall. On the other hand, B-EGARCHM has a mean prediction for FISV (0.0117) which is far above the mean of realised returns (0.0004). This does not support the view that the models rank securities in order of increasing returns in order to overweight those securities with better returns.

An interesting analysis may arise, however, when one combines predicted returns with the standard deviation of predictions. AAPL is the stock with the fourth highest mean realized return of 0.0011. All models rank it in their top 3. For example, S-EGARCHM ranks it highest, although its estimate is only 0.0003, and B-EGARCHM third highest with an estimate of 0.0017. However, all Bayesian models are much less uncertain about this estimate than their frequentist counterparts. B-EGARCHM, for example, gives this estimate the lowest standard deviation of all its own estimates, 0.0086. This could suggest that B-EGARCHM attributes a certain relative level of confidence in this estimate. S-GARCHM gives this estimate the second highest standard deviation of all its own estimates, 0.0028. This may indicate that S-GARCHM is more confident about its other estimates. On the other hand, most models rank FISV, the actual worst performer with a mean realized return of 0.0004, favourably. S-EGARCHM ranks it third, B-EGARCHM ranks it first. However, B-EGARCHM is very uncertain about this estimate: this is the estimate it is the most uncertain about, with a standard deviation of 0.1545. S-EGARCHM, on the contrary, ranks the uncertainty with this estimate of returns as ninth. Combining returns and estimates of uncertainty in a portfolio context may therefore be interesting.

6.6.2 Portfolios

Correlations between the daily returns of the portfolios are presented in Table 6.6. Most correlations are very high. The lowest correlation is 0.810, between the QQQ Index and the equal weights portfolio. The highest, excluding the diagonals, is 0.999 between the B-GARCH and B-GARCHM portfolios. This reflects the fact that the Black-Litterman model starts with market weights and makes adjustments based on predictions of returns. The fact that there are only 10 equities in the portfolio, and that these were selected based upon a high level of historical correlation with the QQQ Index may also explain the fact that even the daily returns of the *Equal Weights* portfolios have correlations in excess of 0.8 with portfolios constructed using the Black-Litterman model.

Participation ratios are presented in Table 6.7. In all cases, most weight is given to five stocks: AAPL, AMZN, GOOG, GOOGL, and MSFT. This is due to the large difference in market

Stock	S-GARCH	S-EGARCH	S-EGARCHM	B-GARCH	B-EGARCH	B-EGARCHM
AAPL	0.0207	-0.0606	-0.0687	-0.0237	-0.0606	-0.0687
ADP	0.0028	-0.0108	0.0222	0.0728	-0.0108	0.0222
AMGN	-0.0568	-0.0739	-0.0602	-0.0486	-0.0739	-0.0602
AMZN	-0.0487	-0.0626	-0.0475	-0.0523	-0.0626	-0.0475
FISV	-0.0233	-0.0116	-0.0173	-0.0465	-0.0116	-0.0173
GOOG	0.0118	-0.0494	-0.0476	-0.0330	-0.0494	-0.0476
GOOGL	0.0235	-0.0295	-0.0258	-0.0051	-0.0295	-0.0258
HSIC	-0.0113	0.0018	-0.0006	-0.0029	0.0018	-0.0006
MSFT	0.0221	-0.0219	0.0285	0.0346	-0.0219	0.0285
SBUX	0.0174	-0.0001	0.0141	0.0389	-0.0001	0.0141

Table 6.3: This table presents correlations between predicted and actual returns. None of the models predicted returns which had high correlations with actual returns. Correlations were sometimes negative. Predictions may nevertheless add value when being used to create portfolios, for example if they help identify those equities which are expected to over or under perform.

Stock	Actual	S-GARCH	S-EGARCH	S-EGARCHM	B-GARCH	B-EGARCH	B-EGARCHM
AAPL	0.0011	0.0005	0.0006	0.0003	0.0018	0.0017	0.0017
ADP	0.0016	0.0004	0.0003	0.0003	0.0005	0.0004	-0.0000
AMGN	0.0010	0.0002	0.0003	0.0001	-0.0010	-0.0009	-0.0010
AMZN	0.0010	0.0010	0.0010	0.0002	0.0013	0.0016	0.0015
FISV	0.0004	0.0005	0.0003	0.0002	0.0027	0.0072	0.0117
GOOG	0.0013	0.0004	0.0004	0.0000	-0.0010	-0.0005	-0.0007
GOOGL	0.0005	0.0003	0.0003	-0.0000	-0.0011	-0.0007	-0.0009
HSIC	0.0016	0.0002	0.0001	0.0002	-0.0007	0.0006	0.0026
MSFT	0.0007	0.0006	0.0006	0.0001	0.0007	0.0009	0.0006
SBUX	0.0006	0.0004	0.0004	-0.0001	0.0007	0.0001	0.0001

Table 6.4: This table presents the mean values of predicted daily returns. With the exception of B-EGARCHM, which estimates a slightly higher return for HSIC, AAPL, and AMZN relative to most other securities, no other approach seems to rank the highest performers overall. On the other hand, B-EGARCHM has a mean prediction for FISV (0.0117) which is far above the mean of realised returns (0.0004). This does not support the view that the models rank securities in order of increasing returns in order to overweight those securities with better returns.

capitalization between these companies and ADP, AMGN, FISV, HSIC, and SBUX. This issue could have been addressed by selecting equities with similar market capitalizations. This may have resulted in portfolios with better participation ratios, at the risk of being equivalent to selecting a large cap or small cap portfolio approach.

Details of the statistical properties of the portfolios generated using the views in Section 6.6.1 are presented in Table 6.8. Performance is measured using both the Sharpe ratio presented in Section 6.3.2.1 and the $M2$ ratio presented in Section 6.3.2.2. In order to assist intuition, the $M2$ ratios calculated on daily returns were annualized. Both the Sharpe ratio and the $M2$ were calculated using the *Federal Funds Rate* as the risk-free rate. The QQQ Index was presented for reference purposes, in order to provide some intuition that the returns obtained were reasonable. The QQQ Index cannot be used as a benchmark: since

Stock	Actual	S-GARCH	S-EGARCH	S-EGARCHM	B-GARCH	B-EGARCH	B-EGARCHM
AAPL	0.0159	0.0018	0.0020	0.0028	0.0095	0.0080	0.0086
ADP	0.0171	0.0009	0.0009	0.0022	0.0109	0.0143	0.0158
AMGN	0.0130	0.0011	0.0011	0.0027	0.0091	0.0083	0.0091
AMZN	0.0216	0.0020	0.0020	0.0029	0.0117	0.0084	0.0099
FISV	0.0142	0.0011	0.0011	0.0019	0.0233	0.1014	0.1545
GOOG	0.0176	0.0012	0.0015	0.0026	0.0131	0.0094	0.0107
GOOGL	0.0213	0.0012	0.0014	0.0027	0.0138	0.0109	0.0127
HSIC	0.0417	0.0015	0.0017	0.0029	0.0273	0.0154	0.0564
MSFT	0.0143	0.0010	0.0010	0.0023	0.0101	0.0068	0.0093
SBUX	0.0143	0.0009	0.0009	0.0017	0.0093	0.0080	0.0089

Table 6.5: This table presents the standard deviation of predicted daily returns. An interesting analysis arises when predicted returns in Table 6.4 are combined with the standard deviation of predictions presented here. Of all its predictions, B-EGARCHM calculates the highest standard deviation to its least successful prediction, FISV. This is contrary to the frequentist models, which appear relatively confident in their unsuccessful predictions. Combining returns and estimates of uncertainty in a portfolio context may therefore be interesting.

Correlations	QQQ	No Views	Equal Weights	S-GARCH	B-GARCH	S-EGARCH	B-EGARCH	S-EGARCHM	B-EGARCHM
QQQ	1.000	0.950	0.810	0.949	0.947	0.949	0.946	0.949	0.945
No Views	0.950	1.000	0.841	1.000	0.997	1.000	0.998	1.000	0.996
Equal Weights	0.810	0.841	1.000	0.835	0.823	0.835	0.842	0.834	0.830
S-GARCH	0.949	1.000	0.835	1.000	0.997	1.000	0.998	1.000	0.996
B-GARCH	0.947	0.997	0.823	0.997	1.000	0.997	0.996	0.997	0.999
S-EGARCH	0.949	1.000	0.835	1.000	0.997	1.000	0.998	1.000	0.996
B-EGARCH	0.946	0.998	0.842	0.998	0.996	0.998	1.000	0.998	0.997
S-EGARCHM	0.949	1.000	0.834	1.000	0.997	1.000	0.998	1.000	0.996
B-EGARCHM	0.945	0.996	0.830	0.996	0.999	0.996	0.997	0.996	1.000

Table 6.6: This table presents correlations between the daily returns of the portfolios. Most correlations are very high which reflects the fact that the Black-Litterman model starts with market weights and makes adjustments based on predictions of returns. The fact that there are only 10 equities in the portfolio, and that these were selected based upon a high level of historical correlation with the QQQ Index may also explain the fact that even the daily returns of the *Equal Weights* portfolios have correlations in excess of 0.8 with portfolios constructed using the Black-Litterman model.

Statistic	No Views	S-GARCH	B-GARCH	S-EGARCH	B-EGARCH	S-EGARCHM	B-EGARCHM
Participation Ratio	5.75	5.57	5.57	5.52	5.57	5.57	5.57

Table 6.7: This table presents participation ratios. In all cases, most weight is given to five stocks: AAPL, AMZN, GOOG, GOOGL, and MSFT, due to the large difference in market capitalization between these companies and ADP, AMGN, FISV, HSIC, and SBUX.

Model	Total Return	$\hat{S}$	SE($\hat{S}$)	P-Value	M2
QQQ	62.4%	0.0539	0.0362		18.7%
No Views	87.6%	0.0619	0.0362		21.8%
Equal Weights	92.8%	0.0738	0.0362	<.001	26.4%
S-GARCH	86.2%	0.0606	0.0362	0.46	21.2%
B-GARCH	89.7%	0.0622	0.0362	0.87	21.9%
S-EGARCH	86.6%	0.0607	0.0362	0.49	21.3%
B-EGARCH	81.8%	0.0580	0.0362	0.03	20.2%
S-EGARCHM	86.4%	0.0607	0.0362	0.50	21.3%
B-EGARCHM	86.0%	0.0602	0.0362	0.34	21.1%

Table 6.8: This table presents the statistical properties of portfolio daily returns. The portfolio with the highest return, Sharpe ratio, and $M2$ is the *Equal Weights* portfolio, which allocates 10% of the portfolio to each equity. The Sharpe Ratio of that portfolio is significantly different from the Sharpe Ratio of the *No Views* portfolio at a 5% level. The B-GARCH portfolio also has a higher Sharpe Ratio than the *No Views* portfolio, however the difference is not significantly different at a 5% level.

the portfolios were created using a subset of 10 equities, these may have individually over or underperformed the index in the period under investigation. A neutral portfolio consisting of these 10 equities in proportions equal to their market weights, which corresponds to the *No Views* portfolio, can however be used.

Total returns on the portfolios are of the same order of magnitude as total returns on the QQQ index and are therefore considered reasonable. The fact that returns on the QQQ are far below the returns on some portfolios can be attributed to the fact that the 10 equities which were selected turned out to have outperformed the index and is not indicative of the success of any algorithm.

The (annualized) Sharpe Ratio of the QQQ Fund is 0.93 when calculated against the ETF's benchmark[3]. When annualized, the ratio presented in Table 6.8, calculated against the risk-free rate, is 0.86. The ratios would not be expected to be equal due to the different approaches used, but the fact that they are similar provides some comfort that the correct methodology was applied.

Of greater interest is a comparison to the *No Views* portfolio constructed using a Black-Litterman algorithm allocating a weight to each equity corresponding to its market capitalization.

The portfolio with the highest return, Sharpe ratio, and $M2$ is the *Equal Weights* portfolio, which allocates 10% of the portfolio to each equity. The Sharpe Ratio of that portfolio is significantly different from the Sharpe Ratio of the *No Views* portfolio at a 5% level.

[3]https://screener.fidelity.com/ftgw/etf/goto/snapshot/performance.jhtml?symbols=QQQ

The B-GARCH portfolio also has a higher Sharpe Ratio than the *No Views* portfolio, however the difference is not significantly different at a 5% level.

All other portfolios either match or underperform the benchmark portfolio on a risk adjusted basis.

6.7 Discussion

This experiment has demonstrated how Bayesian networks may be used to generate views which can be used to construct portfolios. In support of the efficient market hypothesis, the portfolios constructed in this experiment did not outperform their benchmark nevertheless the performance of the portfolios were not far from their benchmark. Further research may therefore yield better results.

The scientific analysis platform which can be used to easily explore and compare models was finalized. Further research should be performed using different factors as input to Bayesian networks. The platform can also easily accommodate other machine learning models.

As seen in Table 6.7, most weight is given to only five stocks due to the large difference in market capitalization between these companies and the five others. Perhaps equity selection was more important than expected. Taking size into account when making equity selection may be useful in future work.

Chapter 7

Conclusions and Future Research

This chapter concludes the thesis and presents the key findings of its experiments. It summarizes the contributions of this work, and identifies areas of further research drawn from this thesis.

7.1 Summary

The main objective of this thesis was to investigate how Bayesian networks could be used to generate views that result in portfolios with lower risk or higher returns compared to a benchmark neutral market portfolio. In current approaches to asset management, views are subjective, time-consuming, and expensive to produce. This thesis aimed to improve on current approaches. Bayesian networks were explored because they enable the measurement of the uncertainty of predictions as well as the inclusion of expert knowledge through the use of a prior.

A second objective was to develop a scientific platform incorporating machine learning in portfolio management, in order to rapidly test machine learning models and improve the communication of research results with the investment community.

The thesis was divided into three experiments. The first focused on the development and testing of a Black-Litterman portfolio model as well as the scientific platform. The second focused on the development and testing of a number of Bayesian networks to generate views. The third combined the results of the first two into a comprehensive tool capable of using different machine learning models and portfolio analysis models.

7.1.1 A Benchmark Portfolio Using Traditional Algorithms

The first experiment established benchmarks against which other models were compared. It begins the development of the scientific platform detailed in Figure 6.1, focussing on the portfolio construction model, its audit trail, and back testing. Data used were the prices of all equities which comprised the NASDAQ-100 over the period from 31/12/1994 to 30/09/2017.

A custom benchmark was necessary because, in Experiment 3, a small subset of all equities in the NASDAQ-100 was used. A direct comparison to either the NASDAQ-100 Index or the QQQ ETF which tracks the NASDAQ-100 could therefore have been biased.

Two portfolios were created: one by equal weighting all equities, another using a Black-Litterman portfolio construction model without views. The scientific analysis platform was presented via a web front-end, along with Excel-based audit trails generated automatically from the python code.

7.1.2 Generating Views Using a Bayesian Network

The second experiment explores the use of Bayesian networks to generate views for the Black-Litterman model developed in Experiment 1, and develops the tools necessary to understand and evaluate the predictions. The Bayesian networks use a number of different functional forms for the relationship between factors and returns: linear regression, AR(1), GARCH, EGARCH, and EGARCHM.

This experiment explored different tools to build the Bayesian network, finally selecting PyMC3 . Another strong option would have been STAN, although recent develops with Pyro using PyTorch may make that the best option in the future.

The factors considered by the Bayesian network include changes in oil prices, credit spreads, the term structure of interest rates, and a stress index.

Implementations GARCH, EGARCH, and EGARCHM in Bayesian networks were tested on synthetic data generated in Microsoft Excel to verify that models were able to recover the original known parameters of the process. All models were then implemented to generate predictions for monthly returns of APPL shares from 30.07.1999 to 30.09.2017. Predictions were compared to realized returns. Correlation coefficients between predicted and realized returns were low and sometimes negative. It was not possible to provide evidence that predicted returns and realized returns were similarly distributed. Only linear regression, AR(1), and EGARCH-M provided reasonable estimates of the uncertainty of predictions. The predictions may still be useful, for example if, when combined with the market returns predicted by the Black-Litterman model, they result in a ranking of the equities which corresponds to realized returns. Experiment 3, using the predictions in the context of portfolio construction, will enable further conclusions.

While the main objectives of this experiment were met, further research is required in order to implement Bayesian networks in a portfolio management context.

7.1.3 Algorithmic Asset Management System

This experiment aimed to investigate how views generated using a Bayesian network could be useful in portfolio construction, and to finalize the scientific analysis platform developed throughout this thesis. Following the platform, a subset of 10 equities which were part of the

NASDAQ-100 Index were selected based on correlation of returns with returns of the Index. The features of a set of factors was extracted using principal component analysis. Bayesian networks with statistical and Bayesian GARCH, EGARCH, and EGARCHM models were used as input to a Black-Litterman portfolio construction model to create six sets of portfolios. The performance of these portfolios over the period from 31 July 2016 to 1 August 2019 with daily rebalancing was compared to a neutral portfolio created using a Black-Litterman model with no views, and an equally weighted portfolio. Sharpe ratios, Modigliani risk-adjusted performance, the Herfindahl-Hirschman Index, as well as common statistical measures were used to understand the performance of the portfolios.

This experiment demonstrates how Bayesian networks may be used to generate views and how those views can be used to construct portfolios. It applies this approach with Bayesian networks in realistic daily trading conditions, utilizes these views to create portfolios using a Black-Litterman portfolio optimization model, and demonstrates tools to compare the performance of these portfolios. In support of the hypothesis that markets are efficient, the portfolios constructed in this experiment were below though close to their benchmark. This may be considered further validation of the efficient market hypothesis.

This experiment finalizes the analysis platform developed throughout this thesis. The platform has full flexibility to use different means of analysis, algorithms for generating views, portfolio management models, and data sets. This platform can be used to easily explore and compare different models for any of the critical roles identified in this approach.

While the main objectives of this experiment were met, further research is required in order to build Bayesian networks which better predict returns as well as estimates of the uncertainty of the predictions in a portfolio management context.

7.2 Contributions

This thesis provides important contributions to the scientific body of knowledge.

First, this thesis presents a scientific analysis platform which can guide the use of machine learning in asset management. This platform can be used as a clear communication tool with members of the asset management industry as it represents an approach they would understand. This thesis also presents in detail one way the platform may be implemented.

Second, this thesis appears to be the first complete exploration of the use of Bayesian networks in asset management which reports on the results of the performance of the portfolios constructed using the models. Previous work appears to be limited to one model, and does not report on the performance of portfolios constructed using the model predictions. Lack of conclusive results, as was the case in this thesis, may explain this absence.

7.3 Future Work

This section discusses future work which should be undertaken based on the results obtained in this thesis.

Using the scientific analysis platform presented in Figure 3.4, future work should focus on factors, data, dimensionality reduction, and algorithms for the prediction of risk and returns. In addition, it should consider some of the issues identified in this thesis.

Future research should consider the use of a broader range of factors, as discussed in Section 6.5.2.1. Research is currently being undertaken at UCL to focus on this creation of a different set of factors based on news. As part of that research, these new sets of factors would be used in the platform to relatively quickly compare the performance of portfolios generated with and without these factors. Further research should consider factors for which data of the required frequency is not available. Missing data could be modelled separately or within the Bayesian network itself, as a separate model. For example, perhaps inflation could usefully be modelled using simple regression within the Bayesian network.

Future research should aim to replicate and extend the results of this thesis by applying the approach to different asset classes. Focussing on specific industries may be useful as some may be more easily modelled than others. In particular, it may be possible to make use of expert knowledge in some industries to specify better models. Another focus could be to consider companies with high market capitalization separately from those with small market caps, as the approach may be better suited to smaller or larger companies. In general, it may be possible to find assets whose returns can be predicted with a higher level of confidence. For example, more news data may be available for some companies, and therefore their returns may be more susceptible to prediction.

The time required for prediction will be a constraint for the foreseeable future. Dimensionality reduction will therefore remain an issue which should be investigated in future studies. It can be addressed, for example, by either selecting only those factors which are most relevant to a specific equity in the specific economic and trading environment at the time for which the prediction is made, by distilling relevant features from a broader range of factors, or using both approaches at the same time.

Future research should investigate different algorithms for the prediction of risk and return. Different functional forms for the relationships between factors and returns should be considered. Candidates include different functional forms in the GARCH family of models. Other algorithms including deep learning, reinforcement learning, or Generative Adversarial Networks should be investigated. Many approaches may be used without substantially modifying the platform presented in Figure 3.4. Work which includes an investigation of very long tuning times could be interesting.

Finally, this research had limitations which could be taken into consideration in future work.

Portfolios constructed in the third experiment were concentrated in five out of a total of ten equities. Future work could consider using equities of similar market capitalization to avoid this situation.

This research only uses closing prices. The added complexity of taking closing bid-ask spreads into account by "selling" at the lower bid price and "buying" at the higher ask price would be useful once models can be shown to useful. This should be included in future work.

Appendix A

Notation

The following notation is used throughout this thesis.

n	number of assets
$\boldsymbol{w}$	an $(n \times 1)$ vector of weights for all assets in a portfolio
$\boldsymbol{w}^*$	an $(n \times 1)$ vector of weights in an efficient portfolio
$\boldsymbol{\mu}$	an $(n \times 1)$ vector of mean asset returns, in excess of the risk free rate
μ_0	the required portfolio (excess) return
μ_p	mean portfolio (excess) return
μ_m	mean (excess) return of a benchmark or market portfolio
$\boldsymbol{\Sigma}$	the $(n \times n)$ non-singular covariance matrix of asset (excess) returns
$\boldsymbol{\Sigma_p}$	Sample variance matrix including impact of views
σ_m	volatility of a benchmark or market portfolio.
σ_p^2	portfolio variance
δ	risk aversion parameter
$\boldsymbol{\Pi}$	an $(n \times 1)$ vector of equilibrium excess returns
τ	scalar value indicating the uncertainty of the CAPM prior
k	number of views
$\boldsymbol{P}$	a $(k \times n)$ matrix of asset weights within each view;
	for absolute views, the sum of the weights is 1, otherwise 0
$\boldsymbol{Q}$	a $(k \times 1)$ matrix of the returns for each view
$\boldsymbol{\Omega}$	a $(k \times k)$ matrix of the covariance of the views
$\boldsymbol{M}$	an $(n \times n)$ posterior variance matrix

Table A.1: Notation

Appendix B

Acronyms

The following acronyms are used in this thesis.

CVaR	Conditional Value-at-Risk or Expected Shortfall
EGARCH	Exponential generalized autoregressive conditional heteroskedasticity process
EGARCH-M	EGARCH in Mean process
EMH	Efficient Market Hypothesis
FRED	Federal Reserve Bank of St-Louis
GARCH	Generalized autoregressive conditional heteroskedasticity process
GARCH-M	GARCH in Mean process
HMC	Hamiltonian Monte Carlo
MCMC	Markov Chain Monte Carlo
MVA	Model-View-Adapter architectural pattern
NUTS	No U-Turn Sampler
SAAS	Software as a Service
VaR	Value-at-Risk

Table B.1: List of acronyms

Appendix C

Data

Source of daily equity prices.

Index	Ticker	Company	From	To	Yahoo	Quandl	None
1	AAL	American Airlines Group	31/12/2014	30/09/2017	AAL		
2	AAPL	Apple Inc.	31/03/1995	30/09/2017	AAPL		
3	ABGX	Abgenix, Inc.	31/12/2000	31/12/2002			ABGX
4	ACCOB	Adolph Coors Company	31/03/1995	31/12/1995			ACCOB
5	ADBE	Adobe Systems Inc.	31/03/1995	30/09/2017	ADBE		
6	ADCT	ADC Telecommunications, Inc.	31/03/1995	31/12/2003			ADCT
7	ADI	Analog Devices	31/12/2012	30/09/2017	ADI		
8	ADLAE	Adelphia Communications Corp.	31/12/1999	30/06/2002			ADLAE
9	ADP	ADP, Inc.	31/12/2008	30/09/2017	ADP		
10	ADPT	Adaptec, Inc.	31/03/1995	31/12/2000			ADPT
11	ADRX	Andrx Group	31/12/2001	31/12/2002			ADRX
12	ADSK	Autodesk, Inc.	31/03/1995	31/12/1999	ADSK		
13	ADTN	ADTRAN, Inc.	31/12/1996	31/12/1998	ADTN		
14	AEOS	American Eagle Outfitters, Inc.	31/12/2006	31/03/2007			AEOS
15	AESC	AES Corp.	31/03/1995	30/09/1996			AESC
16	AGREA	American Greetings Corp.	31/03/1995	31/03/1998			AGREA
17	AKAM	Akamai Technologies, Inc.	30/04/2006	31/12/2009	AKAM		
18	AKLM	Acclaim Entertainment, Inc.	31/03/1995	31/12/1996			AKLM
19	ALEX	Alexander & Baldwin, Inc.	31/03/1995	31/12/1995	ALEX		
20	ALTR	Altera Corp.	30/09/1995	31/10/2015	ALTR		
21	ALXN	Alexion Pharmaceuticals	31/05/2011	30/09/2017	ALXN		
22	AMAT	Applied Materials, Inc.	31/03/1995	30/09/2017	AMAT		
23	AMCC	Applied Micro Circuits Corp.	31/12/1999	31/12/2002		AMCC	
24	AMFM	Chancellor Media Corp.	31/03/1998	30/09/1999			AMFM
25	AMGN	Amgen Inc.	31/03/1995	30/09/2017	AMGN		
26	AMLN	Amylin Pharmaceuticals	31/01/2006	31/12/2008	AMLN		
27	AMZN	Amazon.com, Inc.	31/12/1998	30/09/2017	AMZN		
28	ANDW	Andrew Corp.	31/03/1995	31/12/1999			ANDW
29	APCC	APC Corp.	31/03/1995	31/12/2000			APCC
30	APOL	Apollo Group	31/03/1998	31/12/2000		APOL	
31	ARBA	Ariba, Inc.	31/12/2000	31/12/2001	ARBA		
32	ASAI	Atlantic Southeast Airlines, Inc.	31/03/1995	31/12/1996			ASAI
33	ASND	Ascend Communications, Inc.	30/09/1996	30/06/1999	ASND		
34	ASTA	AST Research, Inc.	31/03/1995	31/12/1995			ASTA
35	ATHMQ	At Home Corp.	31/03/1999	31/12/2001			ATHMQ
36	ATML	Atmel Corp.	31/03/1995	31/12/2002		ATML	
37	ATVI	Activision Blizzard	31/12/2005	30/09/2017	ATVI		
38	ATYT	ATI Technologies Inc.	31/12/2003	31/10/2006			ATYT
39	AVGO	Broadcom Limited	31/12/2011	30/09/2017	AVGO		
40	AWIN	Allied Waste Industries, Inc	31/03/1998	31/12/1998			AWIN
41	BBBY	Bed Bath & Beyond Inc.	31/12/1996	30/09/2017	BBBY		
42	BEAS	BEA Systems, Inc.	31/12/2000	30/04/2008			BEAS
43	BGEN	Biogen, Inc. (Old)	31/03/1995	31/12/2003			BGEN
44	BIDU	Baidu.com, Inc.	31/12/2007	30/09/2017	BIDU		
45	BIIB	Biogen, Inc.	31/12/2000	30/09/2017	BIIB		
46	BMC	BMC Software, Inc.	31/03/1995	31/03/2001			BMC
47	BMET	Biomet, Inc.	31/03/1995	31/07/2007			BMET
48	BMRN	BioMarin Pharmaceutical	31/07/2015	30/09/2017	BMRN		
49	BNET	Bay Networks, Inc.	31/03/1995	31/03/1996	BNET		
50	BOBE	Bob Evans Farms, Inc.	31/03/1995	31/12/1996		BOBE	
51	BOST	Boston Chicken, Inc.	31/12/1995	31/03/1998			BOST
52	BRCD	Brocade Communications Systems	31/03/2001	31/12/2003	BRCD		
53	BRCM	Broadcom Corp.	31/12/2000	30/11/2015		BRCM	
54	BRNO	Bruno's Inc.	31/03/1995	30/09/1995			BRNO
55	BVSN	BroadVision, Inc.	31/12/1999	31/12/2001	BVSN		
56	CA	CA, Inc.	31/05/2008	30/09/2017	CA		
57	CATP	Cambridge Technology Partners	31/03/1998	31/12/1999			CATP
58	CBRL	CBRL Group Inc.	31/03/1995	31/12/1999	CBRL		
59	CDNS	Cadence Design Systems	31/12/2005	31/12/2008	CDNS		
60	CDWC	CDW Corp.	31/12/2001	31/10/2007			CDWC
61	CECO	Career Education Corp.	31/12/2003	31/12/2005	CECO		
62	CEFT	Concord EFS, Inc.	31/12/1996	31/12/2002			CEFT
63	CELG	Celgene Corp.	31/07/2005	30/09/2017	CELG		
64	CEPH	Cephalon, Inc.	31/12/2001	31/12/2004			CEPH

65	CERN	Cerner Corp.	31/07/2009	30/09/2017	CERN		
66	CEXP	Corporate Express, Inc.	31/12/1996	31/12/1999			CEXP
67	CHIR	Chiron Corp.	31/03/1995	30/04/2006			CHIR
68	CHKP	Check Point Ltd.	31/12/2000	30/09/2017	CHKP		
69	CHRS	Charming Shoppes, Inc.	31/03/1995	31/12/1995	CHRS		
70	CHRW	C. H. Robinson Worldwide, Inc.	31/12/2002	31/12/2015	CHRW		
71	CHTR	Charter Communications, Inc.	31/12/2001	31/12/2002	CHTR		
72	CIEN	CIENA Corp.	30/09/1999	31/12/2003	CIEN		
73	CKFR	CheckFree Corp.	31/12/2005	31/12/2007			CKFR
74	CMCSA	Comcast Corp. Class A Common	31/12/2002	30/09/2017	CMCSA		
75	CMCSK (Old)	Comcast Corp. (Old)	31/03/1995	31/12/2002			CMCSK (Old)
76	CMGI	CMGI, Inc.	31/03/1999	31/12/2001			CMGI
77	CMVT	Comverse Technology, Inc.	31/03/1999	28/02/2007	CMVT		
78	CNET	CNET Networks, Inc.	30/06/1999	31/12/2001	CNET		
79	CNTO	Centocor, Inc.	31/03/1995	31/12/1999	CNTO		
80	CNXT	Conexant Systems, Inc.	30/09/1999	31/12/2002	CNXT		
81	COMR	Comair Holdings, Inc.	31/12/1998	31/12/1999			COMR
82	COMS	3Com Corp.	31/03/1995	31/12/2001			COMS
83	COST	Costco Wholesale Corp.	31/03/1995	30/09/2017	COST		
84	CPWR	Compuware Corp.	31/03/1995	31/12/2004	CPWR		
85	CRUS	Cirrus Logic, Inc.	31/03/1995	31/03/1998	CRUS		
86	CSCC	Cascade Communications Corp.	30/09/1996	30/09/1997			CSCC
87	CSCO	Cisco Systems, Inc.	31/03/1995	30/09/2017	CSCO		
88	CSX	CSX Corporation	29/02/2016	30/09/2017	CSX		
89	CTAS	Cintas Corp.	31/03/1995	30/04/2011	CTAS		
90	CTRP	Ctrip	31/12/2010	31/07/2012	CTRP		
91	CTRX	Catamaran Corp.	31/12/2012	31/07/2015		CTRX	
92	CTSH	Cognizant Technology Corp.	31/12/2004	30/09/2017	CTSH		
93	CTXS	Citrix Systems, Inc.	31/03/1998	30/09/2017	CTXS		
94	CYTC	Cytyc Corp.	31/12/2001	31/12/2002			CYTC
95	DELL	Dell, Inc.	31/03/1995	31/10/2013		DELL	
96	DIGI	DSC Communications Corp.	31/03/1995	30/09/1998	DIGI		
97	DISCA	Discovery Comms Class A	31/12/2005	31/12/2008	DISCA		
98	DISH	Dish Network, Inc.	31/12/1999	31/12/2010	DISH		
99	DLTR	Dollar Tree, Inc.	31/12/1998	31/12/2000	DLTR		
100	DTV	DirecTV	30/04/2008	31/07/2015	DTV		
101	DURA	Dura Pharmaceuticals, Inc.	31/03/1998	31/12/1998			DURA
102	EA	Electronic Arts	31/03/1995	31/12/2012	EA		
103	EBAY	eBay Inc.	31/12/1999	30/09/2017	EBAY		
104	EFII	Electronics for Imaging, Inc.	31/12/1996	31/12/1999	EFII		
105	ENDP	Endo International	31/12/2015	31/07/2016	ENDP		
106	EQIX	Equinix, Inc.	31/12/2012	31/03/2015	EQIX		
107	ERIC	Ericsson Telephone Company	31/03/1998	31/12/2003	ERIC		
108	ESRX	Express Scripts, Inc.	31/12/2001	30/09/2017	ESRX		
109	EXDSQ	Exodus Communications, Inc.	31/12/2000	31/12/2001			EXDSQ
110	EXPD	Expeditors Int'l, Inc.	31/12/2002	31/12/2015	EXPD		
111	EXPE	Expedia, Inc.	31/12/2005	31/12/2014	EXPE		
112	FAST	Fastenal Co.	31/12/1996	31/12/1999	FAST		
113	FB	Facebook, Inc.	31/12/2012	30/09/2017	FB		
114	FDLNB	Food Lion, Inc.	31/03/1995	30/09/1999			FDLNB
115	FFIV	F5 Networks	30/04/2011	31/12/2014	FFIV		
116	FHCC	First Health Group Corp.	31/03/1995	31/12/1999			FHCC
117	FISV	Fiserv, Inc.	31/12/1996	30/09/2017	FISV		
118	FLEX	Flextronics Int'l Ltd.	31/12/2000	31/12/2012	FLEX		
119	FLIR	FLIR Systems	31/07/2008	31/12/2011	FLIR		
120	FMCN	Focus Media Holding	31/12/2007	31/01/2009	FMCN		
121	FORE	FORE Systems, Inc.	30/09/1996	30/06/1999			FORE
122	FORT	Fort Howard Corp.	31/12/1996	30/09/1997			FORT
123	FOSL	Fossil Inc.	31/12/2011	31/12/2013	FOSL		
124	FOXA	21st Century Fox Class A	31/01/2009	30/09/2017	FOXA		
125	FSLR	First Solar, Inc.	31/12/2008	30/04/2012	FSLR		
126	FWLT	Foster Wheeler AG	31/07/2007	31/12/2010			FWLT
127	GART	Gartner Group, Inc.	31/12/1995	31/03/1998			GART
128	GATE	Gateway 2000, Inc.	31/12/1995	30/06/1997			GATE
129	GBLX	Global Crossing Ltd.	30/09/1999	31/12/2000	GBLX		

130	GEMS	Glenayre Technologies, Inc.	31/12/1995	31/03/1998	GEMS		
131	GENZ	Genzyme Corp.	31/03/1995	31/05/2011			GENZ
132	GIDL	Giddings & Lewis, Inc.	31/03/1995	31/12/1996			GIDL
133	GILD	Gilead Sciences, Inc.	31/12/2001	30/09/2017	GILD		
134	GMCR	Keurig Green Mountain	31/05/2011	31/12/2012		GMCR	
135	GMSTE	Gemstar-TV Guide International	31/12/1999	31/12/2002			GMSTE
136	GNCI	General Nutrition Companies, Inc.	31/12/1995	31/12/1998			GNCI
137	GNTX	Gentex Corp.	31/12/2002	31/12/2004	GNTX		
138	GOLD	Randgold Resources	31/12/2011	30/11/2013	GOLD		
139	GOOGL	Alphabet Inc. Class A	31/12/2005	30/09/2017	GOOGL		
140	GRMN	Garmin Ltd.	31/12/2003	31/12/2015	GRMN		
141	HBOC	HBO & Company	31/03/1995	31/03/1999			HBOC
142	HGSI	Human Genome Sciences, Inc.	31/12/2000	31/12/2003			HGSI
143	HOLX	Hologic	31/12/2007	31/12/2010	HOLX		
144	HONI	Hon Industries Inc.	31/03/1995	31/12/1995			HONI
145	HSIC	Henry Schein, Inc.	31/12/2002	31/12/2004	HSIC		
146	IACI	IAC/InterActiveCorp	31/03/1998	30/09/1998	IACI		
147	ICOS	ICOS Corp.	31/12/2001	31/12/2003			ICOS
148	IDTI	Integrated Device Technology, Inc.	31/12/2001	31/12/2002	IDTI		
149	IDXX	IDEXX Laboratories, Inc.	31/12/1995	31/03/1998	IDXX		
150	IFMX	Informix Corp.	31/03/1995	31/03/1998			IFMX
151	ILMN	Illumina, Inc.	31/12/2008	31/12/2011	ILMN		
152	IMCL	ImClone Systems Inc.	31/12/2001	31/12/2002			IMCL
153	IMNX	Immunex Corp.	31/03/1998	30/09/2002			IMNX
154	INCY	Incyte Corp	31/10/2015	30/09/2017	INCY		
155	INEL	Intelligent Electronics, Inc.	31/03/1995	31/12/1995			INEL
156	INFY	Infosys Ltd	31/12/2006	31/12/2012	INFY		
157	INKT	Inktomi Corp.	31/12/2000	31/12/2001			INKT
158	INTC	Intel Corp.	31/03/1995	30/09/2017	INTC		
159	INTU	Intuit, Inc.	31/12/1995	31/03/1998	INTU		
160	ISIL	Intersil Corp.	31/12/2003	31/12/2005		ISIL	
161	ISRG	Intuitive Surgical Inc.	28/02/2006	30/09/2017	ISRG		
162	ITWO	i2 Technologies, Inc.	31/12/1999	31/12/2002			ITWO
163	JBHT	J. B. Hunt, Inc.	31/03/1995	31/12/1996	JBHT		
164	JCOR	Jacor Communications Inc.	31/03/1998	30/06/1999			JCOR
165	JD	JD.com	31/07/2015	30/09/2017	JD		
166	JDSU	JDS Uniphase Corp.	30/06/1999	31/12/2006		JDSU	
167	JNPR	Juniper Networks	30/09/2000	31/10/2009	JNPR		
168	JOYG	Joy Global	31/01/2006	31/12/2011			JOYG
169	KELYA	Kelly Services, Inc.	31/03/1995	31/12/1996	KELYA		
170	KHC	Kraft Heinz Co	31/03/2013	30/09/2017	KHC		
171	KLAC	KLA-Tencor Corp.	31/03/1995	29/02/2016	KLAC		
172	KMAG	Komag, Inc.	31/12/1996	31/03/1998			KMAG
173	LAMR	Lamar Advertising Co.	31/12/2002	31/12/2008	LAMR		
174	LBTYA	Liberty Global plc Class A	31/12/2004	31/12/2009	LBTYA		
175	LCOS	Lycos, Inc.	30/06/1999	31/12/2000			LCOS
176	LEAP	Leap Wireless International	31/10/2007	31/12/2008	LEAP		
177	LGNT	LEGENT Corp.	31/03/1995	30/09/1995			LGNT
178	LGTO	Legato Systems, Inc.	31/12/1999	31/12/2000			LGTO
179	LIFE	Life Technologies	31/12/2001	31/12/2005	LIFE		
180	LINB	LIN Broadcasting Corp.	31/03/1995	31/12/1995			LINB
181	LLTC	Linear Technology Corp.	31/03/1995	30/09/2017		LLTC	
182	LNCR	Lincare Holdings Inc.	31/12/1998	31/12/1999	LNCR		
183	LOGI	Logitech	28/02/2007	31/12/2010	LOGI		
184	LOTC	Lotus Development Corp.	31/03/1995	30/09/1995			LOTC
185	LRCX	Lam Research	31/03/1995	31/12/1996	LRCX		
186	LVLT	Level 3 Communications	30/09/1998	31/12/2001		LVLT	
187	MAR	Marriott International, Inc.	30/11/2013	30/09/2017	MAR		
188	MAT	Mattel, Inc.	31/12/2009	30/09/2017	MAT		
189	MCCRK	McCormick & Company, Inc.	31/03/1995	30/06/1999			MCCRK
190	MCHP	Microchip Technology	30/06/1997	31/12/2013	MCHP		
191	MCIC	MCI Communications Corp.	31/03/1995	31/03/1998	MCIC		
192	MCIP	MCI, Inc.	31/12/2004	31/01/2006			MCIP
193	MCLD	McLeodUSA Inc.	31/12/1998	31/12/2001			MCLD
194	MDLZ	Mondelēz International	31/07/2012	30/09/2017	MDLZ		

195	MEDI	MedImmune, Inc.	31/12/1999	30/06/2007			MEDI
196	MERQE	Mercury Interactive Corp.	31/12/2000	31/01/2006			MERQE
197	MFNX	Metromedia Fiber Network, Inc.	31/12/1999	31/12/2001			MFNX
198	MFST	MFS Communications Company, Inc.	30/09/1995	31/12/1996	MFST		
199	MICC	Millicom Int'l Cellular	31/05/2006	31/05/2011			MICC
200	MLHR	Herman Miller, Inc.	31/03/1995	31/12/1996	MLHR		
201	MLNM	Millennium Pharm. Inc.	31/12/2000	31/12/2005			MLNM
202	MMEDC	MultiMedia, Inc.	31/03/1995	31/12/1995			MMEDC
203	MNST	Monster Beverage	31/12/2007	31/12/2009	MNST		
204	MOLX	Molex Inc.	31/03/1995	31/12/2005	MOLX		
205	MRVL	Marvell Technology Group	31/12/2003	31/12/2012	MRVL		
206	MSFT	Microsoft Corp.	31/03/1995	30/09/2017	MSFT		
207	MTEL	Mobile Telecommunication Tech.	31/03/1995	31/12/1996			MTEL
208	MU	Micron Technology, Inc.	31/12/1995	31/12/1999	MU		
209	MWW	Monster Worldwide, Inc.	31/12/2000	31/12/2003	MWW		
210	MXIM	Maxim Integrated Products	31/12/1995	31/10/2007	MXIM		
211	MYL	Mylan, Inc.	31/12/2009	30/09/2017	MYL		
212	NCLH	Norwegian Cruise Line Holdings	31/12/2015	30/09/2017	NCLH		
213	NDSN	Nordson Corp.	31/03/1995	31/12/1995	NDSN		
214	NETA	Network Associates, Inc.	31/12/1996	31/12/2000			NETA
215	NFLX	Netflix	31/12/2010	31/12/2012	NFLX		
216	NIHD	NII Holdings	31/12/2005	31/12/2011	NIHD		
217	NOBE	Nordstrom, Inc.	31/03/1995	30/06/1999			NOBE
218	NOVL	Novell, Inc.	31/03/1995	31/12/2001			NOVL
219	NSCP	Netscape Communications Corp.	30/06/1997	31/03/1999			NSCP
220	NSOL	Network Solutions, Inc.	31/12/1999	30/06/2000			NSOL
221	NTAP	NetApp, Inc.	31/12/1999	30/09/2017	NTAP		
222	NTES	NetEase	31/03/2016	30/09/2017	NTES		
223	NTLI	NTL Inc.	31/12/1998	31/12/2000			NTLI
224	NUAN	Nuance Communications Inc.	31/12/2011	31/12/2013	NUAN		
225	NVDA	NVIDIA Corp.	30/06/2001	31/12/2004	NVDA		
226	NVLS	Novellus Systems Inc.	31/03/2001	31/12/2005			NVLS
227	NWAC	Northwest Airlines Corp.	30/09/1995	31/12/2000			NWAC
228	NXPI	NXP Semiconductors	31/12/2013	30/09/2017	NXPI		
229	NXTL	Nextel Communications, Inc.	31/03/1995	31/08/2005			NXTL
230	OFIS	U.S. Office Products Company	30/09/1997	31/12/1998			OFIS
231	ORCL	Oracle Corp.	31/03/1995	31/07/2013	ORCL		
232	ORLY	O'Reilly Automotive, Inc.	31/12/2008	30/09/2017	ORLY		
233	OSSI	Outback Steakhouse, Inc.	31/03/1995	31/03/1998			OSSI
234	OXHP	Oxford Health Plans, Inc.	31/03/1996	31/12/1998			OXHP
235	PAGE	Paging Network, Inc.	31/03/1995	31/03/1998			PAGE
236	PAIR	PairGain Technologies, Inc.	31/12/1996	31/12/1998			PAIR
237	PALM	Palm, Inc.	31/12/2000	31/12/2001			PALM
238	PAYX	Paychex, Inc.	31/03/1995	30/09/2017	PAYX		
239	PCAR	PACCAR Inc.	31/03/1995	30/09/2017	PCAR		
240	PCLN	The Priceline Group	31/10/2009	30/09/2017	PCLN		
241	PDCO	Patterson Companies	31/12/2002	31/12/2010	PDCO		
242	PDLI	PDL Biopharma Inc	31/12/2001	31/12/2002	PDLI		
243	PETM	PetSmart	31/12/1995	31/03/1998		PETM	
244	PHSY	PacifiCare Health Systems, Inc	31/03/1995	31/12/2000			PHSY
245	PHYB	Pioneer Hi-Bred International, Inc.	31/03/1995	31/12/1995			PHYB
246	PHYC	PhyCor, Inc.	31/12/1996	31/12/1998			PHYC
247	PIXR	Pixar Animation Studios	31/12/2002	31/05/2006			PIXR
248	PMCS	PMC - Sierra, Inc.	31/12/1999	31/12/2002	PMCS		
249	PMTC	Parametric Technology Corp.	31/03/1995	31/12/2001			PMTC
250	PPDI	PPD, LLC. (Pharmaceutical)	31/12/2008	31/12/2009			PPDI
251	PRGO	Perrigo Company plc	31/03/1995	31/12/1996	PRGO		
252	PSFT	PeopleSoft, Inc.	31/12/1995	31/12/2004	PSFT		
253	PTCM	Pacific Telecom, Inc.	31/03/1995	30/09/1995			PTCM
254	PTEN	Patterson-UTI Energy, Inc.	31/12/2002	31/12/2004	PTEN		
255	PYPL	PayPal Holdings	30/11/2015	30/09/2017	PYPL		
256	QCOM	Qualcomm, Inc.	31/03/1995	30/09/2017	QCOM		
257	QGEN	Qiagen	31/12/2009	31/12/2011	QGEN		
258	QLGC	QLogic Corp.	31/12/1999	31/12/2005		QLGC	
259	QNTM	Quantum Corp.	31/03/1995	30/09/1999			QNTM

260	QTRN	Quintiles Transnational Corp.	31/12/1996	31/12/2000			QTRN
261	QVCA	Liberty Interactive	31/10/2006	30/09/2017			QVCA
262	QWST	Qwest Communications Int'l	30/09/1998	31/12/1999			QWST
263	RATL	Rational Software Corp.	31/12/2000	31/12/2002			RATL
264	RDRT	Read-Rite Corp.	31/12/1995	31/12/1996			RDRT
265	REGN	Regeneron Pharmaceuticals	31/12/2012	30/09/2017	REGN		
266	RFMD	RF Micro Devices, Inc.	31/12/1999	31/12/2003		RFMD	
267	RHAT	Red Hat, Inc.	31/12/2005	31/12/2006			RHAT
268	RIMM	Research in Motion	31/12/2003	31/12/2012			RIMM
269	RNWK	RealNetworks, Inc.	31/12/1999	31/12/2001	RNWK		
270	ROAD	Roadway Services, Inc.	31/03/1995	31/12/1995			ROAD
271	ROST	Ross Stores Inc.	30/06/1997	31/12/1999	ROST		
272	RPOW	RPM, Inc.	31/03/1995	31/03/1998			RPOW
273	RTRSY	Reuters Group PLC	31/03/1998	31/12/1999			RTRSY
274	RWIN	Republic Industries, Inc.	30/09/1996	30/06/1997			RWIN
275	RXSD	Rexall Sundown, Inc.	31/03/1998	31/12/1999			RXSD
276	RYAAY	Ryanair	31/12/2002	31/12/2004	RYAAY		
277	SANM	Sanmina Corp.	31/12/1998	31/12/2005	SANM		
278	SBAC	SBA Communications	31/12/2012	30/09/2017	SBAC		
279	SBUX	Starbucks Corp.	31/12/1996	30/09/2017	SBUX		
280	SDLI	SDL, Inc.	31/12/1999	31/03/2001			SDLI
281	SEBL	Siebel Systems, Inc	30/06/1999	28/02/2006			SEBL
282	SEPR	Sepracor Inc.	31/12/2001	31/12/2002			SEPR
283	SHLD	Sears Holdings Corp.	30/09/2004	31/12/2013	SHLD		
284	SHLM	A. Schulman, Inc.	31/03/1995	31/12/1996	SHLM		
285	SIAL	Sigma-Aldrich Corp.	31/03/1995	31/12/2000		SIAL	
286	SIRI	Sirius XM Radio, Inc.	31/12/2004	31/12/2008	SIRI		
287	SNDK	SanDisk Corp.	31/12/2003	31/12/2008		SNDK	
288	SNDT	SunGard Data Systems, Inc.	31/12/1996	30/06/1997			SNDT
289	SNPS	Synopsys Inc.	31/12/1996	31/12/2000	SNPS		
290	SPLS	Staples Inc.	31/03/1995	31/12/2015	SPLS		
291	SPOT	PanAmSat Corp.	31/12/1997	30/09/2004			SPOT
292	SRCL	Stericycle, Inc.	31/12/2007	30/09/2017	SRCL		
293	SSCC	Smurfit-Stone Container Corp.	30/09/1997	31/12/2005			SSCC
294	SSSS	Stewart & Stevenson Services, Inc.	31/03/1995	31/12/1996			SSSS
295	STEI	Stewart Enterprises, Inc.	30/09/1997	31/12/1999	STEI		
296	STJM	St. Jude Medical, Inc.	31/03/1995	31/12/1996			STJM
297	STLD	Steel Dynamics	31/12/2007	31/12/2009	STLD		
298	STRM	StrataCom, Inc.	31/03/1995	30/09/1996	STRM		
299	STRY	Stryker Corp.	31/03/1995	30/09/1997			STRY
300	STRZA	Starz Inc.	31/01/2013	31/03/2013	STRZA		
301	STX	Seagate Technology Holdings	30/11/2008	30/09/2017	STX		
302	SUNW	Sun Microsystems	31/03/1995	31/07/2009	SUNW		
303	SWKS	Skyworks Solutions	31/08/2015	30/09/2017	SWKS		
304	SYBS	Sybase, Inc.	31/03/1995	31/12/1998			SYBS
305	SYMC	Symantec Corp.	31/12/2001	30/09/2017	SYMC		
306	TCOMA	Tele-Communications, Inc.	31/03/1995	31/03/1999			TCOMA
307	TECD	Tech Data Corp.	31/03/1998	31/12/1999	TECD		
308	TECUA	Tecumseh Products Company Class A	31/03/1995	31/12/1995		TECUA	
309	TEVA	Teva Pharmaceutical Industries	31/12/2002	31/05/2012	TEVA		
310	TLAB	Tellabs, Inc.	31/03/1995	31/05/2008	TLAB		
311	TMUS	T-Mobile US Inc	31/12/2015	30/09/2017	TMUS		
312	TRIP	TripAdvisor	31/12/2013	30/09/2017	TRIP		
313	TSCO	Tractor Supply Company	31/12/2013	30/09/2017	TSCO		
314	TSLA	Tesla Motors, Inc.	31/07/2013	30/09/2017	TSLA		
315	TW	Time Warner Inc (Old)	31/12/1995	30/09/1996		TW	
316	TXN	Texas Instruments, Inc.	30/04/2012	30/09/2017	TXN		
317	TYSNA	Tyson Foods, Inc. Class A	31/03/1995	31/12/1997			TYSNA
318	UAUA	UAL Corp.	31/03/2007	31/07/2008			UAUA
319	ULTA	Ulta Salon, Cosmetics & Fragrance Inc	31/12/2015	30/09/2017	ULTA		
320	URBN	Urban Outfitters	31/12/2005	31/12/2006	URBN		
321	USHC	U.S. Healthcare, Inc.	31/03/1995	30/09/1996			USHC
322	USRX	U.S. Robotics Corp.	30/09/1995	30/06/1997			USRX
323	VCELA	Vanguard Cellular Systems, Inc.	31/03/1995	31/12/1996			VCELA
324	VIAB	Viacom Inc.	31/05/2012	30/09/2017	VIAB		

325	VIP	VimpelCom Ltd.	31/10/2013	31/12/2015			VIP
326	VISX	VISX, Inc.	30/06/1999	30/09/2000			VISX
327	VKNG	Viking Office Products, Inc.	31/03/1995	30/09/1998			VKNG
328	VMED	Virgin Media	31/12/2004	31/12/2008	VMED		
329	VOD	Vodafone Group plc	31/12/2009	30/09/2017	VOD		
330	VRSK	Verisk Analytics	31/12/2012	30/09/2017	VRSK		
331	VRSN	VeriSign	30/06/2000	31/12/2012	VRSN		
332	VRTS	VERITAS Software Corp.	31/12/1998	31/07/2005	VRTS		
333	VRTX	Vertex Pharmaceuticals	31/12/2006	30/09/2017	VRTX		
334	VSTR	VoiceStream Wireless Corp.	31/12/1999	30/06/2001	VSTR		
335	VTSS	Vitesse Semiconductor Corp.	31/12/1998	31/12/2002		VTSS	
336	WBA	Walgreens Boots Alliance	31/03/2015	30/09/2017	WBA		
337	WCLX	Wisconsin Central Transportation	31/12/1996	31/12/1998			WCLX
338	WCOEQ	WorldCom, Inc.	31/03/1995	30/09/2002			WCOEQ
339	WCRX	Warner Chilcott	31/12/2008	31/12/2012			WCRX
340	WDC	Western Digital	31/12/2012	30/09/2017	WDC		
341	WFM	Whole Foods Market, Inc.	31/12/2002	31/12/2008		WFM	
342	WMTT	Willamette Industries, Inc.	31/03/1995	31/12/1996			WMTT
343	WTHG	Worthington Industries, Inc.	31/03/1995	31/12/1999			WTHG
344	WYNN	Wynn Resorts Ltd.	31/12/2004	31/12/2015	WYNN		
345	XLNX	Xilinx, Inc.	31/03/1995	30/09/2017	XLNX		
346	XMSR	XM Satellite Radio Holdings Inc.	31/12/2004	31/12/2007			XMSR
347	XOXO	XO Communications, Inc.	31/12/1999	31/12/2001	XOXO		
348	XRAY	DENTSPLY International Inc.	31/12/2002	31/12/2013	XRAY		
349	YELL	Yellow Corp.	31/03/1995	31/12/1995			YELL
350	YHOO	Yahoo! Inc.	31/03/1998	30/09/1998		YHOO	
	Count				189	22	139

Appendix D

PGMPY Implementation

Implementation begins with a first simple Naive Bayesian network exploring the network presented in Figure D.1, beginning with an implementation of the Parameter Estimation class using pgmpy which treats data as discrete.

D.1 Results

Figure D.2 presents a web page showing the forecast returns and variance of a simple Bayesian network implemented using pgmpy. The web page is created using the Django framework (see Section 3.3.4). It presents the current projected result and variance for period $N+1$ as well as the previous projected result at end of period $N-1$ for period N and the actual result of period N. The performance of the algorithm can then be determined based

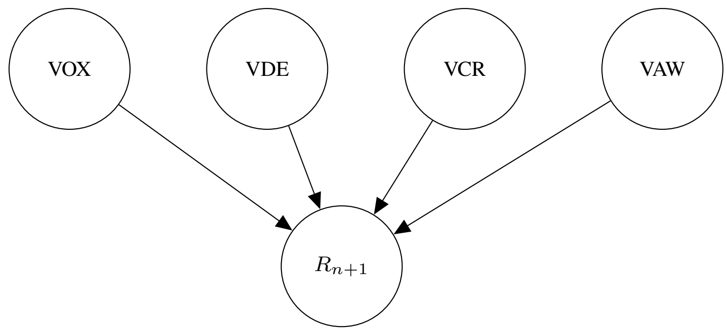


Figure D.1: Naive Bayesian network exploring ETF's as economic predictors. ETF's are described in Table D.1

ETF	Sector
VOX	Communication
VDE	Energy
VCR	Consumer Discretionary
VAW	Materials

Table D.1: Economic sectors of Vanguard ETF's.

upon a comparison of the predicted versus realized results. Figure D.2 also presents the Sharpe ratio (Sharpe, 1994), which is a measure of performance associated with risk-taking. It is calculated as $SharpeRatio = E[R_a - R_f]/\sigma_a$ where R_a is the asset return, R_f is the risk-free rate[1], and σ_a is the standard deviation of the asset excess return.

	Name	1 Mean	2 Variance	3 Sharpe	4 Previous Mean	5 Actual Return	6 Previous Variance
AAL	American Airlines Group	0.1667	1.3056	0.1459	0.1667	-0.0097	1.3056
AAPL	Apple Inc.	0.2500	1.3542	0.2148	0.2500	0.0200	1.3542
ADBE	Adobe Systems Inc.	0.1667	1.3056	0.1459	0.1667	-0.0128	1.3056
ADI	Analog Devices	0.3333	1.5556	0.2673	0.3333	0.0034	1.5556
ADP	ADP, Inc.	0.1667	1.3056	0.1459	0.1667	0.0093	1.3056
ADSK	Autodesk, Inc.	0.3333	1.5556	0.2673	0.3333	-0.0103	1.5556
ADTN	ADTRAN, Inc.	0.3333	1.5556	0.2673	0.3333	-0.0061	1.5556
AKAM	Akamai Technologies, Inc.	0.3333	1.5556	0.2673	0.3333	0.0007	1.5556
ALEX	Alexander & Baldwin, Inc.	0.0000	1.2000	0.0000	0.0000	0.0067	1.2000
ALXN	Alexion Pharmaceuticals	0.3333	1.5556	0.2673	0.3333	-0.0109	1.5556
AMAT	Applied Materials, Inc.	0.3333	1.5556	0.2673	0.3333	-0.0772	1.5556
AMGN	Amgen Inc.	0.2500	1.3542	0.2148	0.2500	0.0050	1.3542
AMZN	Amazon.com, Inc.	0.3333	1.5556	0.2673	0.3333	-0.0023	1.5556
ASND	Ascend Communications, Inc.	0.0000	1.2000	0.0000	0.0000	0.0352	1.2000
ATVI	Activision Blizzard	0.3333	1.5556	0.2673	0.3333	-0.0077	1.5556
AVGO	Broadcom Limited	0.1818	1.4215	0.1525	0.1818	0.0028	1.4215
BIDU	Baidu.com, Inc.	0.3333	1.5556	0.2673	0.3333	0.0247	1.5556
BIIB	Biogen, Inc.	0.3333	1.5556	0.2673	0.3333	0.0067	1.5556

Figure D.2: Equity returns and variance forecast from simple Bayesian network

D.2 Discussion

The prototypes developed using PGMPY led to the conclusion that other solutions such as PyMC3 were better suited to the needs of this research for two reasons. First, this thesis required the ability to use custom functional forms for the relationship between factors and returns. PGMPY was not flexible in this regard. Second, inference was far slower than PyMC3 . For these reasons, PGMPY was rejected.

[1] See Section 3.2.4

Appendix E

Impact of Increased Sample Size in Tuning EGARCH Models

This appendix explores the impact of increasing the sample size for tuning. The number of samples used for tuning improves inference by helping ensure that search over the parameter sample space is efficient. However it increases the time required for training.

Predictions for the monthly returns of Apple shares (ticker AAPL) were performed from 30.06.1999 to 31.05.2005 using 1,000, 6,000, and 50,000 samples for tuning. In addition, one prediction was made using 500,000 samples for tuning. In all cases, the samples used for tuning were discarded and the model was used to generate four chains of 6,000 useful samples.

The times required for generating one prediction for one time point increased from between 15 and 75 minutes when using 1,000 samples for tuning to between 12 and 18 hours when using 500,000 samples.

Table E.1 presents one prediction using the four different sampling sizes. All predictions are very close to each other, considering the standard deviation of

As can be seen in Figure E.1, predictions are almost always very similar, whatever the number of samples used for tuning.

Size	Prediction	Standard Deviation
500000	-0.01919	0.03879
50000	-0.01789	0.03869
6000	-0.01899	0.03787
1000	-0.02073	0.03770

Table E.1: Predicted monthly return of AAPL stock on 30.06.1999 for different sampling sizes.

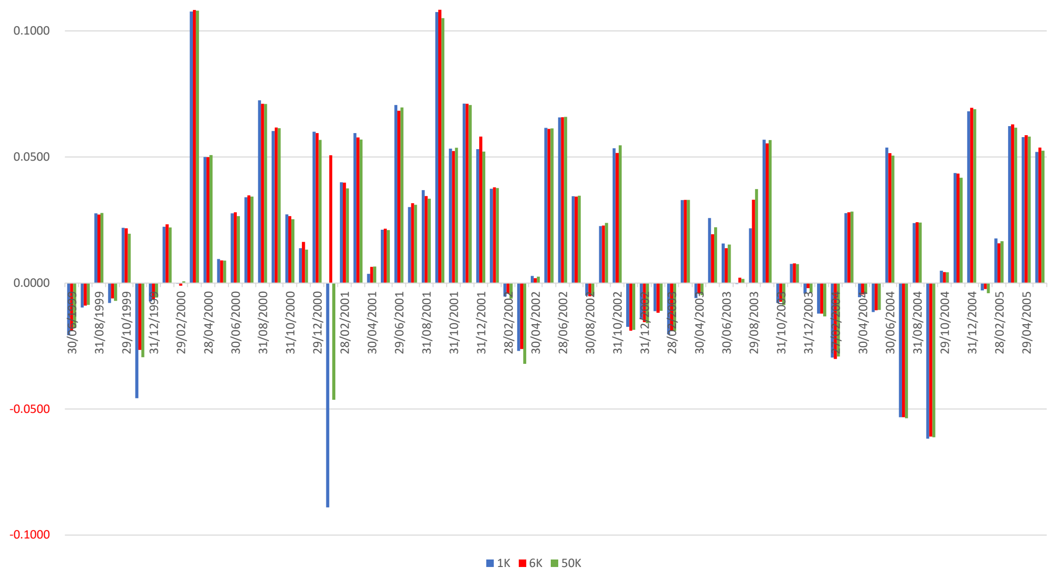
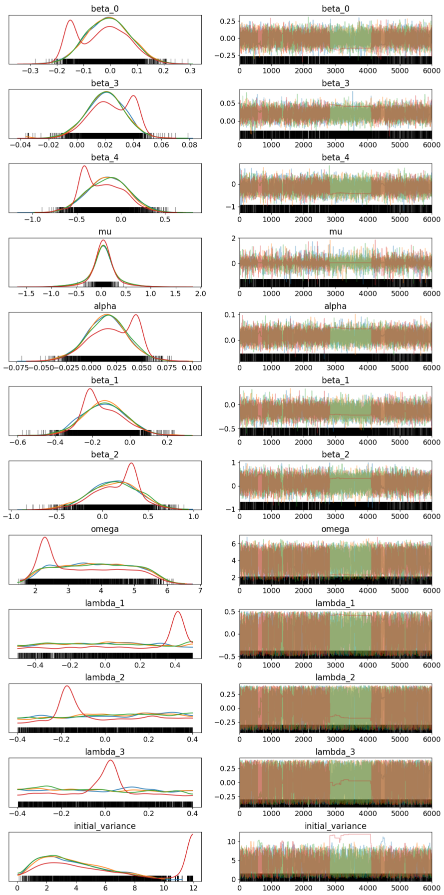


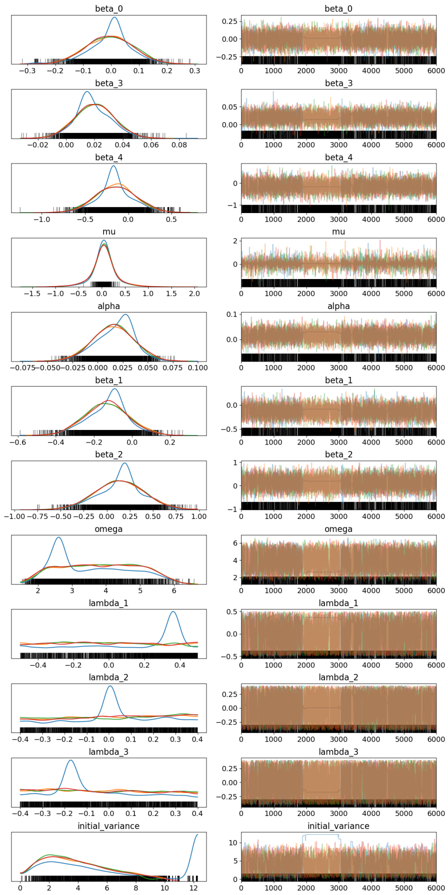
Figure E.1: Comparison of predictions using 1K, 6K, and 50K samples for tuning.

As can be seen in the right parts of the traceplots in Figure E.2, increasing the number of samples used for tuning does improve the search over the parameter space. The chains also appear to converge better with increasing sample sizes, as can be seen in the left part of each traceplot.

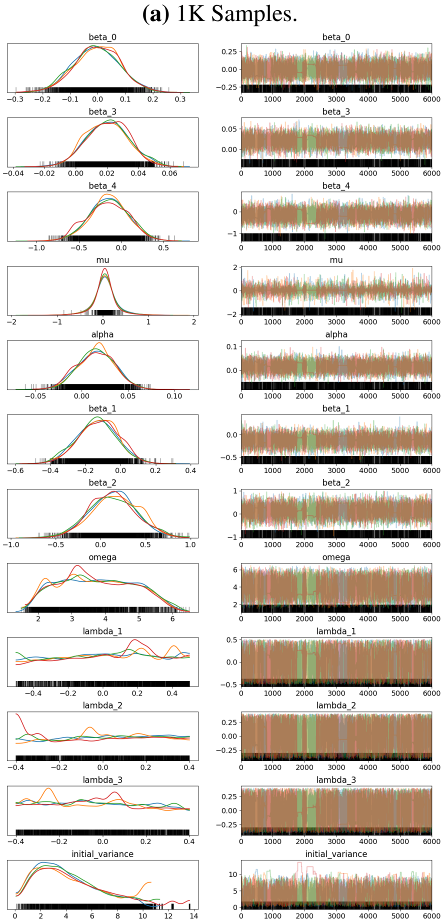
The cost of an improved search over the parameter space afforded by increased sample sizes for tuning is an increase in training time which may be prohibitive if training 10 equities for 200 dates.



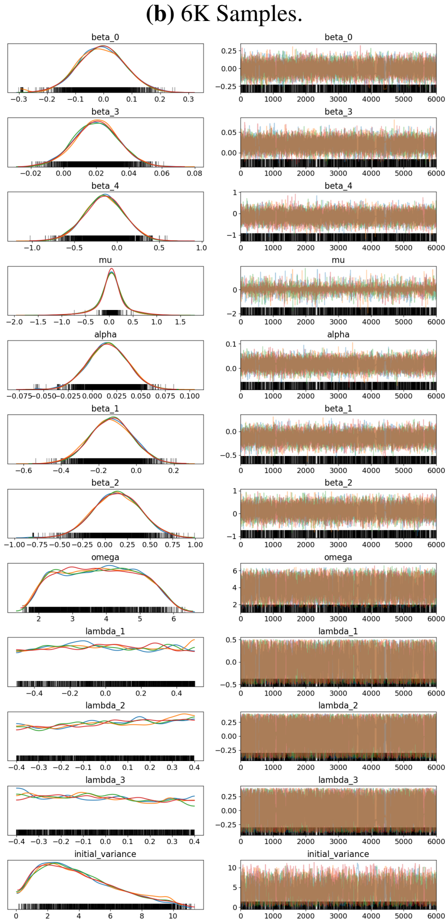
(a) 1K Samples.



(b) 6K Samples.



(c) 50K Samples.



(d) 500K Samples.

Figure E.2: Traceplots using different sample sizes for tuning.

Bibliography

Ahsanullah, M., M. Shakil, and B. M. Golam Kibria (Dec. 2015). "Characterizations of folded student's t distribution". In: *Journal of Statistical Distributions and Applications* 2.1, p. 15.

Almgren, Robert and Neil Chriss (2006). "Optimal portfolios from ordering information". In: *The Journal of Risk* 9.1, pp. 1–47.

Andersson, Magnus, Antonello D ' Agostino, Gabe J De Bondt, and Moreno Roma (2011). "The Predictive Content of Sectoral Stock Prices A US-EURO Area Comparison". In: *European Central Bank*.

Andrew, Alex M. (2004). *Information Theory, Inference, and Learning Algorithms*.

Anyfantaki, Sofia and Antonis Demos (2012). "Estimation and Properties of a Time-Varying EGARCH(1,1) in Mean Model". In: *Econometric Reviews* 35.2, pp. 293–310.

– (2016). "Estimation and Properties of a Time-Varying EGARCH(1,1) in Mean Model". In: *Econometric Reviews* 35.2, pp. 293–310.

Avramov, Doron and Guofu Zhou (Dec. 2010). "Bayesian Portfolio Analysis". In: *Annual Review of Financial Economics* 2.1, pp. 25–47.

Barber, David (2016). *Bayesian Reasoning and Machine Learning*. Cambridge University Press.

Barfuss, Wolfram, Guido Previde Massara, Tiziana di Matteo, and Tomaso Aste (2016). "Parsimonious modeling with Information Filtering Networks". In: *Physical Review* 94.6.

Bauwens, Luc and Michel Lubrano (1998). "Bayesian inference on GARCH models using the Gibbs sampler". In: *The Econometrics Journal* 1.1, pp. 23–46.

Beach, Steven L. and Alexei G. Orlov (2007). "An application of the Black–Litterman model with EGARCH-M-derived views for international portfolio management". In: *Financial Markets and Portfolio Management* 21.2, pp. 147–166.

Bergstra, James, Olivier Breuleux, Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, Guillaume Desjardins, Joseph Turian, David Warde-Farley, and Yoshua Bengio (2010). *Theano: A CPU and GPU Math Compiler in Python*. Tech. rep., p. 1.

Best, M J and R R Grauer (1991). "On the Sensitivity of Mean-Variance-Efficient Portfolios to Changes in Asset Means: Some Analytical and Computational Results". In: *Review of Financial Studies* 4.2, p. 315.

Bevan, Andrew and Kurt Winkelmann (1998). "Using the Black–Litterman Global Asset Allocation Model: Three Years of Practical Experience". In: *Goldman Sachs Fixed Income Research*.

Bingham, Eli et al. (2019). *Pyro: Deep Universal Probabilistic Programming*. Tech. rep., pp. 1–6.

Black, Fischer (1993a). "Beta and Return". In: *The Journal of Portfolio Management* 20.1, pp. 8–18.

– (1993b). "Estimating expected return". In: *Financial Analysts Journal* 49.5, p. 36.

Black, Fischer and Robert Litterman (1992). "Global Portfolio Optimization". In: *Financial Analysts Journal; Sep/Oct* 48.5, pp. 28–43.

Bollerslev, Tim (1986). "Generalized Autoregressive Conditional Heteroskedasticity". In: *Journal of Econometrics* 31, pp. 307–327.

Bonami, P and M. A. Lejeune (2009). "An Exact Solution Approach for Portfolio Optimization Problems Under Stochastic and Integer Constraints". In: *Operations Research* 57.3, pp. 650–670.

Bracegirdle, Christopher Ian (2013). "Inference in Bayesian Time-Series Models". PhD thesis. University College London.

Breeden, Douglas T (1979). "An intertemporal asset pricing model with stochastic consumption and investment opportunities". In: *Journal of Financial Economics* 7, pp. 265–296.

Burkardt, John (2014). "The Truncated Normal Distribution".

Carpenter, Bob et al. (Jan. 2017). "Stan: A Probabilistic Programming Language". In: *Journal of Statistical Software* 76.1, pp. 1–32.

Chandrashekar, Girish and Ferat Sahin (2013). "A survey on feature selection methods". In: *Computers and Electrical Engineering* 40.1, pp. 16–28.

Chen, Wei, Hui Ma, Yiping Yang, and Mengrong Sun (2014). "Application of Artificial Bee Colony Algorithm to Portfolio Adjustment Problem with Transaction Costs". In: *Journal of Applied Mathematics*.

Cont, R (2001). "Empirical properties of asset returns: stylized facts and statistical issues". In: *Quantitative Finance* 1.2, pp. 223–236.

Cooper, Ricky Alyn, Marat Molyboga, and Georgiy Molyboga (2016). "Black-Litterman, Exotic Beta, and Varying Efficient Portfolios: An Integrated Approach". In: *The Journal of Investment Strategies* 6.3, pp. 13–30.

Creamer, Germán G (2015). "Can a corporate network and news sentiment improve portfolio optimization using the Black–Litterman model?." In: *Quantitative Finance* 15.8, pp. 1405–1416.

Cuervo, Edilberto Cepeda, Jorge Alberto Achcar, and Milton Barossi-Filho (2014). "New volatility models under a bayesian perspective: A case study". In: *Economia Aplicada* 18.2, pp. 179–197.

DeMiguel, Victor, Lorenzo Garlappi, and Raman Uppal (2009). "Optimal versus naive diversification: How inefficient is the 1/N portfolio strategy?" In: *Review of Financial Studies* 22.5, pp. 1915–1953.

Denev, Alexander (2011). "Coherent Asset Allocation and Diversification in the Presence of Stress Events". PhD thesis. University of Oxford.

Deng, Guang-Feng, Woo-Tsong Lin, and Chih-Chung Lo (2012). "Markowitz-based portfolio selection with cardinality constraints using improved particle swarm optimization". In: *Expert Systems with Applications* 39.4, pp. 4558–4566.

Di Bernardino, E, J M Fernández-Ponce, F Palacios-Rodriguez, and M R Rodriguez-Griñolo (2015). "On multivariate extensions of the conditional Value-at-Risk measure". In: *Insurance: Mathematics and Economics* 61, pp. 1–16.

Dietz, Laura (2010). "Directed factor graph notation for generative models". In: *Max Planck Institute for Informatics* March, pp. 1–9.

Ding, Shifei, Hong Zhu, Weikuan Jia, and Chunyang Su (2012). "A survey on feature extraction for pattern recognition". In: *Artificial Intelligence Review* 37, pp. 169–180.

Doerner, K F, W J Gutjahr, R F Hartl, C Strauss, and C Stummer (2006). "Pareto ant colony optimization with ILP preprocessing in multiobjective project portfolio selection". In: *European Journal of Operational Research* 171.3, pp. 830–841.

Duane, Simon, A D Kennedy, Brian J Pendleton, and Duncan Roweth (1987). "Hybrid Monte Carlo". In: *Physics Letters B* 195.2.

Engle, Robert F (1982). "Autoregressive Conditional Heteroscedasticity With Estimates Of The Variance Of United Kingdom Inflation". In: *Econometrica* 50.4, pp. 987–1007.

Engle, Robert F and Tim Bollerslev (1986). "Modelling the persistence of conditional variances". In: *Econometric Reviews* 5.1, pp. 1–50.

Engle, Robert F, David M Lilien, and Russell P Robins (1987). "Estimating Time Varying Risk Premia in the Term Structure: The Arch-M Model". In: *Econometrica* 55.2, pp. 391–407.

Fabozzi, Frank J, Sergio M Focardi, and Peter N Kolm (2006). *Financial Modeling of the Equity Market: From CAPM to Cointegration*. Frank J. Fabozzi Series. Wiley.

Fabozzi, Frank J and Petter N Kolm (2007). *Robust Portfolio Optimization and Management*. Ed. by Frank J Fabozzi, Petter N Kolm, Dessislava A Pachamanova, and Sergio M Focardi.

Fama, Eugene F. (1965a). *Random Walks in Stock Market Prices*.

– (1965b). "The Behavior of Stock-Market Prices". In: *Journal of Business* 38.1, pp. 34–105.

– (1970). *Efficient Capital Markets: A Review of Theory and Empirical Work*.

Fama, Eugene F. and Kenneth R. French (June 1992). "The Cross-Section of Expected Stock Returns". In: *The Journal of Finance* 47.2, p. 427.

Fama, Eugene and Robert Litterman (2012). "An Experienced View on Markets and Investing". In: *Financial Analysts Journal* 68.6, pp. 15–19.

Firoozye, Nick and Daniel Blamont (Dec. 2003). "Bayesian Asset Allocation: Black- Litterman". In: *Fixed Income Weekly*, pp. 99–103.

Fong, Kingsley, Craig W Holden, and Charles A Trzcinka (2017). "What Are The Best Liquidity Proxies For Global Research?" In: *Review of Finance* 21, pp. 1335–1401.

Fuhrer, Jeffrey C (2009). "Inflation Persistence". In: *Quarterly Journal of Economics* 09.14.

Gamma, Erich, Richard Helm, Ralph Johnson, and John Vlissides (1995). *Design Patterns Elements of Reusable Object-oriented Software*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc.

Garcia and B ; Gould (1993). "Survivorship bias". In: *Journal of Portfolio Management* 19.

Gelman, Andrew, John B. Carlin, Hal Stern, David B. Dunson, Aki Vehtari, and Donald B Rubin (2014). *Bayesian data analysis*. 3rd ed. Vol. 1. 5. Boca Raton: CRC Press.

Geman, Stuart and Donald Geman (1984). "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-6.6, pp. 721–741.

Giacometti, Rosella, Marida Bertocchi, Svetlozar T Rachev, and Frank J Fabozzi (2007). "Stable distributions in the Black-Litterman approach to asset allocation." In: *Quantitative Finance* 7.4, pp. 423–433.

Golmakani, Hamid Reza and Mehrshad Fazel (2011). "Constrained Portfolio Selection using Particle Swarm Optimization". In: *Expert Systems with Applications* 38.7, pp. 8327–8335.

Grabowski, Roger, Carla Nunes, and James Harrington (2014). "Risk-free Rate". In: *Cost of Capital*. Wiley-Blackwell. Chap. 7, pp. 91–109.

Harris, Richard D F, Evarist Stoja, and Linzhi Tan (2017). "The dynamic Black–Litterman approach to asset allocation". In: *European Journal of Operational Research* 259.3, pp. 1085–1096.

Harvey, Campbell R et al. (2014). ". . . and the Cross-Section of Expected Returns". In: *NBER Working Paper Series, Oct 2014*, p. 20592.

He, Guangliang and Robert Litterman (1999). "The Intuition Behind Black-Litterman Model Portfolios". In: *SSRN Electronic Journal*.

Hoffman, Matthew D and Andrew Gelman (2014). "The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo". In: *Journal of Machine Learning Research* 15, pp. 1593–1623.

Hogg, Robert V. and Allen T. Craig (1978). *Introduction to mathematical statistics*. Macmillan, p. 438.

Huang, Dashan, Shushang Zhu, Frank J Fabozzi, and Masao Fukushima (2010). "Portfolio selection under distributional uncertainty: A relative robust CVaR approach". In: *European Journal of Operational Research* 203.1, pp. 185–194.

Huberman, Gur, Shmuel Kandel, and Robert F Stambaugh (1987). "Mimicking Portfolios and Exact Arbitrage Pricing". In: *The Journal of Finance* 42.1, pp. 1–9.

Idzorek, Thomas M (2007). "A Step-by-Step Guide to the Black-Litterman Model Incorporating user-specified confidence levels". In: *Forecasting Expected Returns in the Financial Markets*, pp. 17–38.

Jaynes, E.T. (1968). "Prior Probabilities". In: *IEEE TRANSACTIONS ON SYSTEMS SCIENCE AND CYBERNETICS* 4.3, pp. 227–241.

Jensen, David (Jan. 2000). "Data Snooping, Dredging and Fishing: The Dark Side of Data Mining a SIGKDD99 Panel Report". In: *SIGKDD Explor. Newsl.* 1.2, pp. 52–54.

Jolliffe, I T (2002). *Principal Component Analysis, Second Edition*. Vol. 30. 3, p. 487.

Jorion, Philippe (1991). "Bayesian and CAPM estimators of the means: Implications for portfolio selection". In: *Journal of Banking & Finance* 15.3, pp. 717–727.

Kita, Eisuke, Yi Zuo, Masaaki Harada, and Takao Mizuno (2012). "Application of Bayesian Network to stock price prediction". In: *Artificial Intelligence Research*.

Koller, Daphne and Nir Friedman (2009). *Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning*. The MIT Press.

Kolm, Petter N, Reha Tütüncü, and Frank J Fabozzi (2014). "60 Years of portfolio optimization: Practical challenges and current trends". In: *European Journal of Operational Research* 234.2, pp. 356–371.

Kolm, Petter and Gordon Ritter (2017). "On the Bayesian interpretation of Black-Litterman". In: *European Journal of Operational Research* 258.2, pp. 564–572.

Kwiatkowski, Jan and Riccardo Rebonato (2011). "A Coherent Aggregation Framework for Stress Testing and Scenario Analysis". In: *Applied Mathematical Finance* 18.2, pp. 139–154.

Ladyman, James, James Lambert, and Karoline Wiesner (2013). "What is a complex system?" In: *European Journal for Philosophy of Science* 3.

Lamont, Owen A (2001). "Economic tracking portfolios". In: *Journal of Econometrics* 105, pp. 161–184.

Lauritzen, Steffen L (1996). *Graphical Models*. Oxford University Press.

LeBaron, Blake (2006). "Agent-based Computational Finance". In: ed. by L Tesfatsion and K L Judd. Vol. 2. Handbook of Computational Economics. Elsevier, pp. 1187–1233.

Lejeune, Miguel A (2011). "A VaR Black–Litterman model for the construction of absolute return fund-of-funds." In: *Quantitative Finance* 11.10, pp. 1489–1501.

Leone, F C, L S Nelson, and R B Nottingham (1961). "The Folded Normal Distribution". In: *Technometrics* 3.4, pp. 543–550.

Lintner, John (1965). "The Valuation of Risk Assets and the Selection of Risky Investments in Stock Portfolios and Capital Budgets". In: *The Review of Economics and Statistics* 47.1, pp. 13–37.

Lo, Andrew W (2002). "The Statistics of Sharpe Ratios". In: *Financial Analysts Journal* 58.4, pp. 36–52.

Malkiel, Burton G (1995). “Returns from Investing in Equity”. In: *The Journal of Finance* L.2.

Mandelbrot, Benoit (1963). “The Variation of Certain Speculative Prices”. In: *The Journal of Business* 36.4, pp. 394–419.

Mankert, Charlotta (2006). *The Black-Litterman Model - mathematical and behavioral finance approaches towards its use in practice*. PhD thesis.

Markowitz, Harry (1952). “Portfolio Selection”. In: *The Journal of Finance* 7.1, pp. 77–91.

– (1955). “Portfolio Selection”. PhD thesis. University of Chicago.

Melas, Dimitris, Raghu Suryanarayanan, and Stefano Cavaglia (2010). “Efficient Replication of Factor Returns: Theory and Applications”. English. In: *Journal of Portfolio Management* 36.2, pp. 39–51.

Metaxiotis, K and K Liagkouras (2012). “Multiobjective Evolutionary Algorithms for Portfolio Management: A comprehensive literature review”. In: *Expert Systems with Applications* 39.14, pp. 11685–11698.

Metropolis, Nicholas, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller, and Edward Teller (1953). “Equation of State Calculations by Fast Computing Machines”. In: *Studies in Molecular Dynamics. I. General Method The Journal of Chemical Physics* 21.6, p. 511.

Meucci, Attilio (2006a). “Beyond Black-Litterman in Practice: A Five-Step Recipe to Input Views on Non-Normal Markets”. In: *Risk*.

– (2006b). “Beyond Black-Litterman: Views on Non-Normal Markets”. In: *Risk*.

– (2008). “Fully Flexible Views: Theory and Practice”. In: *Risk* 21.10, pp. 97–102.

– (2009). “Enhancing the Black-Litterman and related approaches: Views and stress-test on risk factors”. In: *Journal of Asset Management* 10.2, pp. 89–96.

Michaud, Richard O (1989). “The Markowitz Optimization Enigma: Is ’Optimized’ Optimal ?.” In: *Financial Analysts Journal* 45.1, p. 31.

Modigliani, Franco and Modigliani Leah (1997). “Risk-adjusted performance”. In: *Journal of Portfolio Management* 23.2, p. 45.

Neal, Radford M. (2011). “MCMC using hamiltonian dynamics”. In: *Handbook of Markov Chain Monte Carlo*, pp. 113–162.

Nelson, Daniel B (1991). “Conditional Heteroskedasticity in Asset Returns: A New Approach”. In: *Econometrica* 59.2, pp. 347–370.

Oikonomou-Filandras, Panagiotis Agis (2016). “Probabilistic Graphical Models for Mobile Pedestrian Localization in 3D Environments”. PhD thesis. University College London.

Park, Sung Y. and Anil K. Bera (June 2009). “Maximum entropy autoregressive conditional heteroskedasticity model”. In: *Journal of Econometrics* 150.2, pp. 219–230.

Qian, Edward and Stephen Gorman (2001). “Conditional Distribution in Portfolio Theory”. In: *Financial Analysts Journal* 57.2, pp. 44–51.

Rebonato, R and A Denev (2014). *Portfolio Management Under Stress: A Bayesian-Net Approach to Coherent Asset Allocation*. Cambridge University Press.

Rebonato, Riccardo and Alexander Denev (2012). “Coherent Asset Allocation and Diversification in the Presence of Stress Events”. In: *Journal Of Investment Management* 10.4, pp. 19–53.

Rhoades, Stephen A (1993). “The Herfindahl-Hirschman index”. In: *Federal Reserve Bulletin* 79.

Rich, E and K Knight (2009). *Artificial Intelligence*. Third edit. Artificial Intelligence. McGraw-Hill.

Rosenberg, Barr, Kenneth Reid, and Ronald Lanstein (1985). “Persuasive evidence of market inefficiency”. In: *The Journal of Portfolio Management* 11.3, pp. 9–16.

Russell, Stuart J and Peter Norvig (2010). *Artificial Intelligence : a Modern Approach*. 3rd ed., i. Upper Saddle River, N.J. ; London: Pearson.

Samek, Wojciech, Grégoire Montavon, Andrea Vedaldi, Lars Kai Hansen, and Klaus-Robert Muller (2019). *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*.

Satchell, S and A Scowcroft (2000). “A demystification of the Black–Litterman model: Managing quantitative and traditional portfolio construction”. In: *Journal of Asset Management* 1.2, pp. 138–150.

Sefiane, Slimane and Mohamed Benbouzian (2013). “A Meta-Heuristic Ant Colony Optimization Method for Solving Portfolio Optimization”. In: *International Journal of Financial Management* 3.4, pp. 1–8.

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.

– (1966). “Mutual Fund Performance”. In: *The Journal of Business* 39.1, pp. 119–138.

– (1994). “The Sharpe Ratio”. In: *The Journal of Portfolio Management* 21.1, pp. 49–58.

Simard, Richard and Pierre L’Ecuyer (2011). “Computing the Two-Sided Kolmogorov-Smirnov Distribution”. In: *Journal of Statistical Software* 39.11, pp. 1–18.

Stinstra, Erwin and Dick den Hertog (2008). “Robust optimization using computer experiments”. In: *European Journal of Operational Research* 191.3, pp. 816–837.

Takaishi, Tetsuya (2006). “Bayesian estimation of GARCH model by hybrid Monte Carlo”. In: *Proceedings of the 9th Joint International Conference on Information Sciences (JCIS-06)*, pp. 661–664.

– (2013). “Markov chain Monte Carlo versus importance Sampling in Bayesian inference of the GARCH model”. In: *Procedia Computer Science*.

Tran, Dustin, Alp Kucukelbir, Adji B Dieng, Maja Rudolph, Dawen Liang, and David M Blei (2016). *Edward: A library for probabilistic modeling, inference, and criticism*. Tech. rep.

Tütüncü, R H and M Koenig (2004). “Robust Asset Allocation”. In: *Annals of Operations Research* 132.1, pp. 157–187.

Vrontos, I. D., P. Dellaportas, and D. N. Politis (2000). "Full bayesian inference for GARCH and EGARCH models". In: *Journal of Business & Economic Statistics* 18.2, pp. 187–198.

Walters, Jay (2008). *The Black-Litterman Model: A Detailed Exploration*. Tech. rep.

Yuhn, Ky-Hyang, Sang Bong Kim, and James Ross Mccown (2018). "Stock returns, velocity dynamics and inflation volatility". In: *The European Journal of Finance* 24.18, pp. 1755–1771.

Zhang, Xi-li, Wei-guo Zhang, Wei-jun Xu, and Wei-lin Xiao (2010). "Possibilistic approaches to portfolio selection problem with general transaction costs and a CLPSO algorithm". In: *Computational economics* 36.3, pp. 191–200.

Zhou, Guofu (2009). "Beyond Black-Litterman: Letting the Data Speak". In: *Journal of Portfolio Management* 36.1, pp. 36–45.

Zhu, Hanhong, Yi Wang, Kesheng Wang, and Yun Chen (2011). "Particle Swarm Optimization (PSO) for the constrained portfolio optimization problem". In: *Expert Systems with Applications* 38.8, pp. 10161–10169.