Topics in Bayesian machine learning for finance

Trent Spears

Kellogg College

A thesis submitted for the degree of
Doctor of Philosophy

Supervised by
Assoc. Prof. Stefan Zohren and Prof. Stephen Roberts

Department of Engineering Science
University of Oxford

Hilary Term, 2024

To Hadley and Rorik

Declaration

I declare that this thesis is entirely my own work and, except where stated, describes my own research.

Trent Spears
Kellogg College

This document was created using the LaTeX typesetting system. The underlying LaTeX file structure for this thesis was adapted from the open-source Github template provided in [1].

Acknowledgements

Thank you to my supervisors. I would like to extend my deepest gratitude to Prof Stephen Roberts. I appreciate you sharing your energy, unbounded positivity, and immense intellect for the duration of my study period. I am also grateful for the supervision of Dr Stefan Zohren, and the guidance, lively discussion, trust, and encouragement that came with it.

Thank you, Anthony Ledford and the team at the Oxford-Man Institute (OMI), who steward such a wonderful hub for student research. The department provided a supportive and inspiring environment to pursue quantitative finance research. The financial support provided by the OMI is also gratefully acknowledged.

I would like to acknowledge members of the OMI including academics Dr Jan-Peter Calliess, Prof Nir Vulkan and Dr Jeremy Large. Your advice and support had a direct and meaningful impact on my development as a junior researcher. Thank you also to the students within the OMI, especially those in my intake from neighbouring pods including Daniel Poh, Sam Kessler, Max Ahrens and Yin-Cong Zhi, who each contributed to making the research environment so pleasant.

At home, I have been fortunate to have the ultimate companions and support, and I gratefully acknowledge Ingrid, Hadley and, more recently, Rorik. I am also thankful for my other immediate family Jacqueline, Carol, Scott, Brooke, and the late Graham, for many years of love, direction, and belief in my choices.

Finally, thank you to my teachers and mentors from the last 20 years. I have been so incredibly lucky to have you. Your efforts were critical in shaping my education and desire to understand the world. I look forward to paying it on.

Abstract

This thesis presents novel Bayesian machine learning-based approaches to selected problems in finance. We investigate and model perceived market inefficiencies with respect to curated financial data sets, then assess market trading and investment strategies, yielding the following three contributions.

Firstly, we consider a modern deep learning prediction model given intraday limit order book data constituting of a subset of Eurodollar futures contracts, the popular and liquidly traded set of interest rate derivatives. The data is spatio-temporal in nature, such that the model utilises convolutional neural networks for automated feature extraction and recurrent neural networks for time series prediction. We show how to train the network to yield short-term forecasts paired with aleatoric uncertainty estimates by specifying a suitable loss function. Further, we estimate an approximation to epistemic uncertainty via a pseudo-Bayesian deep learning method. This work demonstrates the utility of the model output for deciding the relative allocation of risk capital across trades. That is, investment sizes are scaled between trade opportunities in a principled and data-driven way. We calculate the trading benefit and demonstrate model outperformance relative to strategies that either do not take uncertainty into account, or that utilize an alternative-yet-common market-based statistic as a proxy for uncertainty.

The Black-Litterman model extends the framework of the Markowitz Modern Portfolio Theory to incorporate investor views. In our second work, we recognise this extension as a Bayesian formulation, and relate view uncertainty to its aleatoric or epistemic counterpart. We consider the novel case where multiple view estimates, including uncertainties, are given for the same underlying subset of assets at a point in time. We find that data fusion techniques – for combining similar information from multiple sources – can help to synthesise a collection of views into a single investment strategy. In particular, we present consistency-based methods that yield fused view and uncertainty pairs; such methods are not common to the quantitative finance literature. We show a relevant, modern case of incorporating machine learning model-derived view and uncertainty estimates, and the impact on portfolio allocation, with an example subsuming Arbitrage Pricing Theory. Hence we show the value of the Black-Litterman model in combination with information fusion and artificial intelligence-grounded prediction methods.

Our third contribution further leverages preceding work. We devise a novel Bayes-based conditional factor model for a multivariate portfolio of United States equities in the context of analysing a statistical arbitrage trading strategy. A state space framework underlies the factor model whereby asset returns are assumed to be a noisy observation of a

linear combination of factor values and latent factor risk premia. Filter and state prediction estimates for the risk premia are retrieved in an online way. Such estimates induce filtered asset returns that can be compared to measurement observations, with large deviations representing candidate mean reversion trades. Further, in that the risk premia are modelled as time-varying quantities, non-stationarity in returns is de facto captured. We study an empirical trading strategy respectful of transaction costs and demonstrate performance over a long history of 29 years, for both a linear and a non-linear state space model. Our results show that the model is competitive relative to other methods, including simple benchmarks and other comparable innovative approaches published in the literature. Also of note, while strategy performance degradation is noticed through time – especially for the most recent years – the strategy continues to offer compelling economics, and has scope for further advancement.

List of publications

The following is a list of articles that were published or submitted for review as a result of the research carried out for this thesis.

Publications

- Trent Spears, Stefan Zohren and Stephen Roberts. Investment Sizing with Deep Learning Prediction Uncertainties for High-Frequency Eurodollar Futures Trading. The Journal of Financial Data Science, 3(1): 57-73, 2021.
- Trent Spears, Stefan Zohren and Stephen Roberts. View fusion vis-a-vis a Bayesian interpretation of Black-Litterman for portfolio allocation. The Journal of Financial Data Science, 5(3):23-49, 2023.

Submitted for review

- Trent Spears, Stefan Zohren and Stephen Roberts. On statistical arbitrage under a conditional factor model of equity returns. *https://arxiv.org/abs/2309.02205*, 2023. Submitted for journal review, 12 September 2023.

Table of contents

List of figures

List of tables

Glossary

AI Artificial Intelligence

ANN artificial neural network

APT Arbitrage pricing theory

BbB ‘Bayes by Backprop’

CAPM capital asset pricing model

CART classification and regression tree

CNN convolutional neural network

ELBO evidence lower bound

EMH Efficient Market Hypothesis

FNN feedforward neural network

GMM geometric mean maximization

GP Gaussian process

GPUs graphics processing units

HFT high-frequency trading

i.i.d. independent and identically distributed

LOB limit order book

LSTM long short-term memory network

MPT Modern Portfolio Theory

PCA Principal Component Analysis

RNN recurrent neural network

SQL Structured Query Language

USD United States dollar

WRDS Wharton Research Data Services

Chapter 1

Introduction

1.1 Preliminaries

This thesis concerns novel Bayesian machine learning-based approaches to financial trading and investment. Such approaches amount to quantitative strategies that are necessarily data dependent insofar that economic 'edge' can be identified, tested, and/or modelled, with a subsequent possibility of being earned. In alternative terms, edge represents the knowledge of a logical action one could take that yields an outcome with positive expectancy. We adhere to a stylised workflow for end-to-end strategy development observable – if not explicit – in a large subset of the decades-long history of academic literature and publicly available industry media, and still very much relevant today. This workflow is given by:

(i) Sourcing raw data and constructing a clean data set from it;

(ii) Making use of the cleaned data to investigate and model aspects of a hypothesised market edge; and then

(iii) Implementing a terminal strategy attempting to capture the edge optimising for some preferred statistic(s), such as maximising returns (or risk-adjusted returns).

These steps have been subject to scrutiny, and implemented with due care, in creating the contributions presented within this thesis. Not the least, the interdependence between these three steps is clear; for example, one can hardly estimate a (statistical) model without data, or execute the market-facing (probability distribution-dependent) strategy without an estimated model. Also, each of the three steps require unique technical approaches and nuanced thought processes. Because of this, the consideration and effort devoted to each step was almost equal.

Elaborating on approach, handling data is a critical responsibility, and for a given project a task that often needs to be managed intricately. The workflow item (i) can

be conceptualised in two parts, beginning with sourcing and curating the raw data. In practice, data used for financial applications can take many forms, be it time series data generated by market order books, or data representing visual imagery or text. The term 'big' data has been popularised in recent years, suggestive of curated, managed data sets whose size have become increasingly large with respect to the volume of data that they contain. The management of large data sets often requires dedicated oversight. Also, many data sets are becoming increasingly valuable, evidenced by the growing (real) cost of obtaining them. With respect to useful financial data, avoiding such costs is unlikely, whether the data is obtained via proprietary or otherwise gated repositories, or collected from the public domain. For this work, we make use of two raw data sets collected from well-known financial data providers – Thomson Reuters, and Wharton Research Data Services (WRDS)[1]. Given raw data, a natural next step is to clean – or 'preprocess' – it for its intended use. Preprocessing includes inspecting the data set for erroneous or unexpected values, including missing data or data containing an insidious bug. We support efforts that engage in detailed and early data cleaning to minimise the need to rework the subsequent analysis that is necessarily dependent on that data. Finally, a data set may be expanded upon by engineering and storing additional data features useful for downstream modelling and analysis. Throughout this work, such tasks are supported by the Structured Query Language (SQL) and Python programming language.

Despite the similar importance we ascribe to the workflow items, it is step (ii) above, pertaining to investigating and modelling edge, which commands the greatest amount of attention within this thesis. Ultimately, this work explores modern Bayesian machine learning for finance, and it contains innovations around particular modelling tasks. As George Box stated, "All models are wrong, some are useful" [2]; we certainly find the usefulness of models for gaining insight from well-engineered data and with respect to a perceived edge. To be clear, there is no suggestion that a model itself is *the* edge. Edge can be found in sources such as academic theory, financial market experience and knowledge,

[1] In the spirit of open science we describe how to retrieve and process these data sets, though regretfully they are gated such that we are prohibited from providing free and direct access.

or inspired data exploration and analysis; models often help to sharpen understanding of edge. In any case, present day modelling in cross-domain research continues to evolve alongside the broader, burgeoning theme of Artificial Intelligence (AI). The literature has become saturated with contributions containing machine learning-based modelling. Some of these models were conceptualised in decades past, while others represent bleeding edge innovation. Regardless, researchers refocus on – and further development of – such models has been catalysed by ever more powerful computing resources and big data. The quantitative finance modeller's toolkit has been expanded in large part by deep learning, and innovations to architecture of neural network models. To their advantage, the modeller can conceive of fresh approaches to the enduring problem of estimating predictive or explanatory relationships between variables. Also, particular classes of machine learning models have natural Bayesian or pseudo-Bayesian extensions, yielding frameworks inclusive of a notion of model uncertainty. Such frameworks yield flexible and expressive models; this can benefit terminal strategies for trading and investment that subsume estimates of uncertainty.

Many model-supported edges in quantitative finance entail mappings between data and a vector of expected return estimates for a future time increment, as well as the associated return covariance matrix, for a collection of assets. Given this, one can take the actions to buy or sell these assets with the view to optimise some performance criteria. Deciding exactly how much to buy and sell falls into the remit of the terminal strategy of workflow step (iii). Theoretically optimal strategy can be found in Markowitz's Modern Portfolio Theory (MPT) [3], under whose assumptions one may choose portfolio weights for the target assets on an 'efficient frontier', upon solving a mean-variance optimization problem. Many investors seek a portfolio on the efficient frontier maximising the Sharpe ratio, a performance measure of (excess) return per unit of risk [4]. MPT can be extended to account for practical realities of trading and investing, including transaction costs and the market impact of trade execution – both of which can be a function of portfolio size – as well as adjustments for investor risk preferences. Bayesian modelling extends the

MPT framework, whereby 'epistemic' – or model – uncertainty adjusts the estimates of the return distribution parameter, and hence the optimal portfolio weights. Other strategies include the Kelly criterion [5], which finds the optimal proportion of bankroll to wager to maximise portfolio growth, given a known edge with a known payout. Canonically, Kelly is often described in the context of discrete bets, though it has a continuous extension. We explore the connection between the Kelly criterion and MPT in the next chapter. Thirdly, there are many simplified terminal strategies in the literature, for example, unit sizing long or short based on a directional signal. In that such actions are not theoretically justified, we do not consider them here. There are, however, more ad hoc rules-based sizing strategies for capturing edge, often dependent on strong assumptions about the nature of asset price evolution. One such assumption is the propensity for large error deviations to mean revert; we will visit this in the context of statistical arbitrage trading later in this document. Given a terminal strategy, one can test for capturing an edge on historical data, and/or implement the strategy in production – potentially within a broader portfolio – quantifying performance in terms of various statistics of return and risk. In practice, there are often additional considerations for terminal strategy around *risk management*, though we deem this predominantly beyond the scope of this work[2].

1.1.1 Aside: Edge at (time)scale

A large subset of financial market participants concern themself with the search for edge. Edge can be ephemeral, nebulous and hard found. There are also relatively simple and genuine edges straightforwardly retrievable from the public domain. The existence of edge is despite the Efficient Market Hypothesis (EMH) and its suggestion that market prices are set with respect to all available information at any time [7]. From this, one might infer edge does not exist; we proceed assuming that it does.

Edge – and any associated quantitative strategy workflow – can be conveniently

[2]Those in the investment management industry should see [6] for a thought-provoking resource on this theme.

categorised in a variety of ways. A natural mode of thinking considers edge with respect to the time scale on which (a practitioner, or computer) executes the buying and selling of financial assets relevant to the strategy. Conversely, setting a time scale in which to pursue an edge begets myriad considerations for the tasks of data sourcing, edge modelling, and terminal strategy implementation. Any strategy's scale, or the implied bankroll capacity for pursuing the edge, is an additional and non-trivial consideration.

For example, market actions taken over periods of fractions of a second, or mere seconds, are the domain of high-frequency trading (HFT). An edge in this domain sees market makers compete to settle buy and sell orders brought to market, with an emphasis on speed of order fulfillment and spread capture. Like other reactive intraday trading strategies over periods of seconds and minutes, the trader may achieve remarkable Sharpe ratios (amongst other favourable strategy performance statistics), net of transaction costs. Annualised Sharpe ratios for successful firms have been reported to be in the range of 2-5, and possibly higher (for example, see [8, 9]). Limitations of any such strategies may include their relatively brief period of exploitability (whether edge erosion is a consequence of extreme competition, seasonality, or otherwise), or their relatively low capacity for sizing, within a particular market. An operator in the space might like to consider an ever-wider breadth of markets to trade in search of scale.

Mid-frequency trading, such as strategies executed intraday at the order of minutes and hours, or even interday, may include the well-known momentum trading strategy, mean-reversion trading, and other systematic and opportunistic strategies. These strategies exhibit performance metrics that appear good taken in their own rite, though quoted Sharpe ratios are often significantly worse than for HFT. On the other hand, mid-frequency strategies may be compensated by better trade capacity, and a reduced cost of pursuit – and are still in the keen focus of market participants.

Let us term 'investment' (rather than 'trading') those market actions with the lower – yet still acceptable – net Sharpe ratios, and executed over the longest time frames with portfolio's varying over the timescale of weeks, months, quarters and years. These strategies

are often the target for deployment of the largest pools of capital. Across these categories, and for the broad spectrum of both traditional and modern asset classes, diverse market participants have produced an abundance of theory, experimental results, and practical case studies. In this context, we leverage Ross's Arbitrage pricing theory (APT) [10] and choose an equity factor model as the backdrop to the investment strategy of Chapter 4.

With respect to these preliminaries on quantitative strategy workflow given a perceived edge, we present the objectives of this thesis, and its contributions.

1.2 Objectives

The motivating objective of this thesis was to examine the role of uncertainty estimation within the quantitative strategy workflow introduced above. There were three subobjectives:

- To explore (ii) and (iii) with clear reference to uncertainty estimation. Hence, we could assess how uncertainty estimation contributed to a strategy (if at all).
- To perform experiments with respect to strategies inspired by edges spanning multiple time scales of action.
- To complete the necessary data preparation, per (i), to enable our experimental work, and to clearly present how the processed, clean data set was constructed.

Given these objectives, we were able to make the following contributions to the academic literature.

1.3 Contributions

Our primary contributions are based on the development of research that led to two peer-reviewed, published works, and a third work available in the public domain and submitted for peer review.

In the first work, for a high-to-mid-frequency trading setting, we adapt DeepLOB [11], a deep learning model designed for limit order book (LOB) data, augmented for

modelling spatio-temporal data and pseudo-Bayesian uncertainty estimation. We test the models' predictive capabilities, and aleatoric and epistemic uncertainty estimates, with respect to a trading strategy optimising for portfolio Sharpe ratio. The model is applied to a large collection of United States dollar (USD)-denominated interest rate futures. We find that the model uncovers and exploits lead-lag temporal correlations that are strongest between adjacent contracts within the term structure.

Inspired by field of portfolio optimization, we explore a Bayesian analogue of Markowitz portfolio theory: the well-known Black-Litterman model [12]. We consider the novel case of a manager synthesising multiple investor view estimates – where views consist of expected return and uncertainty assessments – before deciding terminal portfolio allocation. In combining views, we do not assume their independence, but instead call algorithms from the information fusion literature to summarise them, and hence extend the Black-Litterman model for the case of multiple views. We demonstrate the impact of fusion on terminal strategy portfolio metrics, net of transaction costs, for large investors with relatively slow rebalance periods for a large asset pool.

In our final project, noticing a state space representation implicit within the dynamic Black-Litterman conditional factor model framework, we consider a classic mid-frequency trading strategy respective of factor evolution. Indeed, we find a novel approach to statistical arbitrage for a long history of equities data, and demonstrate how high-performing if not simple models are further enhanced when uncertainty is explicitly modelled. Our results are comparable with other well-known published studies, that also serve as meaningful benchmarks; our approach is favourable not only to these works, but also yields commendable performance statistics on an outright basis over a long history of 29 years.

Finally, before any experiments could be completed, it was important to construct two data sets, spanning time scales and asset classes. Throughout this thesis, we present clear and detailed accounts outlining processes for data collection and the construction of finalised, clean data sets. This not only gives clarity with respect to our experimental process, but can also assist interested researchers with reproducibility.

1.4 Outline of thesis

We offer a literature review in Chapter 2, in two parts. The first part reviews the evolution of Bayesian modelling and machine learning, as it relates to understanding the edge within our strategies. We review the concepts of aleatoric and epistemic uncertainty, and estimation (or approximation) thereof. We briefly review classical (Bayesian) time series models, as well as state space models and Gaussian process models for time series. This contrasts with the subsequent sections focussed on deep learning methods. This includes deep neural networks, as well as tree-based ensemble methods, which have both found utility for modelling financial data.

In the second part, we have an explicit focus on portfolio optimization as a framework for our terminal strategy of capturing edge in financial trading and investment. On the one hand, we consider the classic risk-adjusted return framework for portfolio optimization of Markowitz, and a Bayesian extension in the Black-Litterman model. In contrast, and for completeness, we also review the well-known Kelly criterion for optimizing the growth rate of a bankroll assuming the existence of favourable discrete wagers. We also consider an extension for assuming asset returns modelled by a log normal distribution, and compare it with the approach of Markowitz.

Our research depends on two data sets – one for each of the two asset classes that we examined. We describe these data sets in detail throughout, including the sources for the raw data or other data that we accessed, as well as the details of the preprocessing steps including that way the data was (i) cleaned, and (ii) expanded with additional variables upon feature engineering.

In Chapters 3 to 5, we adapt our completed project work into chapters reminiscent of their published (or submitted) manuscripts. This includes, respectively, the aforementioned contributions relating to (i) position sizing with model uncertainties for intraday trading; (ii) view fusion for investing; and (iii) noise filtering for statistical arbitrage trading. We conclude in Chapter 6 with a summary of the thesis.

Chapter 2

Literature review

2.1 Introduction

In this chapter we provide a two-part review of some key ideas from the academic literature that have relevance with regard to modern quantitative finance. The first part pertains to the statistical modelling toolkit that underlies the research contained in later chapters. There is a particular focus on Bayesian model extensions and approaches to uncertainty estimation. The second part revises classical – yet essential – notions of how the prediction and uncertainty estimates gleaned from such models come to have a bearing on trading and investing. Per the prior chapter and with respect to end-to-end strategy development for quantitative finance, the necessity of joint consideration of these two parts anchors the underlying philosophy of this thesis. The chapters subsequent to this brief review contain ideas extending this well-known literature, which provide empirical support to the case for incorporating Bayesian machine learning in finance.

2.2 Models in support of edge

An age-old mantra of financial market traders and investors is ‘buy low, sell high’. This seemingly simple pair of actions is made difficult, or course, in that they are performed sequentially in time. This induces the possibility of a realised profit *or* loss on capital, eventually. In this sense, the notion of ‘edge’ amounts to some minimal useful knowledge about an asset’s value that skews some strategy’s payout (net) positive – at least in expectation. Edge can be found via theory, intuition, and experience, and in present day quantitative finance is often supported by suitable data and scientific modelling frameworks. Edge can manifest in a variety of forms. This manuscript is mostly concerned with cases of

straightforward edge with respect to asset returns[1].

Given an asset price S a canonical model (see, for example, [14]) supposes price changes are Normally distributed on small time increments Δt:

$$\frac{S_{t+\Delta t} - S_t}{S_t} =: r_{t+\Delta t} \sim N(\mu \Delta t, \sigma^2 \Delta t), \tag{2.1}$$

with parameters μ and σ^2 denoting the (annualised) mean and variance, respectively, of returns r_{t+1}. Of course, these parameters are unknown and subject to specification; the parameters can be straightforwardly estimated, for example, by calculating empirical averages given a batch of historical data. There exist many alternative models of asset returns, born of diverse schools of thought. A classic model of expected returns from portfolio theory is given by the Capital Asset Pricing Model (CAPM) [15]:

$$\mathbb{E}r_t = r_{rf,t} + \beta(\mathbb{E}r_{m,t} - r_{rf,t}). \tag{2.2}$$

Here $r_{m,t}$ denotes the market return, and $r_{rf,t}$ the (deterministic) 'risk-free' rate of return. The real-valued parameter β relates the expected asset return, denoted $\mathbb{E}r_t$, in proportion to the expected return of the market, $\mathbb{E}r_{m,t}$; the CAPM relationship says that an asset's expected return depends only on its exposure to undiversifiable (systematic market) risk.

In extension of CAPM, factor models of returns were introduced in Ross's APT [10]. In this model, systematic risk is decomposed into a linear combination of interpretable factors, yielding a more granular expression of asset return drivers:

$$\tilde{r}_t = \boldsymbol{\beta}_f \boldsymbol{r}_{f,t} + \epsilon_f, \tag{2.3}$$

In this relationship, $\tilde{r}_t$ denotes excess returns (that is, returns in excess of the risk-free rate), while the real-valued vectors $\boldsymbol{\beta}_f$ and $\boldsymbol{r}_{f,t}$ contain factor betas (or *exposures*) and factor values, respectively. Further, ϵ_f is an independent and identically distributed (i.i.d.) zero-mean white noise process with some variance σ_f^2. Specifying the factors precisely, especially what they represent and how many factors are used, is an area of decades-long

[1] Evidently, there is a wide range of literature for financial market prediction that does *not* concern returns; in the recent review of [13], note that only about 18% of the 57 articles *do* (per Table 17).

and unsettled research. Regardless, the works of Fama and French [16, 17] and Carhart [18] present fundamental economic factors that continue to endure in both industrial and academic modelling frameworks, such as firm size, value, and return momentum.

A model of returns often analysed from the perspective of researchers in fields such as applied mathematics, statistics and engineering is of the form

$$r_{t+\Delta t} = f(\boldsymbol{x}_t) + \epsilon_n, \tag{2.4}$$

for f a function of a vector of real-valued data $\boldsymbol{x}_t$ known at time t, and an i.i.d. zero-mean white noise process ϵ_n with variance σ_n^2. Here f is some linear or non-linear parametric function, and $\boldsymbol{x}$ can be collected from diverse sources and subsumes both cross-sectional and time series data deemed to be sufficiently predictive of returns. The specification of, and estimation methods for, f has broadened with recent advances in machine learning research. Some of the most transformative methods are introduced in this chapter and used as part of trading or investing strategies as shown in later chapters.

Similarly, a catalyst for the recent innovation with respect to modelling and model estimation has been data development – especially data sets becoming increasingly large in size, and novel in their content. Such data takes various forms. Asset price and return time series data is ubiquitous, though can have high barriers to access. Alternative financial data increasingly includes image data such as satellite imagery of the earth's surface, text data such as news articles or official speech transcripts, and myriad qualitative and quantitative data representing consumer activity. In current times, financial data continues to be collected, stored, and managed as an item of exceedingly high value. Further, data sets may show a low price elasticity of demand [19].

The equations (2.1)-(2.4) are applicable in a variety of trading and investing settings. They supported the investigation of edge that underlies this thesis. With respect to Equation (2.4), we found instances where simpler and more parsimonious specifications of f were a better fit within an end-to-end strategy, though this was not always the case. This suggests that a modeller's toolkit is enriched by a diverse range of modelling methods. Further, the returns model is obviously amenable to various extensions, many that present in later

chapters. The literature contains examples including those extending an asset return to a vector of potentially correlated asset returns, paired with a return covariance; incorporating models for time-varying error (co-)variance; updating the model target to a mean-reverting idiosyncratic error variance of return spreads; and incorporating Bayesian analysis to model parameter uncertainty, which includes hierarchical models for factors. This thesis gives special consideration to the latter theme of Bayesian analysis and its consequences within a broader trading and investing strategy. We review Bayesian frameworks briefly throughout.

2.2.1 Established Bayesian paradigms

A Bayesian framework

The relationship given by Equation (2.4) is inherently probabilistic, and the density function $p(r|\boldsymbol{x})$ is clearly Gaussian. If the density p is known it may be used, along with some query data denoted $\boldsymbol{x}^*$, to make a prediction about the random, unobserved quantity $r|\boldsymbol{x}^*$, which is naturally given by the expected value $\mathbb{E}(r|\boldsymbol{x}^*)$. The associated prediction variance $\text{var}(r|\boldsymbol{x}^*)$ represents a measure of prediction 'uncertainty' that is *aleatoric* in nature. It is an irreducible quantity characteristic of the underlying randomness assumed for $r|\boldsymbol{x}$. Taken as a pair, the prediction and associated uncertainty are critical inputs to the trading strategies used in later chapters.

On the other hand, to progress with model-based inference – still with respect to Equation (2.4) – one typically (i) sets the functional form of *f*, and (ii) assumes that *f* is parameterised by the elements of some estimable parameter vector $\boldsymbol{\theta}_f$. In the case of (i), it is implicitly assumed that the choice of *f* is sufficient for describing the true – but, for our problems of interest, unknowable – relationship f^* mapping $\boldsymbol{x}$ to r. That is, we do not suppose the existence of any *model error* between *f* and f^* induced by a misspecified functional form.

Considering the case of (ii), first define $\boldsymbol{\theta} = (\boldsymbol{\theta}_f, \sigma_n^2)$. The parameter vector $\boldsymbol{\theta}$ has quantities seldom 'known' but that are estimated given suitable data. This can be

achieved by the principle of maximum likelihood: given a set of n observed data points $\mathcal{D} := \{(\boldsymbol{x}_i, r_i) : 1 \leq i \leq n; \boldsymbol{x}_i \times r_i \in \mathbb{R}^d \times \mathbb{R}\}$ an estimate $\hat{\boldsymbol{\theta}}$ is a maximiser of the likelihood function $p(\mathcal{D}|\boldsymbol{\theta})$, with respect to $\boldsymbol{\theta}$. Further, the parameter vector $\boldsymbol{\theta}$ can also be modelled distributionally; then, for a random sample of observed data representative of the true population, beliefs about $\boldsymbol{\theta}$ can be refined. This concept forms a basis for Bayesian inference. Firstly, a prior distribution is assumed that reflects beliefs about the parameters, $p(\boldsymbol{\theta})$. Then by Bayes' rule, the prior is updated via the likelihood yielding the parameter posterior distribution given data: $p(\boldsymbol{\theta}|\mathcal{D}) \propto p(\mathcal{D}|\boldsymbol{\theta})p(\boldsymbol{\theta})$. For a query data point $\boldsymbol{x}^*$, the (posterior) predictive distribution can be written

$$p(r|\boldsymbol{x}^*, \mathcal{D}) = \int p(r|\boldsymbol{x}^*, \boldsymbol{\theta})p(\boldsymbol{\theta}|\mathcal{D})d\boldsymbol{\theta}. \tag{2.5}$$

From this density one can compute the updated point prediction $\mathbb{E}(r|\boldsymbol{x}^*, \mathcal{D})$ and prediction variance $\text{var}(r|\boldsymbol{x}^*, \mathcal{D})$, which integrate probabilistic beliefs about $\boldsymbol{\theta}$ updated for the observed data.

It is convenient that given suitable assumptions about the functional form of the prior, the posterior predictive distribution – and the corresponding expectation and variance – is solvable in closed form. Though this is not always the case. For example, for the class of problem relating to *classification*, which models the probability the target r takes the value of any of a discrete set of class labels, then Equation (2.5) is an intractable integral. In such cases, methods for numerical integral approximation are often necessary [20]. Variational inference is another established statistical technique for estimating intractable distributions; we have added some preliminary details to Appendix A below.

The Bayesian framework implies an additional contribution to model uncertainty induced by the prior and posterior distributions of $\boldsymbol{\theta}$, which we call the *epistemic* uncertainty. While the aleatoric and epistemic uncertainty are not strictly decomposable as components of the total predictive uncertainty, the law of total variance makes for a compelling distinction:

$$\text{var}(r|\boldsymbol{x}^*, \mathcal{D}) = \text{var}_{\boldsymbol{\theta}|\mathcal{D}}\big(\mathbb{E}(r|\boldsymbol{x}^*, \boldsymbol{\theta})\big) + \mathbb{E}_{\boldsymbol{\theta}|\mathcal{D}}\big(\text{var}(r|\boldsymbol{x}^*, \boldsymbol{\theta})\big). \tag{2.6}$$

On the right-hand side of the equation the subscript on the outer variance and expectation terms denote that the integral is calculated with respect to the conditional density for $\boldsymbol{\theta}|\mathcal{D}$. The terms are representative of epistemic and aleatoric uncertainty, respectively, in the Bayesian framework. We will see there are benefits to such a decomposition; for example, for deep learning it forms the basis of a pseudo-Bayesian approximation to estimating uncertainty for a neural network in both regression and classification settings [21, 22]. But first, we recall Gaussian process regression as a paragon of Bayesian modelling.

Gaussian process regression

Gaussian process (GP) regression generalises inference from that concerning finitely many latent random variables as in the Bayesian framework above, to inference that concerns (potentially, infinite dimensional) random processes [23]. This yields an alternative, non-parametric Bayesian methodology for estimating dependence relationships given data. A key introductory reference to GP is given by [24], which influences the following outline (and partially, notation).

Consider a function f mapping from its domain $\mathcal{X}$ to the real numbers $\mathbb{R}$. Then f can be perceived as a stochastic process, that is, a collection of random variables – potentially of infinite size – indexed by each point $\boldsymbol{x} \in \mathcal{X}$. Suppose further that the process f has a GP prior, which we write $f \sim GP(\mu, k)$ for a mean function μ and covariance function k called the kernel. Then by the definition of a GP, for any finite collection on m elements from $\mathcal{X}$ – that we write as $\boldsymbol{X} = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_m\}$ – the corresponding finite set of random variables $f(\boldsymbol{X}) := \{f(\boldsymbol{x}_1), ..., f(\boldsymbol{x}_m)\}$ has a multivariate normal distribution. That is, $f(\boldsymbol{X}) \sim N(\boldsymbol{\mu}(X), K(X, X))$ with mean vector

$$\boldsymbol{\mu}(X) = [\boldsymbol{\mu}(\boldsymbol{x}_1), ..., \boldsymbol{\mu}(\boldsymbol{x}_m)]$$

and covariance matrix defined in terms of the kernel as

$$K(X,X) = \begin{bmatrix} k(\boldsymbol{x}_1,\boldsymbol{x}_1) & ... & k(\boldsymbol{x}_1,\boldsymbol{x}_m) \\ \vdots & \ddots & \vdots \\ k(\boldsymbol{x}_m,\boldsymbol{x}_1) & ... & k(\boldsymbol{x}_m,\boldsymbol{x}_m) \end{bmatrix}.$$

Oftentimes it is assumed that the mean function is everywhere zero, for example, in the case of de-meaned data. The derivations below continue under this assumption. Also, in specifying the kernel, K must be a positive semi-definite matrix.

A GP is suitable for modelling the relationship in Equation (2.4), despite the presence of the additional noise term. Indeed, in assuming the i.i.d. zero-mean white noise nature of the noise process, it can be subsumed into our GP modelling framework by addition to the diagonal of the covariance matrix:

$$K(X,X) \to K(X,X) + \sigma_n^2 I,$$

for I an $m \times m$ identity matrix [25].

Continuing with our returns model, suppose we have the set of m training data[2] $\mathcal{D} := \{(\boldsymbol{x}_i, r_i) : 1 \leq i \leq m; \boldsymbol{x}_i \times r_i \in \mathbb{R}^d \times \mathbb{R}\}$, and write $\boldsymbol{r} = \{r_1, ..., r_m\}$. Then fitting a GP amounts critically to kernel engineering – that is, selecting an appropriate functional form for the kernel before optimizing the free hyperparameters given observed data. The optimisation objective is to minimise the negative log-likelihood

$$-\log p(\boldsymbol{r}|X) \propto \frac{1}{2}\boldsymbol{r}^T(K(X,X) + \sigma_n^2 I)^{-1}\boldsymbol{r} + \frac{1}{2}\log|K(X,X) + \sigma_n^2 I|.$$

At prediction time, given m^* test points $X^* = \{\boldsymbol{x}_1^*, ..., \boldsymbol{x}_{m^*}^*\}$, one yields prediction estimates for the corresponding $\boldsymbol{r}^* = \{r_1^*, ..., r_{m^*}^*\}$ based on a conditioning property of the Gaussian distribution. Indeed, it can be shown that $p(f(X^*)|X^*, \mathcal{D})$ is the density of a Gaussian random variable with mean and covariance given by, respectively,

$$\mu^* = K(X^*,X)[K(X,X) + \sigma_n^2 I]^{-1}\boldsymbol{r}$$
$$\Sigma^* = K(X^*,X^*) - K(X^*,X)[K(X,X) + \sigma_n^2 I]^{-1}K(X^*,X)^T.$$

[2]We drop the time label t, for brevity of notation.

It follows that $\boldsymbol{r}^*|X^*, \mathcal{D}$ is itself Gaussian with mean μ^* and covariance $\Sigma^* + \sigma_n^2 I$.

The chief bottleneck of naively applying GP models is in the necessary inversion of the covariance matrix. Given n data points, and hence an $n \times n$ covariance matrix, one finds $O(n^3)$ computational complexity. The limitation imposed by this requirement is even more challenging in modern 'big data' regimes, where n can be large indeed. There exist approximation-based methods for scaling GPs for big data applications, though this is an ongoing research domain. Recent attempts include those of [26, 27]. On the other hand, for financial applications with relatively smaller data set sizes, GPs are indispensable inclusions to the modeller's toolkit. Further, it should be noted that GPs are suitable for modelling both spatial and time series data [25].

State space modelling

We briefly note an historic class of time series models that, though perceivable as 'classical', nonetheless display utility in Chapters 4 and 5 that follow. State space models relate the evolution of a latent stochastic process indexed by discrete time to noisy process observations. A linear Gaussian state space model is given by the time-varying system of equations:

$$\boldsymbol{x}_{k+1} = \boldsymbol{A}_k \boldsymbol{x}_k + \boldsymbol{\epsilon}_k^{\boldsymbol{x}}, \tag{2.7}$$

$$\boldsymbol{y}_k = \boldsymbol{B}_k \boldsymbol{x}_k + \boldsymbol{\epsilon}_k^{\boldsymbol{y}}. \tag{2.8}$$

All quantities are indexed by time k. Here $\boldsymbol{\epsilon}_k^{\boldsymbol{x}}$ and $\boldsymbol{\epsilon}_k^{\boldsymbol{y}}$ model independent zero-mean white-noise processes with respective covariances $\boldsymbol{\Sigma}_k^x$ and $\boldsymbol{\Sigma}_k^y$, assumed to be known functions of time. The matrices $\boldsymbol{A}_k$ and $\boldsymbol{B}_k$ are also assumed to be known functions of time. We call Equations (2.7) and (2.8) the 'state' and 'measurement' equations, respectively, for the latent random state $\boldsymbol{x}_k$ and noisy measurement $\boldsymbol{y}_k$. Given the system parameters, probabilistic beliefs about the state can be inferred, and these beliefs can be iteratively updated given measurement data by Bayesian inference.

The Kalman filter is a well-known algorithm for efficient, online state estimation given streaming observation data. The algorithm has a long history and has been the focus

of many extensions and improvements. For example, the state space modelling framework can be extended for the inclusion of exogenous data, for modelling parameter uncertainty in the specified terms, and can handle missing data [28, 29]. Per subsequent chapters, the framework can also model non-linearities at the level of both the state and measurement equations and can handle multiple measurements from alternative sources in the same time step. A full account of state space modelling is beyond the scope of this work, and we defer interested readers to the references within this and later chapters of this manuscript.

2.2.2 Deep learning

Artificial neural networks (ANN) are of historic importance for modelling non-linear relationships between data. ANNs were put on a mathematical basis by the physiology-inspired description of an artificial brain neuron by McCulloch and Pitts in the 1940s [30]. Subsequently, Rosenblatt's Perceptron [31] demonstrated a basic neural network for binary classification. The theory of ANNs has seen rapid development since; a good introduction to ANNs starting from the Perceptron can be found in [32]. Of chief importance is the universal approximation theorem that puts the use of ANNs on a strong theoretical basis – we see that single-layer ANNs with a finite number of nodes can approximate a continuous function with compact support arbitrarily well, under certain conditions [33, 34].

Though decades old, the effectiveness of deploying neural networks for solving a variety of problems continues to be demonstrated, and improved. There are many historical reasons for this, but in the last decade the neural network-based field of deep learning has burgeoned. A key driver of the modern resurgence of neural network models is attributed to acceleration in computer hardware development [35]. Indeed, the availability of computing power has allowed for the processing of increasingly large data sets, while concurrently neural network architectures have gained increasingly more layers, parameters and consequently, complexity. So called 'deep' neural network learning is popularly presented in [36], and a thorough recent history of innovation in deep neural networks can be found in [37].

In this subsection, we review some key deep learning model architectures based on feedforward and recurrent neural networks, and some concepts of model estimation. We describe the complexity and practical challenge of estimating theoretically justified deep Bayesian models, and present approximation methods that present compelling substitutes. Deep learning should be of interest to quantitative finance researchers as a collection of supplementary methods in support of data modelling and analysis.

Feedforward neural networks

A feedforward neural network (FNN) can be described in terms of a vector of input data $\boldsymbol{x}$, target output vector $\boldsymbol{r}$, and at least one intermediate – or 'hidden' – layer containing one or more nodes, which taken together parameterise a latent function f. Sometimes this structure is referred to as a 'multi-layer perceptron'.

For a FNN, each of the j nodes of the i^{th} hidden layer are associated a real-valued weight vector $\boldsymbol{w}^{ij}$ and bias term b^{ij}, and calculation $h^{(ij)}$ given by

$$h^{(ij)}(\boldsymbol{x}^{(i-1)}; \boldsymbol{w}^{ij}, b^{ij}) = \sigma^{ij}(\boldsymbol{w}^{ij}\boldsymbol{x}^{(i-1)} + b^{ij}). \tag{2.9}$$

Activation functions $\sigma^{ij}(.)$ are mappings applied to the linear transformation of the node input that (typically) contribute to expressing any non-linear relationship between $\boldsymbol{x}$ and $\boldsymbol{r}$. To be clear: for input data $\boldsymbol{x}^{(i-1)}$ into hidden layer i, the corresponding output data $\boldsymbol{x}^{(i)}$ is a vector with j^{th} element given by the scalar value of Equation (2.9). Of course, $(\boldsymbol{x}^{(0)}, \boldsymbol{x}^{(L)}) = (\boldsymbol{x}, \boldsymbol{r})$ for L the number of network layers. In modern parlance 'deep learning' refers to neural network models where L is at least three, corresponding to two hidden layers – though there is not a strictly defined threshold.

Some FNNs are designed with additional structure – we briefly present two such models, beginning with the autoencoder. An autoencoder is a FNN with multiple hidden layers that can be described as two parts. The first part is an 'encoder' component – a collection of upstream hidden layers within the network architecture with (typically) a decreasing number of hidden layer nodes with layer depth. This is followed by a corresponding 'decoder' – a collection of subsequent hidden layers within the network

architecture with (typically) an increasing number of nodes with layer depth. Often, the input data and the target output are the same. An estimated model can be applied, for example, with respect to the two parts. Firstly, the encoder can be used to reduce the dimensionality of the input data, whereby the network learns a lower-dimensional, compressed representation of the input as preprocessing for downstream tasks. On the other hand, the decoder can be used to reconstruct data of the original dimensionality.

Another FNN dominant in the deep learning literature is the convolutional neural network (CNN) [38]. A CNN has input data with some ordering. For many applications, the input data is spatial in nature, and includes, for example, 2D images represented as numerical data, such as a colour gradient across a pixel grid, and 3D data represented as a collection of 2D representations across multiple 'channels', such as a RGB colour scheme. Regardless, within a CNN a hidden layer amounts to sliding 'filters' – or 'kernels' – of varying size over the input space, with filter weights (and subsequent activation functions) learning summary feature maps across the input. Filters can be stacked, or themselves have channels, to learn multiple latent feature representations, and can potentially learn both fine and coarse features of the input data. Many CNN architectures also include downsampling layers, which can reduce model dimension, and is thought to contribute translation invariance to the model. Of course, adding CNN layers to a model adds to the set of hyperparameters for consideration, such as kernel dimension, stride value, and the particulars of the pooling layers. Choices for these terms can be empirically determined.

There are many examples of the successful modelling of spatial input data using CNNs. For example, in the domain of computer vision, the tasks of model-based image classification and object detection have been vastly improved [39]. FNNs, including CNNs, have also been used for sequence modelling, hinting at use cases for modelling time series [40]. Further, researchers have shown that a combination CNN-LSTM network can be effective for modelling spatio-temporal data [41] – the architecture is designed for sequence prediction problems with spatial inputs. Considering this, we proceed to discuss the long short-term memory network (LSTM), or more generally, the recurrent neural network

(RNN), next.

Recurrent neural networks

Consider a sequence of input data with a discrete index set that we will use to denote time: $\{\boldsymbol{x}_1, ..., \boldsymbol{x}_t\}$. A RNN [42] – an alternative class of neural networks relative to FNNs above – models a target variable with sequence data, connecting historical information, as well as the most recent data, to the current prediction.

To do this, RNNs maintain a rolling summary of the previous hidden state and use it to augment the input data at each time step. Extending the notation used previously to make the time index t explicit, then similar to the case of FNNs, the hidden layer output is calculated as

$$h^{(ij)}(\boldsymbol{x}_t^{(i-1)}, \boldsymbol{x}_{t-1}^{(i)}; \boldsymbol{w}^{ij}, \boldsymbol{w}_l^{ij}, b^{ij}) = \sigma^{ij}(\boldsymbol{w}^{ij}\boldsymbol{x}_t^{(i-1)} + \boldsymbol{w}_l^{ij}\boldsymbol{x}_{t-1}^{(i)} + b^{ij}).$$

The tanh function often serves as the activation function $\sigma^{ij}(.)$ for vanilla RNNs. From the equation above, the hidden layer's output (as a vector of scalar node outputs) feeds back into the hidden layer at the next time increment – along with the next data vector of the given sequence. This has an interpretation of maintaining model 'memory'.

It is important to note that the internal RNN weights are independent of time. This obviously reduces the number of parameters to be learned. Regardless, the standard backpropagation algorithm (reviewed below) does not apply outright for RNN weight estimation. Instead, we depend on 'Backpropagation through time' [43], leading to a now-standard algorithm that adapts backpropagation by, intuitively speaking, 'unrolling' an RNN and calculating accumulated errors across time steps, before updating model weight estimates.

Amongst the limitations of RNN models, two are critical. Firstly, calculated gradients are liable to explode or vanish, causing numerical instability on estimation. Secondly, it is not clear the extent RNNs apply 'memory' to arbitrarily long time series, let alone that layer outputs are sufficiently influenced by observations only a few time steps back. This has led to the development of alternative RNN models, which attempt to remedy these

limitations. Indeed, LSTM models, a type of RNN first outlined in [44], have proven effective for modelling sequences of temporal data. For example, in the field of natural language processing, LSTMs have improved machine-enabled language translation [45]. LSTMs have also been applied to time series forecasting [46]. By design, the LSTM has a hidden layer like that of the RNN in that it takes encoded past information as a second input. However, there are two notable distinctions – the hidden layer computations as below are obviously more intricate; and there exists a second layer output fed back into the hidden layer at the next time step. These differences serve to improve model 'memory', and numerical stability at training time.

The internal LSTM layer structure are slightly more complicated than that of RNNs, containing the following components:

$$
\begin{aligned}
h_{t,u}^{(ij)}(\boldsymbol{x}_t^{(i-1)}, \boldsymbol{x}_{t-1}^{(i)}; \boldsymbol{w}_u^{ij}, \boldsymbol{w}_{u,l}^{ij}, b_u^{ij}) &= \sigma_u^{ij}(\boldsymbol{w}_u^{ij}\boldsymbol{x}_t^{(i-1)} + \boldsymbol{w}_{u,l}^{ij}\boldsymbol{x}_{t-1}^{(i)} + b_u^{ij}),\\
h_{t,o}^{(ij)}(\boldsymbol{x}_t^{(i-1)}, \boldsymbol{x}_{t-1}^{(i)}; \boldsymbol{w}_o^{ij}, \boldsymbol{w}_{o,l}^{ij}, b_o^{ij}) &= \sigma_o^{ij}(\boldsymbol{w}_o^{ij}\boldsymbol{x}_t^{(i-1)} + \boldsymbol{w}_{o,l}^{ij}\boldsymbol{x}_{t-1}^{(i)} + b_o^{ij}),\\
h_{t,f}^{(ij)}(\boldsymbol{x}_t^{(i-1)}, \boldsymbol{x}_{t-1}^{(i)}; \boldsymbol{w}_f^{ij}, \boldsymbol{w}_{f,l}^{ij}, b_f^{ij}) &= \sigma_f^{ij}(\boldsymbol{w}_f^{ij}\boldsymbol{x}_t^{(i-1)} + \boldsymbol{w}_{f,l}^{ij}\boldsymbol{x}_{t-1}^{(i)} + b_f^{ij}),\\
h_{t,g}^{(ij)}(\boldsymbol{x}_t^{(i-1)}, \boldsymbol{x}_{t-1}^{(i)}; \boldsymbol{w}_g^{ij}, \boldsymbol{w}_{g,l}^{ij}, b_g^{ij}) &= \sigma_g^{ij}(\boldsymbol{w}_g^{ij}\boldsymbol{x}_t^{(i-1)} + \boldsymbol{w}_{g,l}^{ij}\boldsymbol{x}_{t-1}^{(i)} + b_g^{ij}).
\end{aligned}
$$

The labels u, o, f and g denote calculations internal to an LSTM 'cell' (with the former three sublabels well-known in the literature for denoting 'input', 'output' and 'forget' terms). Collecting the terms over nodes j into time-dependent vectors u_t, o_t, f_t and g_t (and dropping the layer label i, for brevity), the additional memory term of the LSTM cell output can be compactly written

$$\boldsymbol{x}_t^{(i,2)} = \boldsymbol{x}_{t-1}^{(i,2)} \otimes f_t + g_t \otimes u_t$$

where $\otimes$ denotes the Hadamard product. This term weights the 'memory' of the previous cell state by a 'forgetting' factor then adds the current hidden state update weighted by the input term. Finally, the hidden output is given by

$$\boldsymbol{x}_t^{(i)} = \tanh(\boldsymbol{x}_t^{(i,2)}) \otimes o_t.$$

One can extend this model for temporal output sequences, so that sequence input and/or sequence outputs can be modelled. Of course, in the most recent years yet another class of

sequence-to-sequence models have gained immense popularity – the Transformer network, and the underlying Attention mechanism. We discuss this in the context of future work in later sections.

Concepts for model training

Consider a neural network-based model $\boldsymbol{r} = f(\boldsymbol{x}; \boldsymbol{\theta})$ for f subsuming a hidden layer or composition of hidden layers, such as those described above. Then the p-dimensional parameter vector $\boldsymbol{\theta}$ collects model weights that we might think to estimate, given a data set $\mathcal{D} := \{(\boldsymbol{x}_i, \boldsymbol{r}_i) : 1 \leq i \leq m; \boldsymbol{x}_i \times \boldsymbol{r}_i \in \mathbb{R}^{d_x} \times \mathbb{R}^{d_r}\}$, by solving

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta} \in \mathbb{R}^p}{\arg\min}\, \mathcal{L}(\boldsymbol{\theta}; \mathcal{D})$$

for the network loss function $\mathcal{L}$. In the case of regression setting, the loss function sums the squared estimation error over all m training data[3]:

$$\mathcal{L}(\boldsymbol{\theta}; \mathcal{D}) = \sum_{i=1}^{m} \sum_{j=1}^{d_r} (\boldsymbol{r}_{i,j} - f_j(\boldsymbol{x}_i; \boldsymbol{\theta}))^2,$$

where the subscript j denotes the (scalar) j^{th} vector component. The general optimisation problem is non-convex in the weights, though for typical models, is differentiable. Hence solution estimates are commonly found with the support of gradient-based optimisation algorithms.

Indeed, such algorithms have gradient descent at their core. The ideas underpinning gradient descent-type optimisation algorithms, and in particular backpropagation for machine learning, can be found in work spanning decades of the last century [47, 48, 49]. In more recent times, with ever increasing data set and model sizes (in terms of the number of hidden layers and nodes), there has come an increased requirement for computational resources to support model development. Gradient descent lends itself to parallelisation via graphics processing units (GPUs) that, supported by open-source software, has seen estimation of deep networks become increasingly efficient.

[3] And in the case of classification, typically cross-entropy.

The idea of the gradient descent algorithm is to iteratively update the model weight parameter estimates in a way that minimises the loss. Such weight updates occur in the direction of the negative gradient of the loss (with respect to the target weight). A particular weight w is updated at the $(k+1)^{th}$ iteration according to

$$w^{(k+1)} = w^{(k)} - \gamma \frac{\partial \mathcal{L}}{\partial w^{(k)}}.$$

For the algorithm, the required (partial) derivatives are calculable via an application of the chain rule. The parameter γ is called the learning rate and determines the size of the adjustment to the parameter at each iteration. The algorithm terminates after a suitable convergence criterion (typically with respect to the loss function) is achieved, implying weight estimates in the neighbourhood of some reasonable local minimum.

Neural networks derivative updates via gradient descent encompass the notion of error 'backpropagation' [49]. For a model 'forward pass', the input data is used for calculating the model output error at the final model layer, given the existing weight estimates. Then for intermediate layers, at each node a weighted sum of errors from connected, subsequent nodes can be calculated, and used for computing the required partial derivatives.

Researchers have developed many innovations to the gradient descent algorithm. Learning rate specifications, or schedules, can lead to improved estimation convergence in many cases. The Adagrad algorithm innovates over learning rates parameterwise such that large weight updates are tempered, and relatively smaller weight updates are magnified [50]. The addition of a 'momentum' term – that weights successive gradient estimates with their past values – has been shown to smooth the sequence of weights learnt [49]. The 'Adam' algorithm innovates on gradient descent by calculating normalised gradients, which can add to algorithmic stability [51]. In the weight-update step of gradient descent, the entire training data can be used in calculating gradients, or data subsets (in which case the algorithm is known as 'mini-batch' gradient descent), or even a single observation per update ('stochastic' gradient descent) [52]. Such choices can be made empirically in a trade-off between computational efficiency and accuracy.

The predictive performance of an estimated model can be analysed across measures

of model quality or fit. In a theoretical instance, consider the bias-variance decomposition of [53], and the out-of-sample mean square error further averaged over all possible data sets $\mathcal{D}^*$ (of the same cardinality):

$$\mathbb{E}_{\mathcal{D}^*}\left[\mathbb{E}(r-\hat{f}(\boldsymbol{x}))^2|\boldsymbol{x},\mathcal{D}^*\right] = \sigma^2 + \text{bias}\left(\hat{f}(\boldsymbol{x})\right)^2 + \text{var}\left(\hat{f}(\boldsymbol{x})\right) \tag{2.10}$$

where

$$\text{bias}(\hat{f}(x)) = \mathbb{E}_{\mathcal{D}^*}\hat{f}(x) - \mathbb{E}r|\boldsymbol{x} \tag{2.11}$$

$$\text{var}(\hat{f}(x)) = \mathbb{E}_{\mathcal{D}^*}\left(\hat{f}(x) - \mathbb{E}_{\mathcal{D}^*}\hat{f}(x)\right)^2, \tag{2.12}$$

where $\hat{f}$ depends on the data $\mathcal{D}$ and σ^2 is some irreducible error of aleatoric nature. Then the bias term captures a quantum of model error whereby f is on an average an imperfect estimator of some unknown ground truth $\mathbb{E}r|\boldsymbol{x}$. Also, the variance of the estimator f, as given in Equation (2.4), can be large or small reflective of a high or low sensitivity to a particular draw of training data. It follows that a high variance estimator, for example, may not generalise well to a particular set of 'out-of-sample' data – some collection of data independent of that used in the calculations during model training – as evidenced by a relatively large empirical error. This issue is sometimes referred to as model 'overfitting'.

In this sense, training methods can be adapted via regularisation to reduce overfitting. There are many regularisation techniques, multiple of which can be simultaneously implemented with respect to the estimation of some model. A well-known regularisation method is that of weight decay, which adds a penalty term to the loss function given by an L_p-norm:

$$\mathcal{L}_{\text{reg}}(\boldsymbol{\theta};\mathcal{D}) = \mathcal{L}(\boldsymbol{\theta};\mathcal{D}) + \lambda||\boldsymbol{\theta}||_p^p,$$

for some (nonnegative) hyperparameter λ. Common choices are $p = 1$, yielding the so-called 'LASSO' penalty [54], and $p = 2$, the 'ridge' penalty [55]. A consequence of LASSO regularisation is to estimate relatively more weight parameters equal to 0 compared with ridge regression – acting somewhat as a device for automatic feature selection[4].

[4]Though we leave to the financial practitioner modelling noisy returns by a number of potentially correlated features to consider on what basis one approach may be more desirable than the other.

Another form of regularisation is 'early stopping'. In the sense of an iterative model estimation algorithm such as gradient descent, one can monitor a model error metric evaluated on a data subset independent of the training data. A heuristic rule may be set to define when estimation updates should cease, for example, if the monitored error is decreasing at too slow a rate, or if it has been increasing for too many iterations.

Dropout was first presented in [56] as a regularisation method to prevent neural network overfitting and to improve model performance on hold-out data. Dropout is implemented by randomly omitting some proportion of hidden units (by setting them to zero) in each training iteration of a neural network, and rescaling the final parameter values by this proportion at test time. The dropout rate presents as an additional model hyperparameter. Dropout was originally interpreted as giving a similar effect to calculating many different neural networks architectures (varying on which parameters were omitted) and taking their average result as a better predictor than any one model on out-of-sample data. Dropout has been extended for use in deep neural networks, and as well as being used for regularisation has led to a pseudo-Bayesian method of approximating network epistemic uncertainty. For a review of the many recent developments of dropout for deep learning we recommend [57].

Bayesian methods for neural networks

We end this section with a brief comment on Bayesian methods for uncertainty estimation in neural networks. Early work of MacKay delivers such a framework for feed forward neural networks [58]. A theoretical link between GPs and neural networks is established in the thesis of Neal [59], enabling exact Bayesian inference for regression problems through neural networks [60].

In modern deep learning research [61] is the first to to represent neural network weights as probability distributions and offer a (mean field) variational solution method. Indeed, the authors show that given the negative evidence lower bound (ELBO) of Equation A.1 as a cost function, unbiased gradient estimates are calculable by Monte Carlo sampling

to yield a 'backpropagation-like' algorithm popularly coined 'Bayes by Backprop' (BbB). This approach folds epistemic uncertainty into the modelling framework and yields a regularisation benefit. The approach generalises [62] who use the 'reparameterisation trick' to apply variational inference to the stochastic hidden units of an autoencoder. As with variational methods, in BbB the factorisation assumptions underlying the weight layers can lead to large biases that scale with the network size.

The idea of extending Bayesian uncertainty to neural network-based models has been presented through the lens of a technique called dropout. The recent analysis of [63] has shown that optimizing a neural network with both dropout and L_2-regularisation is equivalent to a variational inference-like model estimation (conditional on mild assumptions). The author's insight is that statistics calculable from Monte Carlo samples drawn from the dropout neural network at test time yield an epistemic uncertainty approximation. The approximating distribution $q(w)$ is the product of Bernoulli random variables and the respective weights. This makes sampling efficient since it amounts to drawing from Bernoulli random variables. There are particulars given different network layers. For example, [64] propose the same dropout mask for each time step for both input, output, and recurrent layers; see [65] further CNNs. Finally, and leading into the next brief section on tree-based models, we leave the reader with the intuitive idea that Monte Carlo dropout (or 'dropout sampling') can be viewed as an ensemble method – such methods are discussed further in the next section.

2.2.3 Decision trees

Decision trees are non-parametric models for learning a target variable as a function of a data set via a sequence of decision rules. They are applicable for both regression and classification problems, and the mechanics for either case are similar. Put simply, starting from a model's initial decision node – commonly called the 'root' – a data set feature is partitioned into two or more mutually exclusive groupings representative of decision node outcomes. For each grouping, the tree has either a subsequent decision node, or terminates

at a 'leaf' node. The leaf represents a value or class label estimate for the target variable. Characteristics of the decision nodes can be optimised with respect to a broad variety of criteria. In the long history of decision trees[5], the classification and regression tree (CART) algorithm of [67] has endured for some decades as a theoretical mainstay, and it can be efficiently implemented via a variety of statistical software packages.

There are many advantages to modelling with decision tree; not the least they are relatively simple to implement (compared to, say, a deep learning model), they can model non-linear relationships, and the decisions made and hence the output estimated are directly interpretable. On the other hand, decision trees have been observed to display 'high variance' [68]. In the case of regression-based trees, that is, the variance per the bias-variance decomposition of Equation (2.10), can be high. A suggested remedy to this is given in ensemble methods which average the estimates of multiple submodels as an approach to variance reduction, while not increasing model bias. For intuition about how this might work for a regression problem, consider a collection of n identically distributed random variable f with finite second moment σ^2. Then one can show that

$$\text{var}\left(\frac{\sum f_i}{n}\right) = \frac{1}{n^2}\left(n\sigma^2 + n(n-1)\bar{\rho}\sigma^2\right) = \frac{\sigma^2}{n} + \frac{(n-1)\bar{\rho}\sigma^2}{n} \tag{2.13}$$

where $\bar{\rho}$ denotes the average of the pairwise correlations between the members of $\{f_i : 1 \leq i \leq n\}$. We perceive immediately that if submodels are such that $\bar{\rho}$ is (i) negative, the variance reduction benefit is greatest; (ii) zero, then the result is the original variance scaled in inverse proportion to the number of submodels; (iii) positive, then any ensembling benefit erodes entirely as the average correlation reaches its maximum (of 1).

A common ensemble method is that of 'bagging' [69]. The main idea of bagging is to create bootstrap data sets from the original data, and then to independently estimate a collection of trees over the bootstrapped data sets. Subtree predictions are combined into an aggregate predictor that may yield a variance reduction benefit as outlined above. In the case of regression, the sample average is the aggregation function, while for classification

[5]The antecedents of decision trees within the statistics literature are outlined in [66], though the earliest tree models predate the references in this body of work.

predictions are combined based on a majority vote. Another ensemble approach, called 'random forests', are a bagging variant that – as well as bootstrapping the data set – withhold a random number of features within each bootstrap sample. This technique can reduce the pairwise correlation amongst the set of predictors, and hence $\bar{\rho}$ as given above. Further, such a reduction can come from a relatively elevated level (of epistemic uncertainty), compared to regular bagging. Regardless, both ensembling approaches are typically built from a CART algorithm-based estimation of the underlying trees, and so maintain a simplicity of implementation, while yielding somewhat powerful predictive models.

An alternative approach to ensembling can be found in 'boosting' models [70, 71, 72]. There are a variety of boosting algorithms; one such foundational procedure is found in Adaboost [73, 74] whereby (i) submodels are learned sequentially (rather than in parallel, as with bagging), and (ii) error feedback is propagated between submodels via a reweighting of the underlying data. That is, when estimating the next submodel, data subsets are assigned relatively larger (or smaller) weights if they were associated with larger (or smaller) errors in previous submodels. Gradient boosting is a second sequential algorithm that, starting from an initial (constant) function, adds successive submodels with the objective of minimising a loss function, for each step [75, 76]. The choice of submodel is set with respect to a principle of gradient descent in function space; it can be shown that a suitable summand is proportional to a tree trained on (pseudo-) residual error terms of the most recent model.

Many modern innovations in decision tree theory extend the set of boosting models. The XGBoost ('extreme gradient boosting') algorithm is similar to that of gradient boosting; an obvious difference is that both the submodel training data and the optimisation problem depend on the first *and* second derivatives of the loss function [77]. That is not to say one algorithm is strictly 'better' than the other; there are trade-offs between the models regarding the details of tree structure estimation, and we defer the interested reader to [78] for a richer comparison. The LightGBM framework is a compelling alternative to XGBoost; the key algorithm can yield faster implementation and execution, with sustained performance [79]. One component of the efficiency gain of LightGBM, for example, is

via a novel weighting of data points, favouring those associated with larger gradients. An alternative gradient boosting algorithm is given by CatBoost, which, amongst other benefits, natively supports categorical features [80].

There are many similarities between the training of decision tree ensembles and the deep learning techniques introduced in this section. For example, the objective function is subject to specification and tuning, both L_1 and L_2 regularisation are applicable, and hyperparameter optimisation – such as with respect to learning rate – is typically a critical practical matter. On the other hand, there are hyperparameters unique to trees that should also be managed by cross-validation; for example, maximum tree depth, and (in the case of XGBoost) the minimum number of samples required for a new node, both of which can be controlled in an attempt to reduce model overfitting. Finally, as with deep learning, ensemble methods can be modified to include SHAP for interpreting predictions via feature importance [81], via a variety of modern software packages.

Further the comparison to deep learning, ensemble methods have been extended via a consideration of uncertainty, which can be modelled in various ways. On one hand, models based on bagging are not explicitly probabilistic, though, as per Equation (2.13), a measure of epistemic uncertainty can be approximated as a function of the collection of ensemble predictions. Recent work on random forests considers model epistemic and aleatoric uncertainty for a classification setting [82]. Gradient boosting, on the other hand, has methods for incorporating both epistemic and aleatoric uncertainty. The NGBoost ('Natural Gradient Boosting') algorithm extends gradient boosting to jointly learn the parameters of a conditional probability distribution via the loss function, yielding an estimate of aleatoric uncertainty [83]. Aleatoric uncertainty, and a pseudo-Bayesian estimate of epistemic uncertainty based on the sampling a collection of models – essentially an *ensemble-of-ensembles* – is considered in [84], and implemented within the Catboost library.

We conclude this section with a note on time series and time-independent data for ensemble methods based on trees. Of course, tree methods are not traditional time series models. A researcher may find techniques applicable to bagging, such as the block

bootstrap [85], that robustify such models to particular characteristics of a time series – such as heteroskedasticity or autocorrelation[6]. Regardless, in that ensemble methods are a workhorse for modelling tabular data, one might therefore aggregate time series data over historical windows to engineer features that conform to the structure of tabular data. The availability of ensemble methods for problems in financial trading and investment is advantageous, not the least for the relative simplicity with which many ensemble methods can be implemented, be it for efficient preliminary and/or exploratory analysis, or in lieu of other more complex modelling approaches outright on a performance basis.

2.3 Risk-return based trading and investment

Models appropriate for describing edge require a terminal implementation strategy. There are many reasons a well-fitting model with naive backtest do not translate to useful tools in practice. At a minimum, terminal strategy for portfolio weightings (or trade 'sizings'), as well as consideration of transactions costs, seem reasonable – if not essential – additions to a strategy model. With this in mind, we present some enduring frameworks and ideas in this subsection.

2.3.1 Mean-variance portfolio optimisation

Consider a single-period investment setting. Assume that there exist n investable assets within a market with *excess* return vector $\boldsymbol{r}$ whose distribution is modelled as multivariate normal over discrete time increments, denoted $\boldsymbol{r} \sim \text{MVN}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. Here $\boldsymbol{\mu}$ denotes the mean vector of returns, and $\boldsymbol{\Sigma}$ denotes the return covariance. Then the foundational Markowitz MPT framework [3, 86] gives a solution to the mean-variance optimisation problem:

$$\max_{\boldsymbol{w} \in \mathcal{C}} \quad \mathbb{E}(\boldsymbol{w}'\boldsymbol{r}) - \frac{\gamma}{2}\text{var}(\boldsymbol{w}'\boldsymbol{r}).$$

where the investor portfolio weights $\boldsymbol{w} = (w_1, ..., w_n)$, $\mathcal{C}$ is a set of weight constraints, if any, and $\gamma > 0$ is the investor risk aversion parameter, denoting preferences balancing

[6] This may be an avenue of future research – if applicable at all.

return and risk.

One can impose various weight constraints such as that the sum of portfolio weights, or absolute weights, is 1, or that weights are non-negative, corresponding to the precise objectives of an investment strategy. This induces complex optimisation problems, though Markowitz also proposes the efficient Critical Line Algorithm in the solution toolkit. Otherwise, in the simple unconstrained case the solution to the optimal portfolio weights $\boldsymbol{w}^*$ in terms of $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is

$$\boldsymbol{w}^* = (\gamma \boldsymbol{\Sigma})^{-1} \boldsymbol{\mu}. \tag{2.14}$$

This solution also yields the maximum portfolio Sharpe ratio. The parameters $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ can be estimated by the sample mean vector and sample covariance matrix, given historical data, so that the weights $\boldsymbol{w}^*$ can be calculated by direct substitution. This approach has led to problems in practice, and particular cases are prone to corner solutions or otherwise impractical weight estimates. Ziemba questions the impact of estimation error in the above parameter estimates with an empirical study [87]. There have been a myriad of suggestions for improved estimation methods added to the literature. The covariance shrinkage method of [88] is recommended as a substitute for the sample covariance matrix, while the Bayes-Stein estimator is given in [89] as a popular alternative for the sample mean. A quite different approach is to express returns in terms of linear combination of factors in the framework of Ross's APT [10]; we revisit this in later chapters.

More recently – research on portfolio optimisation has focus heavily on the risk component; we note this here for context, and to support intuition. The method of volatility targeting [90] and risk parity [91] set portfolio weights in inverse proportion to asset volatility. Similarly hierarchical risk parity [92]: hierarchical models can be understood in terms of a (machine learning-based) cluster model, in lieu of a prediction model, that groups similar assets before applying the risk-targeting optimisation routine on partitions. [93] shows that the simple 1-on-N portfolio (of N assets) can outperform many more sophisticated methods, at least making for a compelling benchmark.

There are many extensions to Markowitz beyond innovations applied to parameter estimation. Investing is extended for a multi-period setting in [94], beyond typical single-period considerations. Another important extension is to incorporate transaction costs, which have noticeably been either omitted or suboptimally implemented in many studies [95]. There are many ways to account for transaction costs, including fixed and linear costs [96, 97]. An elegant approach to this is to make a minimal adjustment to the Markowitz objective of Equation (4.1):

$$\max_{\boldsymbol{w}\in\mathcal{C}} \qquad \mathbb{E}(\boldsymbol{w}'\boldsymbol{r}) - \frac{\gamma}{2}\text{var}(\boldsymbol{w}'\boldsymbol{r}) - TC,$$

with TC denoting transaction costs measured in percentage of invested wealth. Solution methods for a single-period model can be found in [94, 98]. Such methods depend on a transaction cost model accounting for portfolio size, portfolio turnover over time increments, daily volume traded of an asset and hence an estimate of market impact to size traded. Given the above specification dollar transaction-costs erode dollar returns quadratically with increasing wealth. Also, the above specification has the added benefit of regularising turnover. Regardless, these terms are estimable from empirical data. For example, the recent estimates of [99] suggest a 10 basis point market impact when trading 1% of the daily dollar volume of some US equity (whether long *or short*).

Other strategies include Sharpe optimisation for non-linear models at the level of loss function, whereby the estimated model outputs position size directly [100]. Such an approach adds difficulties being so literally data-driven – epistemic uncertainty may be difficult to incorporate; model interpretability can decrease; and the method is not proven to scale to a large portfolio of correlated trades.

We finish this subsection with a concept that will reprise in a later chapter. Implicit in multistrategy hedge funds is the need to allocate from centralised capital between alternative substrategies with the view to maximise overall portfolio Sharpe ratio. A related (but unique) problem of allocating between strategies can be found in [101]. The author considers combining a long-only portfolio, and an uncorrelated long-short strategy. Even if the long-short strategy alpha is small, investor may seek to add it to the core long portfolio

since in the case of (uncorrelated) strategies:

$$S^2 = S_L^2 + S_{LS}^2.$$

For S the combined portfolio Sharpe ratio, and the subscripts L and LS denoting the long-only and long-short portfolios. We see immediately that, given our assumptions, the combined strategy is never worse off.

2.3.2 Black-Litterman

An extension to Markowitz's MPT is given by the Black-Litterman model [12]. The Black-Litterman model amounts to a portfolio optimisation framework that incorporates (subjective) investor beliefs about asset returns. Such beliefs are termed 'views'. Views can be obtained from diverse sources, including market pundit opinions, Reserve Bank statements, and as output of statistical models.

In its original formulation, the Black-Litterman framework assumes asset returns as in Equation (2.1). However, the mean parameter $\boldsymbol{\mu}$, in addition, is assumed to be a random variable with prior distribution such that $\boldsymbol{\mu} \sim \mathrm{N}(\boldsymbol{\Pi}, \boldsymbol{\Sigma}^{\mu})$. The original authors – and many researchers since – initialise the mean parameter $\boldsymbol{\Pi}$ of the prior distribution based on an estimated CAPM model per Equation (2.2). This underwrites the analysis with a foundational theoretical result. The uncertainty term $\boldsymbol{\Sigma}^{\mu}$ is given by $\tau\boldsymbol{\Sigma}$, for $\boldsymbol{\Sigma}$ the return covariance and τ a scaling parameter. The scaling parameter is set heuristically, for example, depending on confidence interval widths as estimated from historical data. The precise specification of the view uncertainty continues to be researched within academic literature; in Chapter 4 we offer a new Bayesian perspective.

Investor views on market asset returns are expressed

$$\boldsymbol{P}\boldsymbol{\mu} = Q + \epsilon,$$

where $\boldsymbol{P}$ is a $K \times N$ matrix whose rows are asset weights, and Q is a K-vector of the expected returns on these portfolios. The error term given by $\epsilon \sim \mathrm{N}(0, \boldsymbol{\Omega})$ is assumed to be is independent of the asset return covariance and the covariance of the prior.

The work of Black and Litterman shows that views induce an update of the return distribution parameters. Firstly, the expected return given the views can be shown to be Normally distributed with mean $\boldsymbol{\mu}_v$ and covariance $\boldsymbol{\Sigma}_v$ given by

$$\boldsymbol{\mu}_v = \boldsymbol{\Sigma}_v^{-1}[(\tau\boldsymbol{\Sigma})^{-1}\boldsymbol{\Pi} + P^T\Omega^{-1}Q],$$
$$\boldsymbol{\Sigma}_v = [(\tau\boldsymbol{\Sigma})^{-1} + P^T\Omega^{-1}P]^{-1}.$$

Then, the return distribution given the views is also Normal with mean parameter $\boldsymbol{\mu}_v$, and covariance $\boldsymbol{\Sigma}_v + \boldsymbol{\Sigma}$. Once the return distribution given the views is estimated, one can proceed with mean-variance optimisation to yield a solution to a portfolio allocation problem.

Since its introduction, the Black-Litterman model has been formalised through the lens of Bayesian statistics. The interested reader can refer to [102] for a summary of the Bayesian Black-Litterman theory, and a review of various extensions and reinterpretations. The works [103, 104] make for intriguing supplements. Notably, [105] extend the Black-Litterman approach with respect to an underlying factor model of returns – as in Equation (2.3) – in a Bayesian setting. In this case, views relate to beliefs about temporal factor evolution, rather than asset returns. More can be found on this approach, and a novel extension, in Chapter 4 of this thesis.

2.3.3 Kelly criterion

In the 1950s – the decade that MPT changed finance – the Kelly criterion also emerged [5]. The Kelly criterion gives the optimal proportion of bankroll to wager to maximise bankroll growth in (i) a multi-period decision-making setting, where (ii) one can optionally wager discrete independent 'bets' with known payoffs to unit staking, b, and known probabilities of receiving those payoffs, p. The optimal proportion can be found to be the ratio of 'edge', $pb - (1-p)$, to 'odds' b – an acceptable wager assuming the ratio is greater than zero. Around a similar time, Latane presented the strategy of wager sizing based on geometric mean maximization (GMM) with respect to a (discrete) probability distribution of returns,

to maximise the probability of realising the greater terminal portfolio value after a large number of (independent) wagers, relative to other 'reasonable' alternative strategies [106].

Some decades later, Thorpe [107] presented a continuous approximation to the Kelly framework subsuming a limiting log normal diffusion process for an underlying asset's price evolution, and an investor instantaneously rebalancing their portfolio weightings between the asset and risk-free cash. The optimal position size in terms of fraction of bankroll f^* wagered on an opportunity is given elegantly:

$$f^* = \frac{\mu}{\sigma^2},$$

where μ denotes the asset return in excess of the risk-free interest rate. This result is extendable to the case of two or more risky assets modelled by an underlying multivariate geometric Brownian motion [108]. In this case, f^* is the same as in Equation (2.14), assuming the risk aversion parameter equals 1. That is, there is an equivalence between this Kelly approach and the MPT framework outlined above.

Risk-aversion can be implemented into the Kelly framework; a well-documented downside of the Kelly strategy is the extreme volatility of portfolio-level returns that may realise. Increasing risk-aversion amounts to wagering in proportion $k \in (0, 1)$ to f^*; for example, for $k = 1/2$ is known as wagering 'half-Kelly'. Here the optimal growth rate is reduced in an attempt to reduce realised volatility . Over-wagering, on the other hand, occurs for $k > 1$ and is detrimental to the portfolio growth rate [107]. For k sufficiently large, the portfolio growth rate becomes negative, and the investor should expect to eventually lose their entire bankroll. This point adds support to the task of careful estimation of the mean and covariance parameters used in portfolio allocation – misestimation in either term could lead to overstaking and eventual ruin.

The Kelly strategy – tempered or not for risk-aversion – is not immune from expert criticism; Samuelson claims the arbitrariness of the strategy with a counterexample for an investor with a particular utility function [109]. Indeed, a more general asset allocation framework can be described in terms of utility theory. This is a deep theory from the economics literature, and nearly 300 years old with its origins in Bernoulli [110]. An

account of utility theory is beyond the scope of this work. We note however, that the theory subsumes (i) Markowitz as a special case given a specific utility function or return distribution assumptions, and, expectedly then, (ii) the Kelly criterion. We recommend to the interested reader [111] for a recent, preliminary review.

We complete this section with a practical observation about temporal portfolio performance and the assumption of Normally distributed underlying asset returns that began this chapter. This could motivate future research, and is also an important practitioner note, with a lesson for anyone that considers decision making under uncertainty.

Consideration of a limit of Equation (2.1), as $\Delta t \downarrow 0$, leads to a well-known continuous time model for stock price evolution called geometric Brownian motion:

$$\frac{dS_t}{S_t} = \mu dt + \sigma dB_t, \tag{2.15}$$

where B denotes a standard Brownian motion. By Ito's lemma, a celebrated result within the theory of stochastic calculus, it can be determined that

$$\ln\left(S_t/S_0\right) \sim N\left(\left(\mu - \frac{1}{2}\sigma^2\right)t, \sigma^2 t\right),$$

for time t and initial asset price S_0 [112].

Now, again as per [14], the continuously compounded rate of return of a process, c, is defined

$$c = \frac{1}{t}\ln\left(S_t/S_0\right).$$

This random variable denotes the (geometric) growth rate of a process, and it immediately follows that it has expectation $\mathbb{E}c = \mu - \frac{1}{2}\sigma^2$. The second term in this expression is known as the 'volatility drag'. The drag differentiates the expected process growth rate from the process's expected annualised return. Intuitively, on inspecting $\mathbb{E}c$, one may conjecture that a process can tend to zero with time, even for some cases where both μ and σ^2 are positive (further consideration of this is beyond the scope of this work). Secondly, consistent with observations in [113], extreme downside risk (or downside 'tail' risk), particularly that inconsistent with our modelling assumptions, may be of practical concern due to its outsized contributions to the volatility drag. This may motivate a market participant to

hedge or otherwise insure such risk, and such a strategy could exist within a portfolio as a supplementary source of 'edge', improving the portfolio growth rate. Secondly, of course, hedges come with associated costs, so while the portfolio risk experienced may decrease in the extremes, portfolio-level expected return may also be eroded – but hopefully sufficiently slowly to justify the hedge. Regardless, in such a case, realised absolute performance could be scaled to achieve targets via an increase in net portfolio leverage. Again, this is beyond our scope, but makes for an interesting and important consideration in applied trading and investing.

2.4 Conclusion

In this chapter we briefly review a wide breadth of information pertaining to quantitative finance. The first part concerns a subset of models from machine learning, statistics, and econometrics with utility for edge modelling. In the second part, we discuss terminal strategy that dictates position sizing and bankroll management given the estimates yielded from a model. We claim the interrelatedness of each part for quantitative finance applications in pursuit of a market edge. Many of the tools and ideas introduced are showcased in the following chapters with respect to both quantitative trading and investing.

Chapter 3

Position sizing using model uncertainties for intraday trading

3.1 Introduction

In practice, it is common for investors to size investment positions based on conviction in trade ideas. Some traders highlight the importance of investment sizing at least with the view to leverage exposure to the positive right-tail of the profit and loss distribution [114]. However, there is relatively little by way of academic discussion regarding achieving the task in a data-driven way. Inspired by recent innovations in financial machine learning, we present a case for utilizing uncertainty estimates gleaned from a prediction model, for the purpose of investment sizing within a trading strategy. Further, we show how this can improve relative investment performance.

The trading strategy depends on a financial asset price prediction model at its core. In this context, and in the era of big data, deep learning models often present as the state of the art. This is particularly true for models estimated given high-frequency limit order book data, whereby many millions of pieces of unique information may be collected and processed according to data-intensive algorithms. In recent work, prediction models for financial time series have been improved on by deep convolutional neural networks (CNN) and long short-term memory (LSTM) networks [115, 116]. A state-of-the-art CNN-LSTM with inception (CNN-LSTM Inc) was shown to be of utility for formulating financial prediction as a classification problem [11].

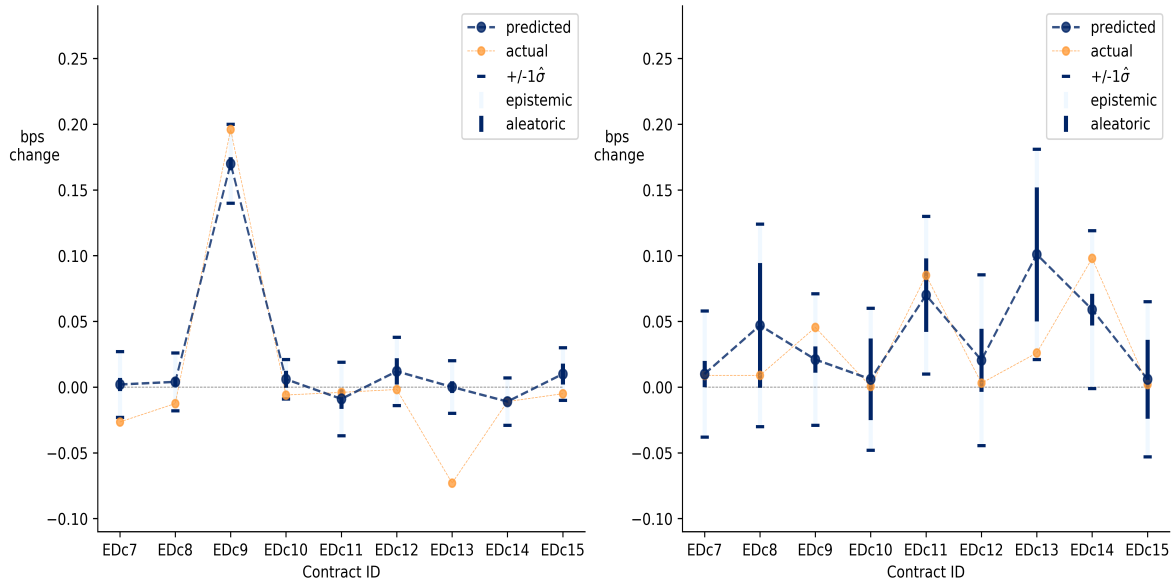

Figure 3.1: We utilize deep learning models of changes in a subset of the interest rate curve for a collection of liquid Eurodollar futures contracts, and recover uncertainty estimates around predictions. Above, predictions for the next event time are shown in dark blue, versus the actual realised outcome in orange. Total uncertainty is shown as the one-standard deviation dashed light blue lines, which subsumes an estimate of aleatoric (epistemic) in the part shaded dark blue (ice blue). The left-hand image depicts a prediction more certain, and corresponding closer to the target outcome, relative to the right-hand image.

We build from this work to model high-frequency data at Level 1 of the Eurodollar Futures[1] limit order book. Important variations include our framing of prediction as a regression problem for modelling outright price changes, and that both our input and output data are over a multi-dimensional asset price space. Further, this approach complements recent literature advancing interest rate derivative price prediction [117, 118].

However, while there is ongoing innovation and success applying deep learning prediction models in finance, less has been shown by way of retrieving meaningful prediction uncertainties. This is despite innovations and demonstrable utility within a diverse range of application domains, including computer vision, medicine, and astrophysics [21, 119, 120].

[1]Here, we note the class of derivative in focus are the popularly traded interest rate derivatives, whose reference contract interest rate applies to a hypothetical United States dollar-denominated unsecured bank funding deposit – rather than a foreign exchange derivative on the EURUSD currency pair.

Of course, prediction uncertainty can arise from a variety of sources, though two stylized facets of uncertainty are often modelled. On one hand, aleatoric uncertainty; we model this explicitly as expressed through a (spatial) heteroskedastic variance-covariance matrix of the predicted prices. On the other hand, and less directly, we model epistemic uncertainty as induced by the model specification. In the context of deep learning, pseudo-Bayesian approximations to epistemic uncertainty have been proposed. For example, it has been shown that epistemic uncertainty can be cheaply approximated in deep learning models with dropout [63]. Given that dropout is often a component of deep learning models for financial prediction, and the relative ease by which dropout sampling can be implemented, this approximation seems a practical, yet relatively underexplored, component of prediction uncertainty for financial applications; refer to [121] for an early attempt.

In any case, the utilization of model predictions and uncertainty estimates – such as displayed in Figure 1 – can become apparent in the details of a trading strategy. A proof of concept of potential usefulness can be found in improving standard and preliminary performance metrics over which trading strategies are typically evaluated. We target improvement in the Sharpe ratio [4]. Some authors have improved on a vanilla long/short trading strategy by directly incorporating a measure of Sharpe in the loss function [100, 122], or by applying parameterised decision rules for investment sizing [123]. Our approach is closer in spirit to the latter: with the addition of reference uncertainty estimates for model-based beliefs of price returns, we can optimize the Sharpe metric inspired by how some successful traders intuitively adjust trade sizes in the real world – by scaling into high conviction trades (corresponding to lower uncertainty) offering sufficiently large reward with relatively more bankroll.

In the subsequent sections, beginning with Section 3.2, we specify the data set and preprocessing steps (3.2.1), the model (3.2.2), and the trading strategy (3.2.3). Results and discussion are offered in Section 3.3. We conclude in Section 3.4 and offer avenues for future research.

Table 3.1: Median daily number of quotes, trades, and total volume executed for Eurodollar futures contracts EDc1 to EDc15. Data recorded on each (US) trading day of 2018. Units are in 000s. Data source: Thomson Reuters.

	1–6	7	8	9	10	11	12	13	14	15
Quotes	14.4	157.5	242.6	243.8	308.8	287.1	286.2	317.3	264.2	274.8
Trades	0.8	2.0	1.8	2.2	1.9	2.3	1.8	1.6	1.5	1.8
Volume	47.9	37.3	34.5	48.7	28.7	24.5	19.5	22.1	10.9	11.7

3.2 Data and modelling choices

3.2.1 Data commentary

Seeking to model high-frequency Eurodollar futures prices, and hence a component of an interest rate curve, we have collected Thomson Reuters data for continuous contract EDc1 to EDc15 for each US trading day of 2018. This data reflects Level 1 limit order book best bid-ask-volume quotes, with some additional trade execution data. The first 15 contracts are a subset of the 44 available market traded contracts through CME group. In terms of the number of quotes and trades, and total volume traded, we observe greater trading activity in contracts EDc7 to EDc15, relative to the earlier contracts; see Table 3.1 for reference. Owing to this market activity, we focus our analysis on EDc7 to EDc15.

A key step of our data preprocessing is to construct a time series of microprices for each contract c, $P^c := \{P^c_t : t \in T^c\}$, such that

$$P^c_t := \frac{\text{ask volume}^c_t \cdot \text{bid price}^c_t + \text{ask price}^c_t \cdot \text{bid volume}^c_t}{\text{ask volume}^c_t + \text{bid volume}^c_t},$$

following the definition of microprice in [124]. Table 3.2 displays various percentile levels for the median daily difference in the high versus the low in P^c, offering some intuition around daily variability by contract. This stands in contrast to the median daily bid-ask spread of 0.5 basis points (bps) for each contract, throughout 2018.

A second key step is to align and down-sample $\{P^c : c \in [\text{EDc7}, \dots, \text{EDc15}]\}$. Aligning over index sets T^c is essential given the irregular temporal sampling across contract microprices, while down-sampling conveniently filters for sufficiently small microprice fluctuations, significantly reducing the size of the aligned data set. We define a

Table 3.2: Summary statistics for the yield data used in these notes: difference in contract microprice levels (bps) high and low on each (US) trading day of 2018; percentiles and daily standard deviation. Data source: Thomson Reuters.

	EDc7	EDc8	EDc9	EDc10	EDc11	EDc12	EDc13	EDc14	EDc15
min	0.3	1.2	1.4	1.6	1.7	1.8	1.3	1.1	1.6
25th perc.	2.3	2.8	3.1	3.4	3.6	3.8	4.0	4.0	4.0
50th perc.	3.0	3.7	4.3	4.6	4.8	5.0	5.1	5.1	5.3
75th perc.	4.4	5.3	5.9	6.5	6.9	6.9	7.2	7.3	7.5
max	16.2	17.1	20.6	22.1	23.3	24.9	25.8	26.2	26.2
σ(daily Hi-Lo)	2.4	2.6	2.8	3.0	3.2	3.2	3.3	3.4	3.3

cutoff parameter M and set it to 0.1bps, and for each trading day construct a time series of curve observations $C := \{\boldsymbol{P}_t = (P_t^c : c \in [\text{EDc7}, \dots, \text{EDc15}]) : t \in T\}$ for a common temporal index set T, according to the following rules:

1. For a new trading day, record an initial curve observation at the first time that there exists a microprice for each contract;
2. Given an observation time t^*, find the earliest time $t > t^*$ such that $|P_t^c - P_{t^*}^c| \geq M$ for at least one contract c, and record an observation of each contract's most recent microprice at or equal to t. If such a time does not exist, revert to 1. for the next trading day.

Finally, we construct a time series $D := \{(\boldsymbol{D}_t, \boldsymbol{Y}_t) = ((\boldsymbol{P}_{t+k} : k \in \{-99, \dots, 0\}), \boldsymbol{P}_{t+1}) : t \in T\}$ consisting jointly of a rolling window of 100 sequential curve observations[2], and the curve observation at the next event time. Further, for each member of D, we implement a window normalization whereby each microprice subseries in $\boldsymbol{D}_t$ is shifted by its empirical mean within the window, and scaled by the standard deviation of the set of all (shifted) window values. Similarly, we then normalize the corresponding members of $\boldsymbol{Y}_t$ using these same statistics (computed over the backward-looking window). To be clear, we comment that all no step of our data preprocessing introduces look-ahead bias.

[2]The value '100' is chosen for comparison with [11], and has been seen to work well in practice. Note no extensive optimization of this parameter has been performed here.

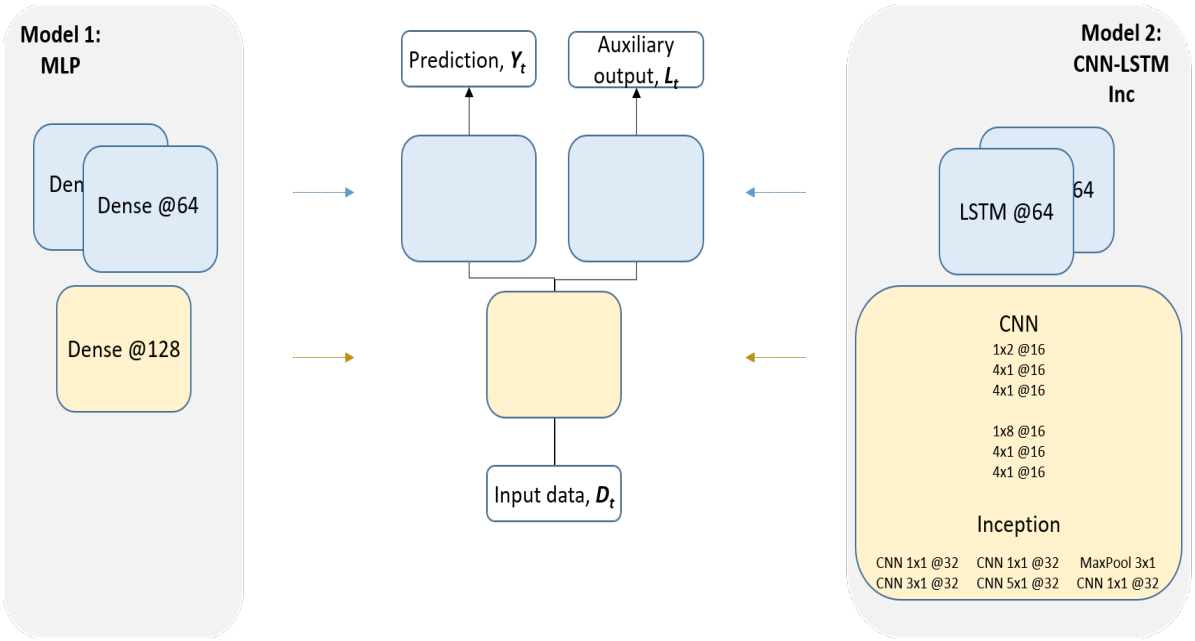

Figure 3.2: Model architectures for predicting curve moves with simultaneous aleatoric uncertainty learning. The MLP specification was chosen such that it had a similar number of parameters to the CNN-LSTM Inc – the latter model has nearly $100,000$ parameters, while the former has about $133,000$. Anecdotally, we determined that the branching architecture, featuring a common module before branching (shaded yellow above), improved fit significantly relative to the exclusion of a common module.

3.2.2 Model specification

Given our data set construction, we model the prediction target $\boldsymbol{Y}_t$ as a multivariate normal random variable with parameters dependent on our window observations:

$$\boldsymbol{Y}_t \sim \text{MVN}(\boldsymbol{\mu}(\boldsymbol{D}_t), \boldsymbol{\Sigma}(\boldsymbol{D}_t)). \tag{3.1}$$

For brevity, we continue by writing $\boldsymbol{\mu}_t := \boldsymbol{\mu}(\boldsymbol{D}_t)$ to denote the mean vector of dimension $1 \times c$ and $\boldsymbol{\Sigma}_t^A := \boldsymbol{\Sigma}(\boldsymbol{D}_t)$ to denote the variance-covariance matrix of dimension $c \times c$ (and with a superscript A to denote "aleatoric" uncertainty [21]).

We consider two model architectures for estimating $\boldsymbol{\mu}_t$ and $\boldsymbol{\Sigma}_t^A$. With respect to $\boldsymbol{\mu}_t$, we are inspired to adapt the CNN-LSTM Inc of [11], and we seek to compare trading strategy performance to a more straightforward 2-hidden layer multi-layer perceptron

(MLP). Further details for the architectures are offered in Appendix B, and a schematic of the model architectures are as depicted in Figure 3.2. A key point of interest, and novel for financial applications, is the network branching, whereby we estimate our targets over two output heads. We recall [125] that appends an auxiliary output to a network architecture to yield a heteroskedatic variance estimate in conjunction with predicting the mean of a univariate Gaussian. This was extended recently for the multivariate case [126]. Requiring that $\mathbf{\Sigma}_t^A$ is a symmetric positive definite matrix, its inverse is also symmetric positive definite and can be expressed as its Cholesky decomposition. Hence, writing $(\mathbf{\Sigma}_t^A)^{-1} = \boldsymbol{L}_t \boldsymbol{L}_t^T$ for a lower triangular matrix $\boldsymbol{L}_t$, as per [126], and with minor notional updates for our specific problem, the network loss function $\mathcal{L}$ has been shown to be

$$\mathcal{L} = -2\Big[\sum_{i=1}^{c} \log l_t^{ii}\Big] + (\boldsymbol{Y}_t - \boldsymbol{\mu}_t)^T \boldsymbol{L}_t \boldsymbol{L}_t^T (\boldsymbol{Y}_t - \boldsymbol{\mu}_t). \tag{3.2}$$

Here l_t^{ii} denotes the ith diagonal entry of $\boldsymbol{L}_t$. It is interesting to note that $\mathbf{\Sigma}_t^A$ is implicitly learnt through the loss function, without reference to explicitly labelled variance-covariance data. We estimate $\boldsymbol{L}_t$ as output to each model, applying a linear activation function after the final layer. The diagonal elements are then exponentiated, ensuring the positive definiteness and hence invertibility of $(\mathbf{\Sigma}_t^A)^{-1}$, so that $\mathbf{\Sigma}_t^A$ is recoverable.

There are various proposals for approximating epistemic uncertainty for deep neural network models, including pseudo-Bayesian approximations. One popular technique applies to models estimated with dropout.

Recall that dropout was first presented in [56] as a regularization method to prevent neural network overfitting and improve model performance on out-of-sample data. The intuition for implementing dropout is to randomly omit some proportion of hidden weights (by setting them to '0') for each training iteration of a network, and rescaling the final model weights by this proportion at test time. This was originally interpreted as giving a similar effect to calculating an ensemble of models (depending on which parameters were omitted) and taking their average result as a better predictor than any one model on out-of-sample data. In the context of modern deep learning, dropout is popular amongst some practitioners for the perceived regularization benefit and for its relative ease of

implementation. The key to retrieving the epistemic uncertainty approximation is the technique of 'dropout sampling' [63]. This technique is applied by generating many model predictions for a single input datum with random dropout sampling enabled at test time. It has been shown that one can naturally retrieve an approximation to epistemic uncertainty for predictions, calculating the estimate in a straightforward manner by averaging [63].

Hence, we have that for N stochastic dropout samples associated with a single model prediction, our parameter estimates are calculated as:

$$\widehat{\mathbb{E}\boldsymbol{Y}_t} := \hat{\boldsymbol{\mu}}_t = \frac{1}{N}\sum_{n=1}^{N}\hat{\boldsymbol{\mu}}_{t,n} \qquad \text{and} \qquad \hat{\boldsymbol{\Sigma}}_t^A = \frac{1}{N}\sum_{n=1}^{N}\hat{\boldsymbol{\Sigma}}_{t,n}^A,$$

with the expectation estimate according to [63], and where the covariance estimate conforms to [127]. Again, as in [127], an estimate of the epistemic uncertainty of the prediction is given by

$$\hat{\boldsymbol{\Sigma}}_t^E = \frac{N}{N-1}\cdot\left(\frac{1}{N}\sum_{n=1}^{N}\hat{\boldsymbol{\mu}}_{t,n}\hat{\boldsymbol{\mu}}_{t,n}^T - \hat{\boldsymbol{\mu}}_t\hat{\boldsymbol{\mu}}_t^T\right),$$

where we have made an additional adjustment to yield an unbiased sample covariance estimate. Hence the total prediction uncertainty estimate can be written

$$\widehat{\text{Var}(\boldsymbol{Y}_t)} := \hat{\boldsymbol{\Sigma}}_t = \hat{\boldsymbol{\Sigma}}_t^A + \hat{\boldsymbol{\Sigma}}_t^E.$$

3.2.3 Trading strategy

In this section, we present a trading strategy that depends on our model outputs $\hat{\boldsymbol{\mu}}_t$ and $\hat{\boldsymbol{\Sigma}}_t$ for selecting α_t, the investment size for a particular trade at each decision time t.

In the literature, a typical backtest given a classification or regression model often amounts to selecting a trading decision based on one of 3 choices: sell 1 unit, do nothing or buy 1 unit, respectively corresponding to $\alpha \in \{-1, 0, 1\}$. However, a more flexible range of permissible investment sizes, such as the case of continuous $\alpha \in [-1, 1]$, could lead to trading strategies showing non-trivial improvement across real-world performance outcomes. When interpreting $\hat{\boldsymbol{\mu}}_t$ and $\hat{\boldsymbol{\Sigma}}_t$ as true parameters of the underlying return distribution, one can intelligently scale relative investment sizing depending on a pre-defined trading objective. An example objective that we present is to maximise the

	Loss		Mean (Std dev.) SE	
month	MLP	CLI	MLP	CLI
7	-4.91	**-6.74**	**0.483*** (2.335)	0.491 (2.275)
8	-1.83	**-5.64**	0.744 (2.691)	**0.719*** (2.482)
9	-1.83	**-4.93**	**0.738*** (2.593)	0.756 (2.474)
10	-1.22	**-4.33**	0.647 (2.141)	**0.615*** (2.012)
11	-4.08	**-6.66**	0.519 (2.093)	**0.474*** (2.011)
12	-	-	-	-
Avg	-2.77	**-5.66**	0.626 (2.371)	**0.611** (2.251)

*p ≤ 0.0001 for each one-tailed test of least model MSE by month

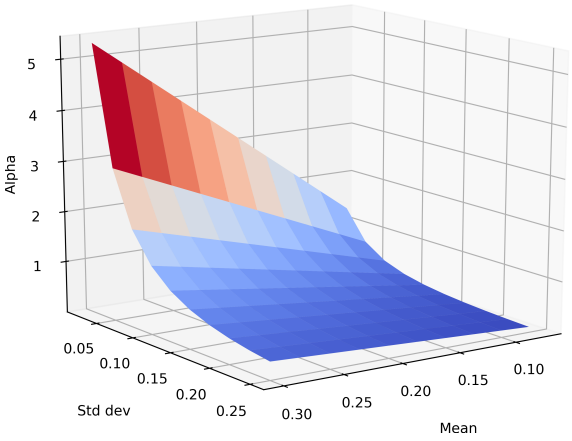

Figure 3.3: Left: Validation loss and standard error statistics by month for MLP and CNN-LSTM Inc models. Each model is estimated using the full year's data, up to but excluding the validation month. Right: α-surface of Section 3.2.3, with α set to 1 for $(\mu, \sigma) = (0.3, 0.1)$.

out-of-sample Sharpe ratio trading performance metric. Assuming additive rather than multiplicative returns, and further assuming the simplified setting whereby trade opportunities present for each estimated pair of distribution parameters a uniformly equal number of times, we show in Appendix B that the optimal investment size is given by

$$\alpha_{t,i} = \frac{\hat{\mu}_{t,i}}{\hat{\sigma}^2_{t,i}}.$$

Here, $\mu_{t,i}$ denotes the ith element of $\hat{\boldsymbol{\mu}}_t$, and $\hat{\sigma}_{t,i}$ denotes the square root of the ith diagonal element of $\hat{\boldsymbol{\Sigma}}_t$.

To assist intuition, we chart the corresponding α-surface on the right-hand panel of Figure 4.1, where we have made the additional arbitrary choice to rescale the surface such that the point $(\mu, \sigma) = (0.3, 0.1)$ corresponds to $\alpha = 1$.

There are two pleasing outcomes of this approach relative to [121], an early work to utilize an approximation to Bayesian uncertainty estimation for trading given the predictions of a deep classification model. Firstly, we do not increasingly penalise a trade based on increased uncertainty alone, but rather based on risk-adjusted returns. This corresponds to real-life, where high uncertainty for a trade is not necessarily outright undesirable – a trade proposition should have its estimated uncertainty considered in conjunction with the expected return size. Secondly, we do not depend on an ad hoc rules-based approach for

setting α_t, and instead select it in a more intuitively pleasing way.

3.3 Results

Given we have one full year of data, we partition by month and evaluate and verify our models by training on months 1 to $M-1$, validating using month M, and finally testing our trading strategy using month $M+1$, for $M \in \{7, 8, 9, 10, 11\}$. This designation is arbitrary, but somewhat logical within, say, a trading platform that may be updated on a monthly basis.

All models are estimated using Keras [128]. We trained with the objective of minimising validation loss and implemented an early stopping rule with a patience of 15, restoring the best model weights. We utilize the Adam optimization algorithm when training models [51]. We batch-train our models whereby each batch contains 1024 samples, with the whole year of data consisting of about 7450 batches[3]. All results were generated using Tensorflow-GPU 1.9.0 on a 6 core Xeon W-2133 3.2GHz / 12GB NVIDIA GTX 1080 ti / 64GB RAM machine. Of important note, we trained the MLP and CNN-LSTM Inc with diagonal variance-covariance matrices to compare to the proposed 'full' variance-covariance case. Also, for practicality, we tuned an L_2 weight regularisation parameter, and selected a uniform dropout rate, by grid search over the validation data, for the MLP with diagonal covariance model. Hence, these parameter values were respectively set to 1e-8 and 0.1 for each of the four models[4]. Other key details of a linear Bayesian baseline model, that we offer as a relevant benchmark when discussing out-of-sample trading performance, are included in Appendix B.

In the left-hand table of Figure 4.1, we see that the validation data loss is uniformly lower for CNN-LSTM Inc relative to the MLP model, on average differing by a factor of

[3]We collected 25GB of raw data in .csv files for our target contracts, and then prepared just over 50GB in .h5 files at conclusion of preprocessing (and accounting for windows of input data).

[4]Where applicable, dropout layers are included as per [63]; for convolutional and recurrent layers we follow [65] and [64], with the exception that we add no dropout within the inception layer. We calculate 30 dropout samples for each prediction. Further, we found that trading strategy performance showed negligible difference using 60 samples.

2.0. On a mean-square error basis, the CNN-LSTM Inc outperforms the MLP for 3 out of 5 months for a one-tailed t-test of least MSE, with p-values ≤ 0.0001 in each case. On average across all months, the CNN-LSTM Inc MSE is 2.5% smaller. The outperformance of the CNN-LSTM Inc model in these respects supports the claim of the usefulness of this model for the prediction task.

Next, we discuss the results for the trading strategy described in Section 3.2.3, focussing on the monthly and cumulative Sharpe performance on out-of-sample data. Firstly, for all strategies, we note that we have defined a trading threshold of 0.1bps such that any trade entered must have a predicted (absolute) asset price change greater than or equal to the threshold. This threshold was chosen to be equal to the cutoff parameter defined in Section 3.2.1. Next, we define a strategy called *Base* that enters a trade position $\alpha \in \{-1, 1\}$ depending on the sign of the predicted price change (respectively, lower or higher, and also corresponding to short or long). We also analyse a trading strategy that takes the notion of prediction uncertainty into account as proposed in Section 3.2.3, and explore 2 additional substrategies whereby we estimate uncertainty by alternative methods. For the strategy *Rlsd vol*, the realised volatility of the input data window is used as a proxy for uncertainty and serves as an alternative benchmark. For *Alea*, aleatoric uncertainty is instead used, and for *Al+Ep*, we use the sum of aleatoric and epistemic uncertainty as per Section 3.2.2. We do not present results for an analogous strategy based on epistemic uncertainty alone, on finding cases of unreasonably large α corresponding to an uncertainty estimate for a particular prediction close to zero.

The out-of-sample Sharpe ratios are shown for the performance of each model by investment sizing strategy in Figure 3.4. For comparison, we show the results for both the diagonal (top row) and full (bottom row) variance-covariance model variants. For three of the four subtables, we notice that the cumulative 5-month Sharpe performance is greatest for the *Al+Ep* strategy, with *Al* second greatest in each case (with outperformance between 0.7 and 2.2%). For the fourth subtable, *Al* is greatest and marginally ahead of *Al+Ep* (by 0.7%). More strikingly, the relative outperformance across *Al+Ep* strategies compared to

Figure 3.4: Monthly Sharpe performance for MLP and CNN-LSTM Inc models for a set of investment sizing strategies, on out-of-sample data. Cumulative 5-month Sharpe is shown in the last row of each table. Investment size is scaled by realised window volatility (*Rlsd vol*), the aleatoric uncertainty estimate (*Alea*), and the sum of aleatoric and epistemic uncertainty (*Al+Ep*). *Base* has no investment sizing by uncertainty.

		MLP				CNN-LSTM Inc			
		Rlsd vol	Base	Alea	Al+Ep	Rlsd vol	Base	Alea	Al+Ep
	7	-	-	-	-	-	-	-	-
Diagonal	8	0.87	0.81	0.88	**0.89**	**1.01**	0.88	0.97	0.98
$\boldsymbol{\Sigma}$	9	1.99	**2.32**	1.98	2.00	1.39	1.64	**1.86**	1.81
	10	1.76	**2.47**	1.96	1.98	1.75	**2.45**	2.25	2.23
	11	**1.33**	1.14	1.21	1.21	**1.31**	1.20	1.23	1.25
	12	2.69	2.90	2.98	**2.99**	2.57	**2.99**	2.95	2.94
	Cuml	1.29	1.40	1.42	**1.44**	1.24	1.27	1.37	**1.38**
	7	-	-	-	-	-	-	-	-
Full	8	0.89	0.83	0.95	**0.96**	0.95	0.82	1.01	**1.02**
$\boldsymbol{\Sigma}$	9	1.84	**2.19**	2.12	2.12	1.65	1.85	**2.17**	2.15
	10	1.71	**2.47**	1.84	1.88	1.73	**2.46**	2.45	2.41
	11	**1.34**	1.12	1.24	1.24	**1.27**	1.15	1.18	1.20
	12	2.65	2.87	2.87	**2.87**	2.57	**2.94**	2.91	2.91
	Cuml	1.25	1.38	1.39	**1.42**	1.27	1.26	**1.42**	1.41

Base ranges from 2.9 to 12.7%, and we observe a minimum 11.3% outperformance relative to *Rlsd vol)*. These results offer a preliminary demonstration of the utility of incorporating estimates of both aleatoric and epistemic uncertainty in the context of investment sizing within our high-frequency trading setting.

Of interest, we highlight what on one hand appears to be the outperformance of the diagonal MLP *Al+Ep* strategy versus any other, and on the other hand the apparent strong performance of the Bayesian OLS *Base* strategy as shown in Appendix B in Figure B.1. To rationalise this, we first consider the sample paths of returns as in the left-hand panel of Figure 3.5. We observe that the out-of-sample daily cumulative (basis point) return of the presented strategies appear mostly without outliers and exhibit relative smoothness, and we do not attribute any outperformance due to any obvious lucky or outsized earnings.

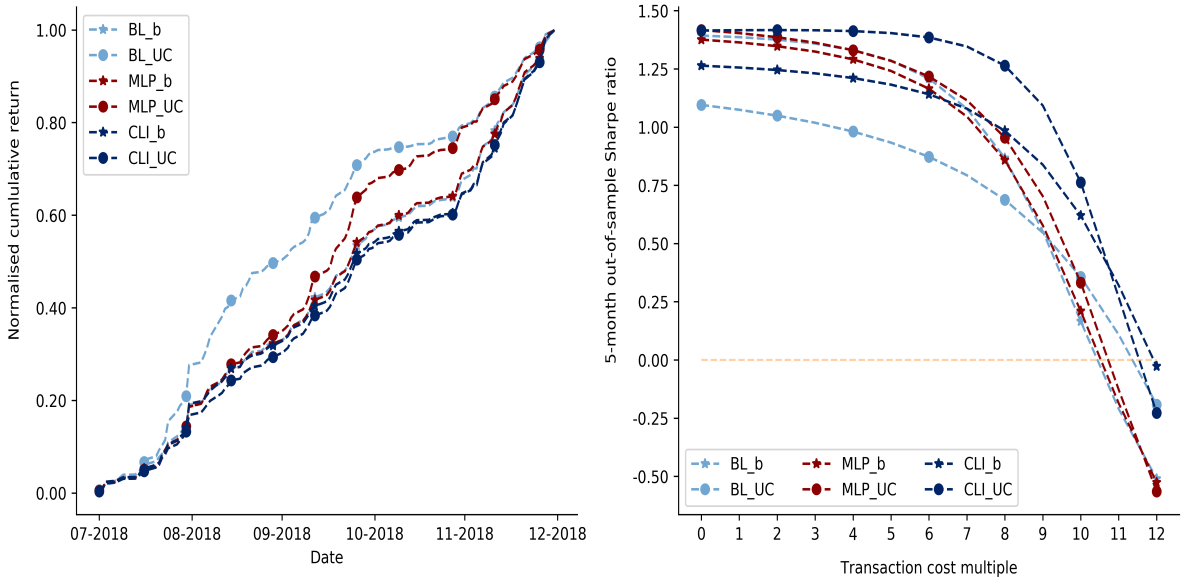

Figure 3.5: Left: Out-of-sample normalised daily cumulative basis point return for *Base* and *Al+Ep* long/short trading strategies for the baseline, MLP and CNN-LSTM Inc models. Right: 5-month out-of-sample Sharpe ratio for *Base* and *Al+Ep* long/short trading strategies for the baseline, MLP and CNN-LSTM Inc models, accounting for transaction costs. The x-axis varies by multiple of transaction costs, where a multiple of '1' represents a cost of $\frac{1}{20}^{\text{th}}$ of the trading threshold of 0.1bps.

Instead, we next consider the stylised impact of transaction costs for comparing model and strategy outperformance.

Firstly, we define an element of total net transaction cost to be equal to 0.005bps per unit volume. Based on our trading threshold of 0.1bps, this corresponds to 5% of the minimum prediction size such that $|\alpha_t| > 0$ for any trade. We accrue the transaction cost in proportion to volume transacted at each time step. To be clear, we only trade the delta-adjustment to achieve the required position of size α_t for the specific asset, relative to the existing position on the day's most recent trade time. From the right-hand panel of Figure 3.5 we see 5-month out-of-sample Sharpe ratio for *Base* and *Al+Ep* long/short trading strategies for the linear baseline, MLP and CNN-LSTM Inc models, accounting for multiples from 0 to 12 of our defined element of transaction cost. To be clear, the label '0' denotes the absence of transaction costs, and the upper label of '12' is the minimal

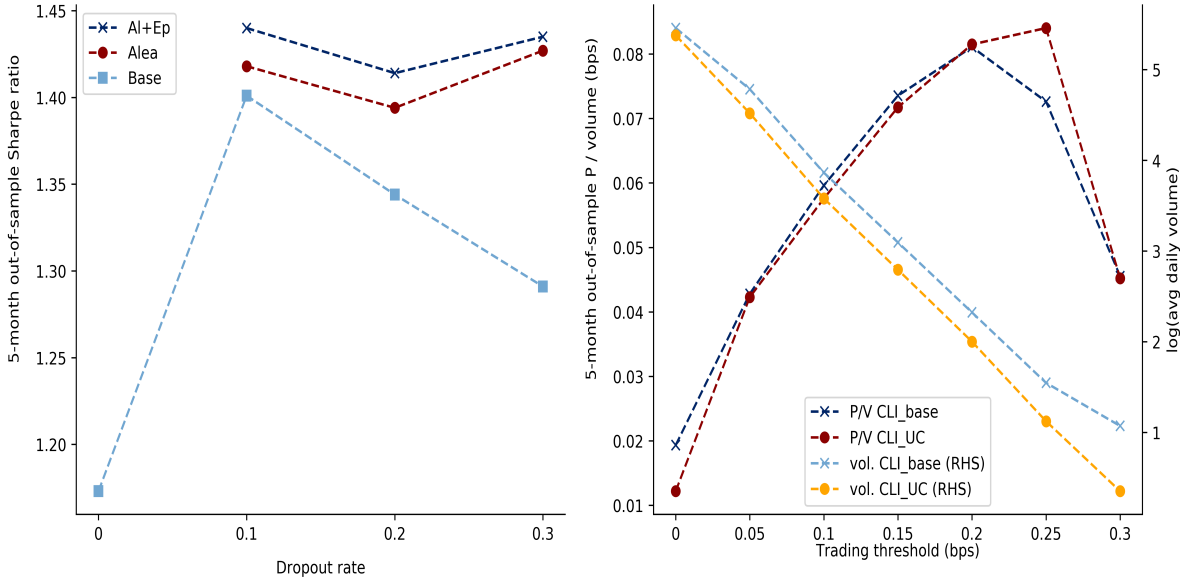

Figure 3.6: Left: 5-month out-of-sample Sharpe ratio for trading strategy based on long/short positions without uncertainty (*Base*), with aleatoric uncertainty only (*Alea*), and using our full uncertainty estimate (*Al+Ep*), for varying dropout rates, given a diagonal covariance MLP model. Right: 5-month out-of-sample profit over volume (left-hand axis) for varying trading thresholds (x-axis) for the full covariance CNN-LSTM Inc model. The right-hand axis the log average daily volume traded for each trading threshold.

integral multiple of transaction costs such that the Sharpe ratios for each presented strategy is negative. From this Figure it is immediately clear that the CNN-LSTM Inc model with uncertainty outperforms all other models at most levels of transaction costs, and that the outperformance is greatest at the middle of the range. This analysis contributes toward understanding the relative utility between alternative models, and strongly vindicates CNN-LSTM Inc with uncertainty for trading. Of course, the careful reader may notice that the transaction costs analysed here are significantly less than the median daily bid-ask spread of 0.5bps quoted in Section 3.2.1. With respect to this observation, it is important to understand that the approach presented may not necessarily be traded outright as a standalone strategy in practice. The utility of the approach may be derived from its addition to a broader market making or execution strategy, where trading is occurring anyway. Hence a deeper understanding of the impact of transaction costs, especially relative to the

bid-ask spread, may depend on a more realistic and sophisticated trading framework. We leave these considerations for future work.

Other points of interest around our backtest include observing how the choice of dropout rate or trading threshold impacts out-of-sample performance. In the case of varying the dropout rate, for the diagonal MLP, we tested cumulative 5-month Sharpe for varying dropout rates for *Base*, *Al+Ep* and *Al*, and chart the results in the left-hand panel of Figure 3.6. Interestingly, we see that for either of the two trading strategies utilizing uncertainty, the Sharpe ratios are relatively stable, ranging by less than 1.8% each across our three tested non-zero dropout rates. However, similar stability is clearly not evident in the case of the *Base* strategy, which achieves peak Sharpe performance at the choice of dropout rate of 0.1, but displays sharp decrease in performance for increasing dropout rates, and when dropout is not utilized at all. One may wonder to what extent this trading performance stability for varying dropout rates might be observed in increasingly sophisticated models such as CNN-LSTM Inc. If so, this could dominate the need for a large degree of tuning of the dropout rate when estimating the model – this is an increasingly resource consuming task with such model complexity. We leave this as a consideration for future work.

Next, we consider the case of varying the trading threshold, and refer to our results in the right-hand panel of Figure 3.6. Here we see 5-month out-of-sample profit over volume (left-hand axis) for varying trading thresholds (x-axis) for the full covariance CNN-LSTM Inc model. This statistic can be alternatively recognised as the breakeven net transaction cost for a strategy, with higher values corresponding to strategies with greater average net profit per trade. It is interesting to note that we can optimize the threshold to maximize the Sharpe ratio, though we leave further consideration of this to the interested practitioner. For added interest, we also plot the log average daily volume traded for each trading threshold on the right-hand axis and see that it is approximately log linearly decreasing with an increasing threshold.

Finally, by the right-hand panel of Figure B.1 in Appendix B, we plot estimated model reward-risk ratio versus realized monthly Sharpe on out-of-sample data, for the diagonal

MLP. In that this plot is sufficiently representative of each of the four models, we infer that despite the obvious underperformance in the tails (where sample sizes are relatively small anyway) the models appear somewhat sensibly calibrated, though we leave any other tuning in this respect as future work.

3.4 Conclusion

We present a model for price change prediction of the Eurodollar Futures curve, as a function of a recent history of asset price data, within a high-frequency domain. In doing so, we extend the existing state-of-the-art model deep learning architecture of [11] to multivariate, correlated price data, and improved price prediction relative to benchmark models, for small time horizons and in a regression setting. Importantly, we develop a preliminary analysis describing the estimation of deep learning model-generated uncertainty for financial prediction, and further showed how these uncertainties can be useful for investment size scaling within trading strategies. There is future work that could improve on this analysis, with much of it potentially important for applications at an industrial scale. With respect to data, a larger subset of available Eurodollar futures contracts could be used, as could alternative, correlated asset prices or economic information. One could also include deeper order book data, such as Level 2 quotes. With respect to the models used, there is more to be done in optimally engineering model architecture for aleatoric uncertainty estimation. Alternative methods for estimating epistemic uncertainty could be explored. On one hand, concrete dropout could be attempted for automating the tuning of the dropout rate [129]. On the other, Hamiltonian Monte Carlo methods could be explored as an alternative to estimating epistemic uncertainty, and potentially applied, for example, with `hamiltorch` [130]. Finally, with respect to the realities of implementing a trading strategy in practice, we allude to three main avenues of further research; a deeper appreciation of the impact of transaction costs beyond our preliminary analysis, the return to fine-tuning the dropout rate if performance metrics around trading are relatively stable, and finally further consideration of model calibration.

Chapter 4

View fusion for investing

4.1 Introduction

The Black-Litterman model extends the Markowitz portfolio optimisation framework to incorporate investor beliefs about asset returns [12]. Such beliefs are termed ‘views’. In its original formulation, the Black-Litterman framework is such that a given view induces an update of a prior assumption about the mean parameter of a Gaussian return distribution. Making use of views in this way can result in non-trivial differences to the estimated weights of a portfolio allocation strategy, and hence the relative investment outcome.

Views can be obtained from diverse sources, including market pundit opinions, Reserve Bank statements, and as output of statistical models. Consequently, in practice, a portfolio manager or trader may possess multiple views about the same underlying asset, or subset of assets, over the next investment period. A contribution of this work is to both consider and extend the Black-Litterman framework to incorporate multiple such views.

It is natural to formalise elements of the Black-Litterman model with respect to ideas from Bayesian statistics. See [102] for a review summary, with [103, 105] intriguing supplements. We note that the modelling of this paper presupposes the canonical Bayes-based Black-Litterman formalism. This approach is useful for practitioners in allowing for a strong theoretical grounding of the model prior in the Capital Asset Pricing Model (CAPM) [15], and for the relative ease with which one can incorporate market views. Specifically, views are given as a function of asset mean return estimates, and accompanied by an uncertainty estimate that encapsulates a notion of confidence about that mean. The specification of the view uncertainty is a domain for debate in the literature. In contrast, the uncertainty specification is of critical importance to the mechanics of the Bayesian estimation framework. The literature often tacitly assumes that uncertainty is to be specified

in excess of the given mean estimate. A prevailing method is to set the uncertainty term in proportion to a quantity such as an estimate of the underlying return covariance. The proportionality constant can then be tuned akin to a model hyperparameter. Such ad hoc specification can be unsatisfying; fortunately, scientific modelling offers an alternative.

Machine learning methods are enjoying a resurgence as oracles for financial markets, particularly given the advances underlying modern deep learning. Indeed, such methods have utility for forecasting time series, and predictions are often naturally partnered with uncertainty estimates. Such uncertainties are typically categorised in the literature as *aleatoric* or *epistemic*. Aleatoric uncertainty relates to the modelled random error of the underlying data generating process, while epistemic uncertainty encompasses much more, including error relating to the model specification, which also subsumes uncertainties about parameter estimates. These two categories of uncertainty are recognised across multiple research domains, and the models contained therein. In any case, the estimation of views by statistical forecasting models, coupled with the uncertainty estimate, allows for a more principled approach to setting view parameters when modelling with Black-Litterman. We describe this further in Section 4.3 below. This interpretation stands in stark contrast to existing methods – with one popular technique described above – that are more ad hoc. Our approach has the further benefits of preserving the intent of the canonical model, and has utility for automating an aspect of the investment decision-making process.

Further our extended modelling framework, we consider combining multiple views for expected returns over the next time increment with respect to view correlation that may be non-zero. Although we consider the single-step case, we are inspired by the relationship between Black-Litterman and a state space modelling framework as recently described in [104]. Indeed, at the core of our approach are techniques from the information fusion literature, which allow us to combine views in a principled way. In Section 4.3, we present methods for fusing views assuming both zero and non-zero correlation between them, and consider a conservative fusion method for sets of views that may be inconsistent with one other (according to some distance metric between distributions). In their relative simplicity,

the principles of these fusion methods are understandable, and the resulting fused estimate are explainable. Some practitioners may find this advantageous relative to alternative black box methods. In our empirical work, we find fusion-based methods that improve on of the mean and median of our 3 view-generating models in a statistically meaningful way. This is interesting to the investment practitioner that cannot know ex-ante which of a set of views might outperform; a fusion-based method may be selected instead.

To show the utility of data fusion, and in connection with recent work by [105], we offer an application of the Black-Litterman model assuming a multi-factor model for asset returns. Factor models present in finance with various specifications; see [131] for a recent review. We consider a conditional factor model, as inspired by the Arbitrage Pricing Theory (APT) of Ross [10]. In Section 4.2, before we further develop view fusion in Section 4.3, we expound on the Black-Litterman-APT as a Bayesian hierarchical model; a schematic for the model is given in Figure 4.1 below. In Section 4.2.3 we carefully derive the solution for the optimal portfolio weights under an unconstrained Markowitz portfolio allocation as a further contribution to the literature. With a desire to extend our empirical work, we offer a preliminary model accounting for transaction costs, with details specified in Section 4.2.4 This model permits a reasonable proof-of-concept, which could be built upon for industrial application or as future academic work.

Notes for the data set construction are offered in Section 4.4. To the extent the raw data is available in the public domain (via the Wharton Research Data Service), these notes could lead to a straightforward replication. In Section 4.5, we use the ideas and results of the preceding sections to give empirical results that showcase view fusion whereby estimates – including measures of uncertainty – are sourced from a collection of machine learning models. We discuss the ways in which fusion-based methods lead to investment outperformance relative to methods based on a singular view. We also contrast our methods with the difficult benchmark of a buy-and-hold investment in the S&P 500. We conclude and summarise avenues for future work in Section 4.6.

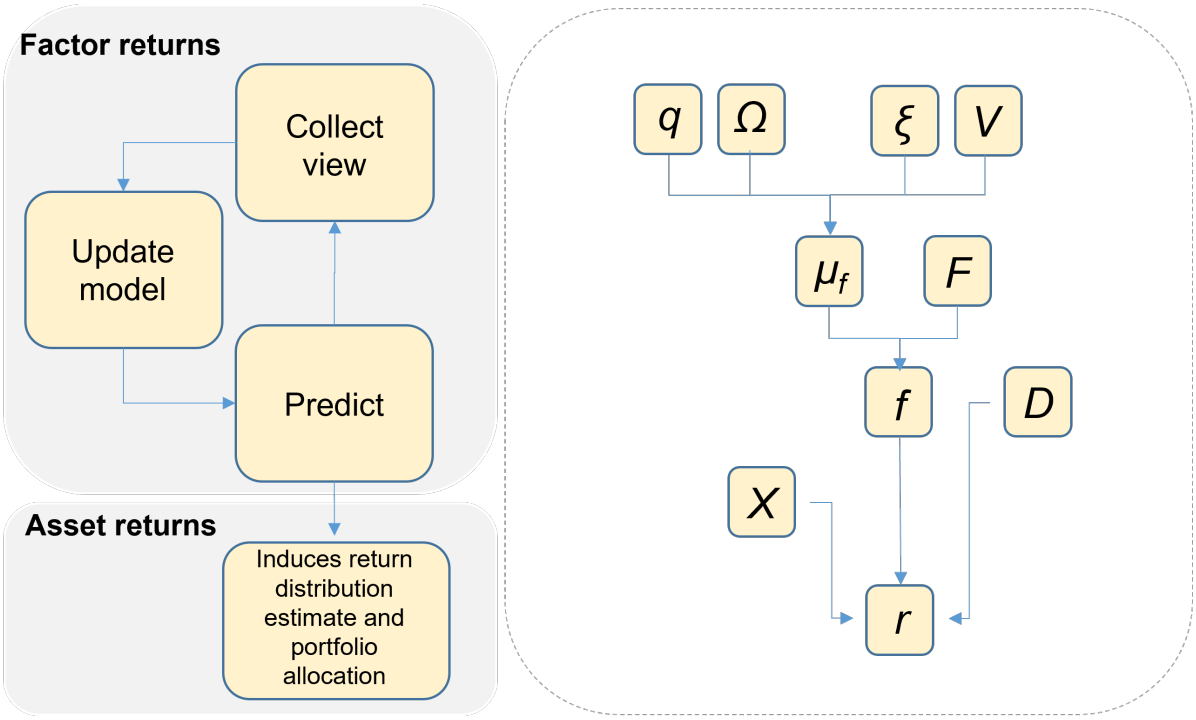

Figure 4.1: LHS: Schematic of the Black-Litterman methodology as it applies to the APT model. Asset returns are linearly projected onto factor returns, and the algorithm at the level of the factor model is such that at each time-step views are collected, before the factor returns model is updated and predictions gleaned. RHS: Hierarchical schematic suggestive of the relationship between terms in the BL-APT model, as described in Section 4.2.

4.2 Black-Litterman

4.2.1 A recap

Assume there exist n investable assets within a market with *excess* return vector $\boldsymbol{r}$ whose distribution is modelled as multivariate normal over discrete time increments, denoted $\boldsymbol{r} \sim \text{MVN}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. Here $\boldsymbol{\mu}$ denotes the mean vector of returns, and $\boldsymbol{\Sigma}$ denotes the return covariance. Then in a Markowitz Modern Portfolio Theory framework [3], assuming a single-period investment setting, the investor portfolio weights $\boldsymbol{w} = (w_1, ..., w_n)$ are a solution to the mean-variance optimisation problem:

$$\max_{\boldsymbol{w} \in \mathcal{C}} \quad \mathbb{E}(\boldsymbol{w}'\boldsymbol{r}) - \frac{\gamma}{2}\text{var}(\boldsymbol{w}'\boldsymbol{r}). \tag{4.1}$$

Here $\mathcal{C}$ is a set of weight constraints, if any, and $\gamma > 0$ is the investor risk aversion parameter, denoting preferences balancing return and risk. In the unconstrained case, differentiating Equation (4.1), setting it to zero and solving for the optimal portfolio weights $\boldsymbol{w}^*$ in terms of $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ yields

$$\boldsymbol{w}^* = (\gamma \boldsymbol{\Sigma})^{-1} \boldsymbol{\mu}. \tag{4.2}$$

On analysing the second derivative one finds that this solution is indeed a maximum. In practice, these weights could be solved for – and a portfolio subsequently implemented – upon estimation of the return distribution parameters, for example, upon substitution of a computed sample mean vector and sample covariance matrix, given historical data.

The Black-Litterman model [12] extends this framework to incorporate (subjective) investor beliefs about asset returns, that are called 'views'. A single view is assumed to be a function of real-valued vectors of length n denoting portfolio weights $\boldsymbol{p}$ and expected asset returns $\boldsymbol{\mu}$, a real-valued scalar denoting the portfolio value q, and a strictly positive real-valued scalar ϵ denoting uncertainty in the portfolio value, that can be written

$$\boldsymbol{p}' \boldsymbol{\mu} = q + \epsilon. \tag{4.3}$$

To be explicit, let us call a collection of k views a 'range', for natural numbers k. Then similar to the original presentation of Black and Litterman, a range of views can be arranged thus:

$$\boldsymbol{P}\boldsymbol{\mu} = \boldsymbol{q} + \boldsymbol{\epsilon}, \qquad \boldsymbol{\epsilon} \sim N(0, \boldsymbol{\Omega}). \tag{4.4}$$

Here $\boldsymbol{P} \in \mathbb{R}^{k \times n}$ is a real-valued matrix of portfolio weights, where the k^{th} row is given by $\boldsymbol{p}_k'$ as defined above; $\boldsymbol{q} \in \mathbb{R}^k$ is a real-valued vector collecting k portfolio values; and $\boldsymbol{\Omega} \in \mathbb{R}^{k \times k}$ is assumed to be a diagonal matrix of portfolio value estimate uncertainties, where we have further assumed independence between views.

Given the above formulation, a view can express both 'absolute' and 'relative' relationships between mean asset returns. The case of an absolute (normalised) view corresponds to $\boldsymbol{p}$ taking a unit value at the i^{th} entry, and zero otherwise, so that the left-hand side of

(4.3) equals μ_i. Any other (non-trivial) view we call relative; a common relative view, for example, is a belief on the return spread $\mu_i - \mu_j = q$. This could be expressed by setting alternate positive and negative unit values at the i and j index of $\boldsymbol{p}$, and zero elsewhere. In the work that follows, we assume absolute views. We further assume that $\boldsymbol{P}$ is of full rank.

Given these preliminaries, the Black-Litterman model is such that the existing return distribution parameter assumption is updated given views. We understand such updating in the context of Bayesian statistics, as prior authors have noted [102, 103, 105, 104].

The Bayesian specification assumes a prior distribution $p(\boldsymbol{\theta})$, where $\boldsymbol{\theta}$ denotes the target parameter of the return distribution that we wish to express via a parametric probabilistic model. Most commonly, $\boldsymbol{\theta} = \boldsymbol{\mu}$. The original authors – and many since – set the prior given an estimate of the CAPM model [15], and hence underwriting the prior with a foundational theoretical result.

Next, the view is related to the target parameter via a likelihood function $p(\boldsymbol{q}|\boldsymbol{\theta})$, such that, by the laws of conditional probability and Bayes' Theorem, an updated posterior distribution for the target parameter is given by

$$p(\boldsymbol{\theta}|\boldsymbol{q}) \propto p(\boldsymbol{q}|\boldsymbol{\theta})p(\boldsymbol{\theta}). \tag{4.5}$$

Finally, we can write a posterior predictive distribution for the asset returns given the views as

$$p(\boldsymbol{r}|\boldsymbol{q}) = \int p(\boldsymbol{r}|\boldsymbol{\theta})p(\boldsymbol{\theta}|\boldsymbol{q})d\boldsymbol{\theta}. \tag{4.6}$$

Hence, the expectation and covariance of the random quantity $\boldsymbol{r}|\boldsymbol{q}$ can be computed analytically – or otherwise estimated if the integral is intractable – before substitution into, say, Equation (4.2) to yield a solution to the portfolio allocation problem.

4.2.2 Black-Litterman in the context of Factor models

In line with recent work by [105], we offer an application of the Black-Litterman model assuming a factor model for asset returns, and we conveniently conform with some of their notation.

For context, we note that in many investment studies of equity markets the empirical estimation of the return covariance is challenging, especially when the universe of equities under consideration is large. The linear factor model that we outline below, whereby $k \ll n$, is relatively practical and efficient. This has contributed to its popularity within the literature, and bolsters the case for industrial use [132]. We consider a conditional factor model, as inspired by the Arbitrage Pricing Theory (APT) of Ross [10]. In the single-period investment setting, we model

$$\boldsymbol{r} = \boldsymbol{X}\boldsymbol{f} + \boldsymbol{\epsilon}_r, \qquad \boldsymbol{\epsilon}_r \sim N(0, \boldsymbol{D}). \tag{4.7}$$

Here $\boldsymbol{r}$ is the n-dimensional random vector containing the cross-section of excess returns for n equities over the next time increment $[t, t+1]$; $\boldsymbol{X}$ is a non-random $n \times k$ matrix of observable factor exposures estimated at t, for k the number of explanatory asset factors; $\boldsymbol{D}$ is a covariance matrix for the returns, also known at t and assumed to be diagonal; and $\boldsymbol{f}$ is a k-dimensional latent factor vector that we estimate at each time step by cross-sectional regression. Within a Black-Litterman framework, consider that our parameter of interest is now $\boldsymbol{\theta} = \boldsymbol{\mu_f}$, such that

$$\boldsymbol{f} = \boldsymbol{\mu_f} + \boldsymbol{\epsilon}_f, \qquad \boldsymbol{\epsilon}_f \sim N(0, \boldsymbol{F}). \tag{4.8}$$

We refer to the elements of $\boldsymbol{\mu_f}$ as the factor risk premia.

Black-Litterman in the context of APT, henceforth labelled ‘BL-APT’, amounts to a Bayesian hierarchical model. A schematic for the model is given in the right-hand panel of Figure 4.1. We note that in comparison to the Black-Litterman model applied to the return mean, an analogue to the CAPM prior does not exist. In the work that follows, we set the prior parameters as a simple function of recent data, per the discussion in Section 4.5. On the other hand, view estimates for the factor risk premia at time t induce return forecasts (with uncertainties) over the next time period. These parameter estimates can then be used to estimate portfolio weights.

In the next section, we carefully show the derivation for the optimal portfolio weights of an unconstrained ‘BL-APT’ as a reference contribution to the literature.

4.2.3 Solving for optimal weights – Hierarchical Bayesian Black-Litterman for APT

We can express our model for asset returns, factor returns and views carefully as a hierarchical Bayesian model, with a suggestive schematic shown in Figure 4.1 above. We derive the optimal portfolio allocation after finding $p(\boldsymbol{r}|\boldsymbol{q})$, as defined below.

We begin by writing the joint distribution of random variables of interest as a conditional probability:

$$p(\boldsymbol{r}, \boldsymbol{f}, \boldsymbol{\theta}, \boldsymbol{q}) = p(\boldsymbol{r}, \boldsymbol{f}|\boldsymbol{\theta}, \boldsymbol{q})p(\boldsymbol{\theta}, \boldsymbol{q}).$$

Substituting $p(\boldsymbol{\theta}, \boldsymbol{q}) = p(\boldsymbol{\theta}|\boldsymbol{q})p(\boldsymbol{q})$ and dividing both sides by $p(\boldsymbol{q})$ yields

$$p(\boldsymbol{r}, \boldsymbol{f}, \boldsymbol{\theta}|\boldsymbol{q}) = p(\boldsymbol{r}, \boldsymbol{f}|\boldsymbol{\theta}, \boldsymbol{q})p(\boldsymbol{\theta}|\boldsymbol{q}).$$

Integrating both sides over $\boldsymbol{\theta}$ we find that

$$p(\boldsymbol{r}, \boldsymbol{f}|\boldsymbol{q}) = \int \underbrace{p(\boldsymbol{r}, \boldsymbol{f}|\boldsymbol{\theta}, \boldsymbol{q})}_{=p(\boldsymbol{r}, \boldsymbol{f}|\boldsymbol{\theta})} p(\boldsymbol{\theta}|\boldsymbol{q}) d\boldsymbol{\theta}.$$

Again, integrating both sides, this time with respect to $\boldsymbol{f}$, we have that

$$p(\boldsymbol{r}|\boldsymbol{q}) = \iint p(\boldsymbol{r}, \boldsymbol{f}|\boldsymbol{\theta})p(\boldsymbol{\theta}|\boldsymbol{q}) d\boldsymbol{\theta} d\boldsymbol{f}. \tag{4.9}$$

With one more simplification, writing $p(\boldsymbol{r}, \boldsymbol{f}|\boldsymbol{\theta}) = p(\boldsymbol{r}|\boldsymbol{f}, \boldsymbol{\theta})p(\boldsymbol{f}|\boldsymbol{\theta}) = p(\boldsymbol{r}|\boldsymbol{f})p(\boldsymbol{f}|\boldsymbol{\theta})$, we find

$$p(\boldsymbol{r}|\boldsymbol{q}) = \int p(\boldsymbol{r}|\boldsymbol{f}) \underbrace{\int p(\boldsymbol{f}|\boldsymbol{\theta})p(\boldsymbol{\theta}|\boldsymbol{q}) d\boldsymbol{\theta}}_{=p(\boldsymbol{f}|\boldsymbol{q})} d\boldsymbol{f}. \tag{4.10}$$

Since we assume $\boldsymbol{\theta} = \boldsymbol{\mu}_{\boldsymbol{f}}$, there holds the well-known facts that the posterior $p(\boldsymbol{\theta}|\boldsymbol{q})$ is normal and that the posterior predictive $p(\boldsymbol{f}|\boldsymbol{q})$ is normal. Hence it is clear that $p(\boldsymbol{r}|\boldsymbol{q})$ is also normal, with its expectation and covariance given in closed-form by the following three-step process.

Firstly, choosing the prior $p(\boldsymbol{\theta}) \sim N(\boldsymbol{\xi}, \boldsymbol{V})$, by Bayes' rule for linear Gaussian systems, as in [133], we have that the posterior $p(\boldsymbol{\theta}|\boldsymbol{q})$ is multivariate normal with covariance

and mean

$$\begin{aligned}\text{var}(\boldsymbol{\theta}|\boldsymbol{q}) &= (\boldsymbol{V}^{-1} + \boldsymbol{\Omega}^{-1})^{-1},\\ \mathbb{E}(\boldsymbol{\theta}|\boldsymbol{q}) &= \text{var}(\boldsymbol{\theta}|\boldsymbol{q}) \cdot (\boldsymbol{V}^{-1}\boldsymbol{\xi} + \boldsymbol{\Omega}^{-1}\boldsymbol{q}).\end{aligned}$$

Secondly, recall the convolution integral for sums of random variables: if $\boldsymbol{X}_i \overset{i.i.d.}{\sim} N(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i), i = 1, 2$, then

$$f(\boldsymbol{z}) := \int f_{\boldsymbol{X}_1}(\boldsymbol{x}) f_{\boldsymbol{X}_2}(\boldsymbol{z} - \boldsymbol{x}) d\boldsymbol{x} \sim N(\boldsymbol{\mu}_1 + \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2). \tag{4.11}$$

Writing the inner integral of Equation (4.10) with the change of variable $\boldsymbol{f} \mapsto \boldsymbol{f} - \boldsymbol{\theta}$, we find

$$p(\boldsymbol{f}|\boldsymbol{q}) = \int \underbrace{p(\boldsymbol{f} - \boldsymbol{\theta}|\boldsymbol{\theta})}_{\sim N(0, \boldsymbol{F})} p(\boldsymbol{\theta}|\boldsymbol{q}) d\theta$$

so that by Equation (4.11) we have that $p(\boldsymbol{f}|\boldsymbol{q})$ is normally distributed with expectation and covariance given by

$$\begin{aligned}\mathbb{E}(\boldsymbol{f}|\boldsymbol{q}) &= (\boldsymbol{V}^{-1} + \boldsymbol{\Omega}^{-1})^{-1} \cdot (\boldsymbol{V}^{-1}\boldsymbol{\xi} + \boldsymbol{\Omega}^{-1}\boldsymbol{q}) = \mathbb{E}(\boldsymbol{\theta}|\boldsymbol{q}),\\ \text{var}(\boldsymbol{f}|\boldsymbol{q}) &= (\boldsymbol{V}^{-1} + \boldsymbol{\Omega}^{-1})^{-1} + \boldsymbol{F} = \text{var}(\boldsymbol{\theta}|\boldsymbol{q}) + \boldsymbol{F}.\end{aligned}$$

Thirdly, we solve the outer integral of Equation (4.10) (deferring some of the details of the algebraic manipulations to the Appendix, for brevity). Writing

$$p(\boldsymbol{r}|\boldsymbol{q}) = \int p(\boldsymbol{r}|\boldsymbol{f}) p(\boldsymbol{f}|\boldsymbol{q}) d\boldsymbol{f}, \tag{4.12}$$

we collect and expand the terms in the exponent up to a factor of $-\frac{1}{2}$, complete the square in $\boldsymbol{f}$, and integrate what is recognisable as a Gaussian integral. By the recursive nature of the integrals, we also retrieve a Gaussian for $\boldsymbol{r}|\boldsymbol{q}$. Hence, as supported by further calculations within the Appendix below, we have that the posterior predictive distribution $p(\boldsymbol{r}|\boldsymbol{q})$ is multivariate normal with covariance and mean

$$\begin{aligned}\text{var}(\boldsymbol{r}|\boldsymbol{q}) &= \left[\boldsymbol{D}^{-1} - \boldsymbol{D}^{-1}\boldsymbol{X}\left[\boldsymbol{X}^T\boldsymbol{D}^{-1}\boldsymbol{X} + \text{var}(\boldsymbol{f}|\boldsymbol{q})^{-1}\right]^{-1}\boldsymbol{X}^T\boldsymbol{D}^{-1}\right]^{-1},\\ \mathbb{E}(\boldsymbol{r}|\boldsymbol{q}) &= \text{var}(\boldsymbol{r}|\boldsymbol{q})\boldsymbol{D}^{-1}\boldsymbol{X}\left[\boldsymbol{X}^T\boldsymbol{D}^{-1}\boldsymbol{X} + \text{var}(\boldsymbol{f}|\boldsymbol{q})^{-1}\right]^{-1}\text{var}(\boldsymbol{f}|\boldsymbol{q})^{-1}\mathbb{E}(\boldsymbol{f}|\boldsymbol{q}).\end{aligned}$$

By Equation (4.2), we write the optimal weights for a single-step Black-Litterman portfolio allocation as

$$\begin{aligned}\boldsymbol{h}^*_{blb} &= \gamma^{-1}\text{var}(\boldsymbol{r}|\boldsymbol{q})^{-1}\mathbb{E}(\boldsymbol{r}|\boldsymbol{q}) \\ &= \gamma^{-1}\boldsymbol{D}^{-1}\boldsymbol{X}\big[\boldsymbol{X}^T\boldsymbol{D}^{-1}\boldsymbol{X} + \text{var}(\boldsymbol{f}|\boldsymbol{q})^{-1}\big]^{-1}\text{var}(\boldsymbol{f}|\boldsymbol{q})^{-1}\mathbb{E}(\boldsymbol{f}|\boldsymbol{q}). \end{aligned} \tag{4.13}$$

This expression differs meaningfully with other optimal weights given in the literature. This is the case in that we have carefully accounted for all of the specified terms within the hierarchy of the returns model.

4.2.4 An optimal weight model under transaction costs

In this section we consider a more realistic objective function for solving for portfolio weights, in that it incorporates transaction costs. We do this by making a minimal adjustment to the Markowitz objective of Equation (4.1):

$$\max_{\boldsymbol{w}\in\mathcal{C}} \qquad \mathbb{E}(\boldsymbol{w}'\boldsymbol{r}) - \frac{\gamma}{2}\text{var}(\boldsymbol{w}'\boldsymbol{r}) - TC, \tag{4.14}$$

with TC denoting transaction costs measured in percentage of invested wealth. Of the many ways to account for transaction costs, we broadly follow the single-period model recently presented in [98]. Despite the differences in our application domains, our analysis and results contrast somewhat consistently, and certainly interestingly, with theirs.

Firstly, we choose a transaction cost model inspired by [94] so that transaction costs measured in dollars, TC^d, is given by

$$TC^d_t = \frac{1}{2}\boldsymbol{\Delta}'_t\boldsymbol{\Lambda}_t\boldsymbol{\Delta}_t.$$

Here $\boldsymbol{\Delta}_t$ is the portfolio turnover vector for a time increment, defined as

$$\boldsymbol{\Delta}_t := \Pi_t(\boldsymbol{w}_t - \boldsymbol{R}_{t-1}\boldsymbol{w}_{t-1}),$$

and the 'market impact' vector $\boldsymbol{m}$ is given by

$$\boldsymbol{m}_t = \frac{1}{2}\boldsymbol{\Lambda}_t\boldsymbol{\Delta}_t.$$

We use the subscript t to explicitly account for time-dependence. The real scalar Π_t denotes total wealth investable at time t. Clearly, given the above specification dollar transaction-costs erode dollar returns quadratically with increasing wealth. The term $\boldsymbol{R}_{t-1}$ denotes a diagonal matrix whose diagonal elements account for the realized returns for the assets held in the portfolio at $t-1$, over the time step $[t-1, t)$, and adjusted for wealth delta. We assume that the term $\boldsymbol{\Lambda}_t$ is a diagonal matrix whose elements depend on the daily (dollar) volume traded in each underlying asset. Given the recent estimates of [99], we assume a 10-basis point market impact when trading 1% of the daily dollar volume $\boldsymbol{L}_t$ of US equity underlyings whether long *or short*, so

$$\begin{aligned} \boldsymbol{m}_t = 10\text{bps} \cdot 1 &= \frac{1}{2}\boldsymbol{\Lambda}_t \times 1\% \boldsymbol{L}_t \\ \implies \boldsymbol{\Lambda}_t &= \frac{1}{5}\boldsymbol{L}_t^{-1}. \end{aligned} \tag{4.15}$$

In the work that follows, we set the values for daily dollar volumes at their 6-month rolling average. Further, we assume that investor wealth grows proportionally with the market; specifically, we make the somewhat arbitrary assumption that at each time step the total investable capital is one-tenth of the sum of the daily dollar volumes of the assets in our trade universe, as known at the most recent time step.

Hence Equation (4.14) can be written, noting that $TC = TC^d/\Pi$, as

$$\max_{\boldsymbol{w}_t \in \mathcal{C}} \quad \mathbb{E}(\boldsymbol{w}_t' \boldsymbol{r}_t) - \frac{\gamma}{2}\text{var}(\boldsymbol{w}_t' \boldsymbol{r}_t) - \frac{\Pi_t}{2}(\boldsymbol{w}_t - \boldsymbol{R}_{t-1}\boldsymbol{w}_{t-1}^*)' \boldsymbol{\Lambda}_t (\boldsymbol{w}_t - \boldsymbol{R}_{t-1}\boldsymbol{w}_{t-1}^*) \tag{4.16}$$

and conveniently solved in closed-form for optimal weights:

$$\boldsymbol{w}_t^* = (\gamma \boldsymbol{\Sigma}_t + \Pi_t \boldsymbol{\Lambda}_t)^{-1} (\boldsymbol{\mu}_t + \Pi_t \boldsymbol{\Lambda}_t \boldsymbol{R}_{t-1} \boldsymbol{w}_{t-1}^*). \tag{4.17}$$

Hence the optimal portfolio weights for the BL-APT model can be found by substituting estimates for $\mathbb{E}(\boldsymbol{r}|\boldsymbol{q})$ and $\text{var}(\boldsymbol{r}|\boldsymbol{q})$ into Equation (4.17), as an alternative to Equation (4.13), when taking transaction costs into account.

4.3 View fusion

4.3.1 Preliminaries

The Black-Litterman model – sans solving for the portfolio weights – can be expressed via a state space representation; this was recently presented in [104] in the context of modelling temporal dependency in asset returns and parameter view estimates[1]. In a simple case, consider ranges of views generated from a source at discrete time increments $k = 0, 1, \dots$. We can infer the underlying target mean $\boldsymbol{\mu}_k$ in an online way, via the linear time-varying system of equations:

$$\boldsymbol{\mu}_{k+1} = \boldsymbol{\mu}_k + \boldsymbol{\epsilon}_k^{\boldsymbol{\mu}}, \tag{4.18}$$

$$\boldsymbol{q}_k = \boldsymbol{P}_k \boldsymbol{\mu}_k + \boldsymbol{\epsilon}_k^{\boldsymbol{q}}. \tag{4.19}$$

Here $\boldsymbol{\epsilon}_k^{\boldsymbol{\mu}}$ and $\boldsymbol{\epsilon}_k^{\boldsymbol{q}}$ model independent zero-mean white-noise processes with respective covariances $\boldsymbol{\Psi}_k$ and $\boldsymbol{\Sigma}_k$. We call Equations (5.1) and (5.2) the 'state' and 'measurement' equations, respectively. Solving the state space representation of Black-Litterman, assuming absolute views, we have the probability distribution for $\boldsymbol{\mu}_k | \boldsymbol{q}_k$, analogous to Equation (4.5) and solved equivalently to solving for $p(\boldsymbol{\theta}|\boldsymbol{q})$ per Section 4.2.1. The predictive distribution for $\boldsymbol{\mu}_{k+1} | \boldsymbol{q}_k$, analogous to Equation (4.6) can be solved equivalently to solving for $p(\boldsymbol{f}|\boldsymbol{q})$, also of Section 4.2.1. On the other hand, within the Black-Litterman literature, as in our model from Section 4.2, the prior is typically re-initialised to $p(\boldsymbol{\mu}_0)$ at each time-step, rather than dynamically updated given a sequence of view estimates. In the following work, we follow this approach, and leave consideration of prior updates based on a larger subset of recent data as future work. In any case, such state space modelling naturally inspires an extension of the Black-Litterman framework to handle the combination – or 'fusion' – of multiple views held about a particular portfolio.

[1]There exists an extensive literature on state space modelling, with applications across many domains. See, for example, [134] for an instructive review of filtering methods for mathematical finance.

4.3.2 Information fusion

To this point we have described the Black-Litterman model assuming a singular source for a range of views. Next, we consider the realistic – though in the context of Black-Litterman modelling, unexplored – case of multiple sources generating views simultaneously for each time increment, and analyse the impact on the predictive distribution. We extend Equation (5.2) thus (for brevity, keeping to the case of absolute views):

$$\boldsymbol{q}_{k,s} = \boldsymbol{\mu}_{k,s} + \boldsymbol{\epsilon}_{k,s}^{(\boldsymbol{q})}, \tag{4.20}$$

where $\boldsymbol{q}_{k,s}$ denotes the view is from source s, $1 \leq s \leq S$, and each $\boldsymbol{\epsilon}_{k,s}^{(\boldsymbol{q})}$ is a zero-mean white-noise process with covariance $\boldsymbol{\Sigma}_{k,s}$. Multiple views could realistically arise in the context of implementing a systematic trading or asset management strategy, whereby multiple predictions across multiple (potentially, broadly diverse) models need to be fused before capital allocation.

Information fusion has been common to the state space literature since the work of [135]. However, fusion is not limited to the state space framework, and instead applies more generally to the problem of fusing multiple estimates of an underlying random variable[2]. Correspondingly, data fusion can take place at the level of the state or measurement equations, though we assume the former for our empirical work below.

Typically, fusion techniques amount to solving for an optimal fused estimate $(\widehat{\boldsymbol{\mu}}, \widehat{\boldsymbol{\Sigma}})$ for the true underlying values $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ given a collection of S underlying estimates $\{(\widehat{\boldsymbol{\mu}}_s, \widehat{\boldsymbol{\Sigma}}_s) : 1 \leq s \leq S\}$, with the mean parameter a weighted linear combination of the underlying estimates:

$$\widehat{\boldsymbol{\mu}} = \sum_s \boldsymbol{W}_s \, \widehat{\boldsymbol{\mu}}_s. \tag{4.21}$$

[2]Various approaches exist in the data fusion literature for incorporating information from multiple sources; introductory reviews can be found in [136, 137].

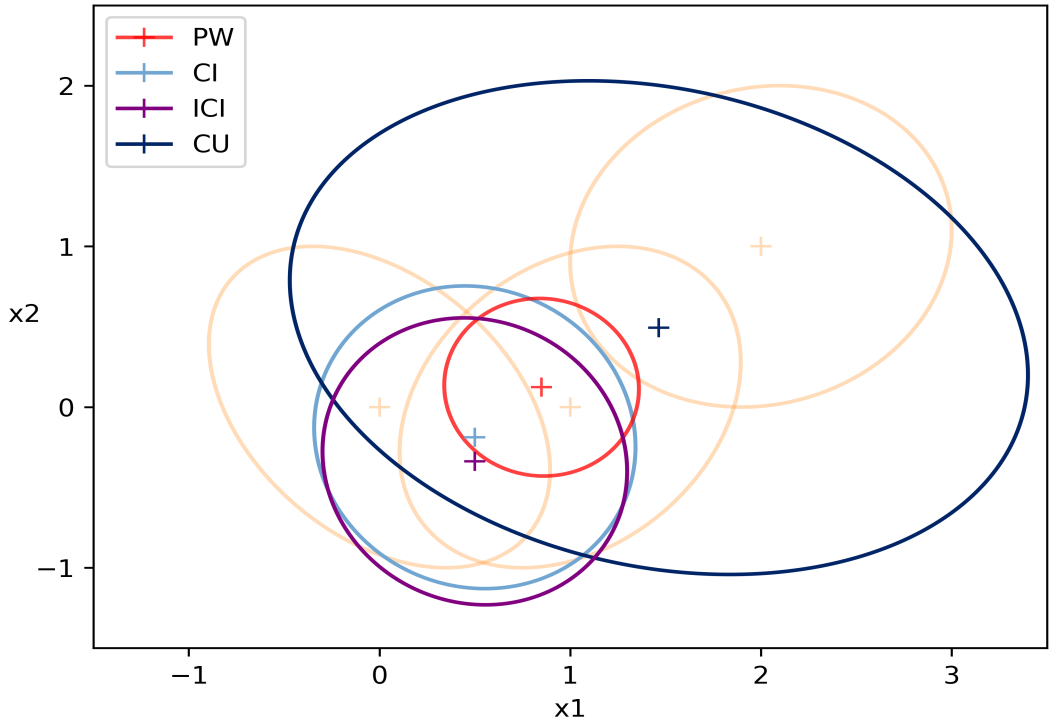

Figure 4.2: Concentration ellipses for three example bivariate Gaussian measurements of a single state $\boldsymbol{x} = (x_1, x_2)$ (light orange), and the optimal fused state estimates for the four methods outlined in Section 4.3.

(Omitting the subscript k, for ease of notation,) 'Optimal' fusion is such that (i) $\widehat{\boldsymbol{\Sigma}}$ is minimal in some sense – typically with respect to matrix trace or determinant; and also that (ii) $\widehat{\boldsymbol{\Sigma}}$ is *consistent*. Within the fusion literature, a consistent estimate is such that $\widehat{\boldsymbol{\Sigma}} \geq \boldsymbol{\Sigma}$, in the sense that $\widehat{\boldsymbol{\Sigma}} - \boldsymbol{\Sigma}$ is positive semi-definite. That is, the consistent estimate subsumes a covariance no less than the covariance $\boldsymbol{\Sigma}$ of the distribution of the true underlying process; see, for example, [138]. We note that this definition is independent of the well-known definition of a consistent estimator from the statistical literature, whereby a parameter is called consistent if it converges in probability to the true value of the parameter. The consistency property is crucial for many investors conscious of risk, particularly those that do not want to understate risk[3]. Another benefit of consistency is that probabilistic bounds

[3]In the case of (multi-period) Kelly investing, to which the fusion of this paper could directly apply, investment size is inversely proportional to variance. But the investor faces almost sure ruin if they sufficiently over-invest (which can happen if uncertainty is estimated to be too small) relative to the optimum strategy [139].

for the state of the system can be deduced, as discussed in [140].

Given the mean and covariance of a bivariate random variable, recall the 'concentration ellipse' [141] defined as the locus of points $\{\boldsymbol{x} : (\boldsymbol{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}) = 1\}$. For visual intuition, we depict such ellipses, estimated using the fusion methods described in the following subsections, in Figure 4.2 above. These results are shown for the case of three underlying data sources, assumed to each yield a noisy estimate of the same target.

Precision-weighted fusion

We first consider precision-weighted (PW) fusion for the case that the cross-covariances between sources is known to be zero. Suppose that for each s, $\widehat{\boldsymbol{\mu}}_s$ is an unbiased estimate, and each $\widehat{\boldsymbol{\Sigma}}_s$ is a known error covariance estimate that is consistent. It follows that the fused estimate given in Equation (4.21) is an unbiased estimate if and only if $\sum_s \boldsymbol{W}_s = \boldsymbol{I}$, for weight matrices $\boldsymbol{W}_s$, and $\boldsymbol{I}$ the identity matrix. To find the weight estimates we solve

$$\min_{\boldsymbol{W}} \text{ trace}(\widehat{\boldsymbol{\Sigma}}) \text{ subject to } \sum_s \boldsymbol{W}_s = \boldsymbol{I}.$$

Let $\widehat{\boldsymbol{\Sigma}}^{ij}$ denote the cross-covariance estimate between the unique sources s_i and s_j, and write

$$\widehat{\boldsymbol{\Sigma}} = \sum_s \boldsymbol{W}_s \widehat{\boldsymbol{\Sigma}}_s \boldsymbol{W}_s^T + \sum_{s_i \neq s_j} \boldsymbol{W}_{s_i} \widehat{\boldsymbol{\Sigma}}^{ij} \boldsymbol{W}_{s_j}^T. \tag{4.22}$$

Assuming the cross-covariances are everywhere zero, solving for the optimal weights one finds that, on weight substitution,

$$\widehat{\boldsymbol{\Sigma}} = (\widehat{\boldsymbol{\Sigma}}_1^{-1} + ... + \widehat{\boldsymbol{\Sigma}}_S^{-1})^{-1}, \tag{4.23}$$

$$\widehat{\boldsymbol{\mu}} = \widehat{\boldsymbol{\Sigma}}(\widehat{\boldsymbol{\Sigma}}_1^{-1}\widehat{\boldsymbol{\mu}}_1 + ... + \widehat{\boldsymbol{\Sigma}}_S^{-1}\widehat{\boldsymbol{\mu}}_S). \tag{4.24}$$

Further details of this well-known result can be found in [142, 135]. The analogous result for fusing two sources that have known, non-zero correlation is given in [143], though for our financial modelling framework this is less useful since the cross-correlations are not estimated. On the other hand, the case of unknown, potentially non-zero cross-correlation is the domain of the Covariance Intersection method, as presented in the following subsection.

Covariance Intersection

One could reasonably expect that views on the same underlying asset returns, as functions of similar or equal underlying data, exhibit non-zero cross-correlation. However, the cross-correlation is not necessarily known, or easily estimable. Such a setting is the domain of application for the Covariance Intersection (CI) fusion method of [144]. The insight of this approach is that, given Equation (4.22), the covariance ellipses for $\{\widehat{\mathbf{\Sigma}}_s\}$ contain $\widehat{\mathbf{\Sigma}}$ in their intersection, for any values of the cross-covariance terms. Hence the method yields

$$\widehat{\mathbf{\Sigma}}^{-1} = \sum_s \omega_s \widehat{\mathbf{\Sigma}}_s^{-1}, \tag{4.25}$$

$$\widehat{\boldsymbol{\mu}} = \widehat{\mathbf{\Sigma}} \sum_s \omega_s \widehat{\mathbf{\Sigma}}_s^{-1} \widehat{\boldsymbol{\mu}}_s, \tag{4.26}$$

for scalars $\omega_s \in [0, 1]$ with $\sum_s \omega_s = 1$, since a convex combination of covariances ensures $\widehat{\mathbf{\Sigma}}$ encloses the intersection region. The ω_s terms can be computed via a standard optimization routine subject to minimising the trace of $\widehat{\mathbf{\Sigma}}$. Further, it has been shown that the CI method applied to a pair of consistent estimates yields a consistent estimate [138]. Finally, Equations (4.25) and (4.26) as given above extend the canonical case of two sources, as offered in [145].

CI is a straightforward method that holds for any cross-covariance, but a cost of its flexibility is that the estimate can be too conservative for many practical use cases. We next consider Inverse Covariance Intersection that attempts to retain the consistency benefit of CI, while being less conservative.

Inverse Covariance Intersection

The method of inverse Covariance Intersection (ICI) is similar in approach to that of CI, but achieves a less conservative estimate while still maintaining consistency. The algorithm was presented recently in [146]. A central insight, inspired by [147], is to first consider

estimates pairwise, such that for $s = 1, 2$:

$$\widehat{\boldsymbol{\mu}}_s = \widehat{\boldsymbol{\Sigma}}_s\Big((\widetilde{\boldsymbol{\Sigma}}_s)^{-1}\widetilde{\boldsymbol{\mu}}_s + \boldsymbol{\Gamma}^{-1}\boldsymbol{\gamma}\Big), \tag{4.27}$$

$$\widehat{\boldsymbol{\Sigma}}_s = \Big((\widetilde{\boldsymbol{\Sigma}}_s)^{-1} + \boldsymbol{\Gamma}^{-1}\Big)^{-1}. \tag{4.28}$$

This is to say, that each estimate is itself the PW fusion of a sub-estimate from the collection $\{(\widetilde{\boldsymbol{\mu}}_s, \widetilde{\boldsymbol{\Sigma}}_s) : 1 \leq s \leq 2\}$, for sub-estimates uncorrelated between sources, and a shared set of information denoted $(\boldsymbol{\gamma}, \boldsymbol{\Gamma})$, also uncorrelated with each sub-estimate; we proceed assuming such a decomposition exist. If the shared information is known, then fusion can be solved optimally by subtracting the shared covariance from the PW fusion covariance for the original pair of estimates [146]. But since it is not known, the second insight is to instead solve for, and subtract, an upper bound for the shared covariance, valid for any true underlying shared information. Further, regardless of the shared information, it is shown that a consistent, tight covariance estimate – if it exists – is of the form

$$\widehat{\boldsymbol{\Sigma}}^{-1} = \sum_{i=1}^{2}\widehat{\boldsymbol{\Sigma}}_s^{-1} - \Big(\sum_{s=1}^{2}\omega_s\widehat{\boldsymbol{\Sigma}}_s\Big)^{-1}, \tag{4.29}$$

for scalars ω_s as in the case of CI, with the ICI covariance estimate achieving this form exactly [146]. With a modicum of algebra, it follows that the fused mean is given by

$$\widehat{\boldsymbol{\mu}} = \sum_{s=1}^{2} C_s\widehat{\boldsymbol{\mu}}_s, \tag{4.30}$$

$$\text{where } C_s = \widehat{\boldsymbol{\Sigma}}\cdot\Big(\widehat{\boldsymbol{\Sigma}}_s^{-1} - \omega_s\Big(\sum_{s=1}^{2}\omega_s\widehat{\boldsymbol{\Sigma}}_s\Big)^{-1}\Big).$$

It is important to recognise that this formula does not generalise to more than two information sources in an obvious way – the ICI algorithm fuses information sources pairwise. However, ICI can be applied recursively. This follows from the assumption that any pairwise-fused estimate has a decomposition analogous to Equations (4.27) and (4.28) [146]. For further results and details we also recommend [148].

Covariance Union

The fusion methods outlined above assume that each source's estimate is itself consistent with respect to the target. We relax this assumption by requiring that at least one of the

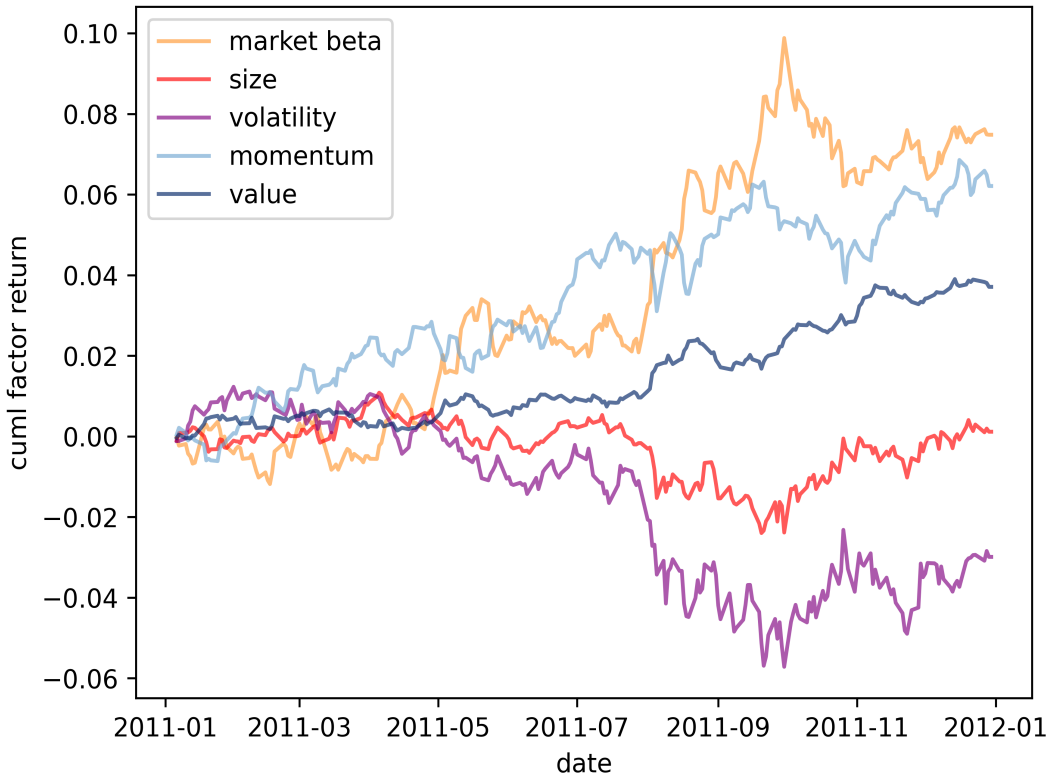

Figure 4.3: Cumulative factor returns to risk premia as described in Section 4.2.2; replication of Figure 1 of [105]. A single year is shown for ease of visualisation. The year 2011 is arbitrarily chosen.

source estimates are consistent. A given estimate could be inconsistent, as evidenced by, say, mean components sufficiently different between estimates, given relatively smaller covariance estimates. This could be apparent, for example, when the Mahalanobis distance between some pair of estimates exceeds a critical threshold. Further, we may not know precisely which of the underlying estimates is consistent, or it may be (in some sense) costly to resolve. The Covariance Union (CU) algorithm has been proposed as a fusion method for this setting [149]. CU constructs a fused estimate guaranteed to be consistent, since the fused estimate is consistent with respect to each underlying estimate:

$$\widehat{\boldsymbol{\Sigma}} \geq \widehat{\boldsymbol{\Sigma}}_s + (\widehat{\boldsymbol{\mu}} - \widehat{\boldsymbol{\mu}}_s)(\widehat{\boldsymbol{\mu}} - \widehat{\boldsymbol{\mu}}_s)^T, \qquad 1 \leq s \leq S.$$

While the idea behind CU is relatively simple, the numerical implementation for estimating the fused estimate is more challenging. In an attempt to simplify the computation, we offer the following details. Firstly, CU can be implemented using non-convex

optimisation software such as SolvOpt [150]. We follow [151] for implementing a SolvOpt-based solution for calculating the results of Section 4.5 below. Secondly, a key step is to define the coefficient vector that we wish to solve for, $\boldsymbol{x}$. The first n elements of $\boldsymbol{x}$ are the elements of $\widehat{\boldsymbol{\mu}}$, and the remaining $\frac{n(n+1)}{2}$ elements of $\boldsymbol{x}$ are the elements of the upper triangle of $\widehat{\boldsymbol{\Sigma}}$. Then $\boldsymbol{x}$ becomes the optimization target for SolvOpt. We seek to minimise the determinant of $\widehat{\boldsymbol{\Sigma}}$ subject to the constraint that

$$\forall s, \qquad \widehat{\boldsymbol{\Sigma}}_s^* := \widehat{\boldsymbol{\Sigma}} - \widehat{\boldsymbol{\Sigma}}_s - (\widehat{\boldsymbol{\mu}} - \widehat{\boldsymbol{\mu}}_s)(\widehat{\boldsymbol{\mu}} - \widehat{\boldsymbol{\mu}}_s)^T \geq 0. \tag{4.31}$$

A particular feature of the SolvOpt software is that it requires a constraint $c(\boldsymbol{x})$ such that, for any input, $c(\boldsymbol{x}) \leq 0$. Since $\widehat{\boldsymbol{\Sigma}}_s^*$ is required to be positive semi-definite, it must have non-negative eigenvalues for all choices of s, and the constraint is satisfied for a choice of c such that

$$-c(\boldsymbol{x}) := \min\{\{\text{eigenvalues}(\widehat{\boldsymbol{\Sigma}}_s^*) : \forall s\}\} \geq 0.$$

Finally, for the interested reader, the works of [151, 152] offer further efficient approximation methods for implementing CU, as required for particular practical use cases.

4.4 Data

We describe the data set construction for replicating the empirical example for BL-APT of [105]. This construction matches the original descriptions as closely as possible, and we make note of any additional required assumptions made. While we have attempted to carefully reproduce the database, our results need not match exactly. Not in the least, we have no guarantee that the underlying database that we call is the same 5 years later, even if our data replication process is perfectly faithful. A visual comparison can be made between our factor values against the source given Figure 5.3 above, which demonstrates a reasonable replication.

The data is sourced from CRSP and IBES databases with access granted via the Wharton Research Data Service. The original work sourced daily US equity market data

from 1992-2015; we extend our data set till 2022 to include the market crash of March 2020, and subsequent market rebound, for interest's sake.

For each day, set $n = 2000$ and select the top n stocks within the US market sorted by market capitalisation. The data set is filtered for USD denominated common stock only – there are no closed end funds, REITs, ETFs, unit trusts, depository receipts, warrants etc. We also collected the S&P500 (excess) return time series, stock market capitalisations, and earnings per share, adjusted for splits.

The data set requires the creation of 5 risk factors: market beta, size, volatility, momentum and value. The reasonableness of our data set construction is per the output of Figure 5.3, which accords with expectations. We also create 70 industry factors; these are one-hot encoded based on the industry given by SIC. Calculating size is relatively straightforward in CRSP as the product of shares outstanding and the daily close price. Value was determined by analysing IBES data, and could be merged back to the CRSP database based on the CUSIP identifier. We note that risk factor construction is often non-trivial. Momentum was codable using CRSP data and the definitions of [153]; we calculated the daily compounded 12 month returns sans the most recent 1 month of data. We created the market Beta and volatility factors using the relevant CRSP data and the details of [105]. Indeed, over a two-year rolling window we estimate a collection of linear functions by regressing each asset's daily excess returns against the S&P500 excess returns. The market Beta is the regression gradient, while the volatility factor is set to be the regression mean-square error.

The data is constructed for semimonthly rebalancing to facilitate the experiments of the following section.

4.5 Results

View models. To proxy multiple sources of view generation, we implemented various models with utility for forecasting financial time series. Indeed, machine learning models have proven useful for not only prediction, but also for estimating prediction uncertainty.

We implement an ARIMA, boosted regression and Gaussian Process model for generating views, and calculate model-derived uncertainty estimates. Relevant details are included in the Appendix, since time series prediction modelling is not the primary focus of this paper. We denote the three view models 'ARIMA', 'Boost' and 'GP', respectively. We also implement the 4 fusion methods outlined in Section 4.3.

Parameter choices. Per Figure 4.1 and the outline in Section 4.2, the BL-APT model requires the specification of $\boldsymbol{q}$, $\boldsymbol{F}$, $\boldsymbol{\Omega}$, $\boldsymbol{\xi}$, $\boldsymbol{V}$ and $\boldsymbol{D}$. We set $\boldsymbol{q}$ as the mean prediction estimate given by a view model, with the view models as previously described. The accompanying epistemic uncertainty estimate is set to $\boldsymbol{\Omega}$, and $\boldsymbol{F}$ is set equal to the aleatoric uncertainty estimate. The prior hyperparameter $\boldsymbol{\xi}$ is estimated as an average of $\boldsymbol{f}$ over a recent rolling lookback window of the most recent 20 observations. This window size was arbitrarily decided, and is the same for all rolling windows introduced hereafter. The total variance of each underlying factor is calculated relative to out-of-sample data, and we yield an approximation to epistemic uncertainty by the estimation method as described for the ARIMA model in the Appendix below. We set $\boldsymbol{V}$ as this estimate. Our estimate for $\boldsymbol{D}$ is a diagonal covariance matrix equal to $\hat{\sigma}^2\boldsymbol{I}$, where $\hat{\sigma}^2$ is the mean square error yielded from estimating Equation (5.16), and $\boldsymbol{I}$ is the identity matrix. Finally, we set the relative risk aversion parameter to a constant value of 10.

Transaction costs. An effect of accounting for transaction costs, and carefully setting the transaction cost model parameters, is to significantly reduce portfolio turnover relative to the unconstrained Markowitz approach with costs assumed nil. Indeed, without transaction costs, it is well-known that this latter approach usually results in unrealistic and unreasonably large leverage suggested for the optimal portfolio, and high portfolio turnover. Recent results of [98] show that incorporating an appropriate transaction cost model goes some way to mitigate these limitations in empirical studies. We implement the model outlined in Section 4.2.4 above to the same ends. We set our portfolio rebalance period to be semimonthly, which is sufficiently long – given our assumed wealth process, and the potential return

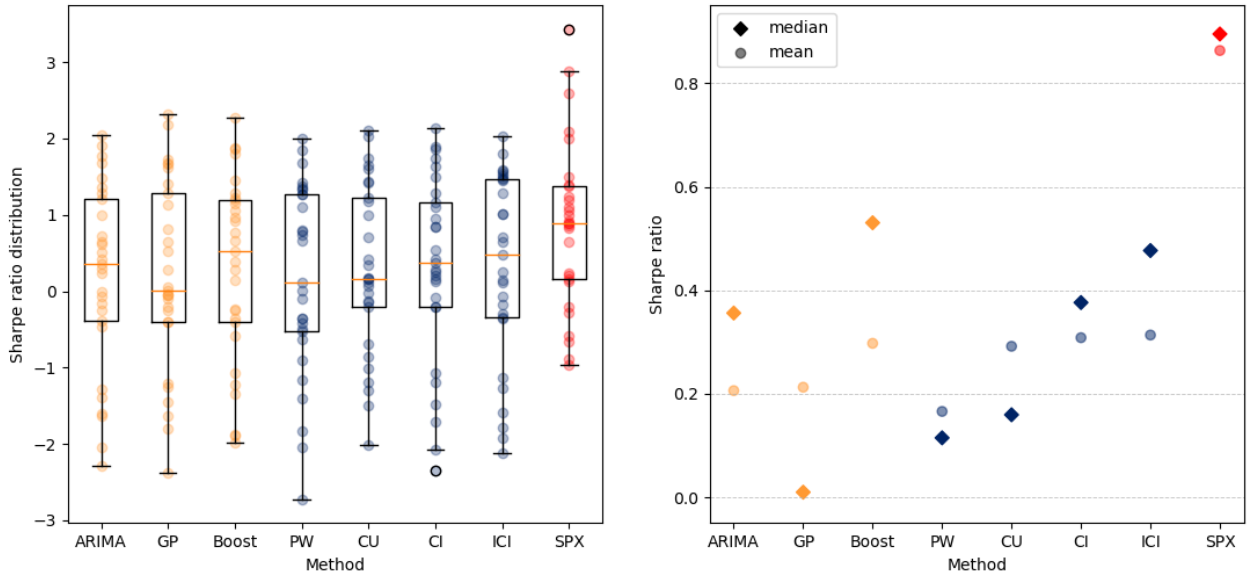

Figure 4.4: View-based portfolio excess return performance net of trading costs. Benchmark SPX strategy is gross of (negligible) transaction costs, but total returns are calculated in excess of the risk-free rate.

being significantly larger than the cost of rebalancing – such that the reward of updating weights is not dominated by the cost of trading.

Results – benchmark. We have tabulated performance results by year and model over a 29-year period, and include them in the Appendix. The method 'S&P500 (TR)' denotes the total return of the S&P 500 for the year. The annual returns may appear slightly less than commonly reported since they are net of the risk-free rate. The S&P500 (TR) represents a useful benchmark in that it represents the market buy-and-hold strategy, accessible to the modern-day investor and with a potential for exceptionally competitive fees, especially in the more recent years. For comparisons sake, all results dependent on an underlying view model are reported based on normalised underlying data, such that ex-post annualised return volatility is equal to that of the benchmark portfolio. We make a note that, on perusing the tables, there is obvious decay in the performance of the view-based models over the 30-year period, as evidenced, for example, by the significant decline in Sharpe ratios over time.

Results – statistics. We depict the Sharpe ratio statistics from the tabulated data in Figure 4.4 above. The left-most chart shows box & whisker plots for the annual Sharpe ratio for the collection of portfolios over the 29-year period. Investment in the S&P500 (TR) is clearly superior to our underlying modelling approach. On the other hand, there are improvements that could increase the view model performance that we deem beyond the scope of this work. For an easier comparison of centrality for the various method's Sharpe distributions, we show each portfolio's median and mean Sharpe ratio over the 29-year period in the right-most chart. This visual depiction summarises how fusion and non-fusion based methods performed relative to each other through time (and relative to the market total return portfolio). We observe that fusion-based medians are increasing through PW (0.11), CU (0.16), CI (0.38) and ICI (0.48), with the means showing a similar but less pronounced trend. The set of mean Sharpe ratios for the single-view based methods have a smaller range than the medians (0.21-0.30 versus 0.01–0.53, respectively); similarly for the fusion-based methods (0.17–0.31 versus 0.11–0.48). When considering the mean Sharpe ratio over the full data sample, ICI outperforms amongst the set of fusion-based methods and the single-view-based models. Similarly to the median, ICI outperforms amongst the set of fusion-based methods and sits closely to the best of the single-view-based models. We test for statistical significance of differences in these measures of distribution centrality per the results in Table 5.1 below. Despite the small sample sizes there is some interesting supporting evidence for the outperformance of fusion-based methods.

We test for the difference of means between methods with a collection of pairwise 1-sided t-tests. Such tests make a critical assumption of independence in the pairwise Sharpe ratio differences as indexed by year. We note that despite individual time series of model performance statistics potentially displaying serial correlation, it seems reasonable to suggest that the performance between models over the years does not. This is clearly a critical claim. We examine the autocorrelation function for the 28 possible pairwise comparisons and find 1 case of rejecting the null hypothesis of no autocorrelation in a Ljung-Box test at the 10% level of significance. We further assume normality of the

differences and test this with a (one-sided) Shapiro-Wilk test for normality, also at the 10% level of significance. We find that this assumption is rejected for 3 cases. Finally, we assume the non-existence of outliers in the difference data, and validate this claim on visual inspection of box and whisker plots.

On testing for the difference of means with a paired-t test, we find, of particular interest, that ICI outperforms the ARIMA and GP models, while CI outperforms the ARIMA only (though in this case the Shapiro-Wilks test leads us not to rely on some of the t-test p-values). To support testing for significance in the difference of means between groups, we bootstrap 2-sided bias-corrected and accelerated (BCa) bootstrap 90% confidence intervals for paired data [154]. We find evidence that ICI outperforms the ARIMA and GP models, and again the CI outperforms the ARIMA model. We also test for the difference in medians between groups with BCa confidence intervals, but they are broadly less conclusive. This is potentially owing to the small sample size. On the other hand, we have evidence that both ICI and CI outperform GP on this metric.

Alternatively, to testing for the difference of medians across all samples, we can test for the median of the differences between pairwise annual Sharpes. To this end, we utilise the Wilcoxon signed-rank test for paired samples [155]. In addition to assuming independence, a critical assumption is that of symmetry about the distributions of paired differences. Given the results of the Shapiro-Wilk tests, and visual inspection of what appear to be reasonable QQ plots, we continue assuming symmetry is satisfied. We perform 1-sided tests at the 10% level of significance and find evidence that ICI outperforms the ARIMA model for this comparison.

It is interesting that the CI and ICI methods achieve more compelling outperformance than the CU and PW fusion methods, relative to the underlying view models. The assumption of PW that the underlying views have 0 cross-covariance may be suboptimal, such that the method is yielding inconsistent estimates and in turn impacting trading performance. On the other hand, perhaps CU could be used more sparingly and strategically, such as when the underlying views have relatively high dispersion and so appear sufficiently

Pairwise comparison		Ljung-Box	Shapiro-Wilk	paired-t p-vals (diff of means)	BCa CIs (diff of means)	BCa CIs (diff of med.)	Wilcoxon p-vals (med. of diffs)
SPX	ICI	0.14	0.21	**0.06**	**[0.15, 0.98]**	[-0.13, 0.90]	**0.07**
	CI	0.14	0.45	**0.05**	**[0.15, 0.99]**	[-0.07, 0.70]	**0.07**
	CU	0.12	0.75	**0.05**	**[0.19, 1.01]**	**[0.31, 1.03]**	**0.06**
	PW	0.17	0.36	**0.03**	**[0.29, 1.22]**	**[0.07, 1.30]**	**0.06**
	ARIMA	0.13	0.41	**0.03**	**[0.25, 1.09]**	**[0.17, 0.92]**	**0.03**
	GP	0.17	0.51	**0.03**	**[0.20, 1.09]**	**[0.34, 1.10]**	**0.03**
	Boost	0.16	0.32	**0.05**	**[0.18, 1.01]**	[-0.10, 0.83]	**0.08**
ICI	CI	0.80	**0.03**	0.47	[-0.08, 0.08]	[-0.26, 0.28]	0.31
	CU	0.88	0.12	0.41	[-0.11, 0.14]	**[0.01, 0.67]**	0.63
	PW	0.94	0.90	0.15	[-0.03, 0.32]	**[0.04, 1.00]**	0.19
	ARIMA	0.70	0.75	**0.08**	**[0.02, 0.21]**	[-0.26, 0.36]	**0.10**
	GP	0.33	0.65	**0.09**	**[0.01, 0.19]**	**[0.25, 0.73]**	0.12
	Boost	0.68	0.19	0.46	[-0.21, 0.25]	[-0.51, 0.39]	0.58
CI	CU	0.56	0.85	0.43	[-0.12, 0.14]	**[0.05, 0.51]**	0.55
	PW	0.67	0.56	0.15	[-0.02, 0.32]	[-0.29, 0.65]	0.18
	ARIMA	0.47	**0.00**	**0.06**	**[0.03, 0.19]**	[-0.17, 0.22]	0.93
	GP	0.11	0.74	0.13	[-0.01, 0.20]	**[0.30, 0.67]**	0.16
	Boost	0.35	0.91	0.47	[-0.17, 0.24]	[-0.52, 0.10]	0.50
CU	PW	0.29	0.28	0.17	[-0.04, 0.28]	[-0.57, 0.43]	0.17
	ARIMA	0.48	0.89	0.23	[-0.05, 0.22]	**[-0.46, -0.02]**	0.20
	GP	0.75	0.22	0.23	[-0.06, 0.20]	[-0.13, 0.37]	0.21
	Boost	0.73	0.98	0.49	[-0.21, 0.19]	[-0.78, 0.01]	0.52
PW	ARIMA	0.95	0.64	0.36	[-0.17, 0.11]	[-0.65, 0.24]	0.65
	GP	0.95	0.58	0.33	[-0.19, 0.08]	[-0.31, 0.66]	0.41
	Boost	0.17	0.45	**0.09**	**[-0.27, -0.03]**	**[-1.12, -0.14]**	**0.10**
ARIMA	GP	**0.01**	0.94	0.46	[-0.08, 0.07]	**[0.22, 0.54]**	0.45
	Boost	0.20	**0.06**	0.27	[-0.29, 0.09]	[-0.51, 0.19]	0.73
GP	Boost	0.74	0.34	0.27	[-0.26, 0.07]	**[-0.96, -0.32]**	0.64

Table 4.1: Tabulated p-values and confidence intervals for a battery of statistical tests for comparing annual 29-years of annual Sharpe ratios pairwise between strategies. Yellow highlighting is applied for significance at the 10% level. The rightmost four columns represent one-sided tests.

contradictory. An alternative fusion method such as ICI could be employed otherwise.

Corroborating our intuition about the relative strength of the S&P500 (TR) strategy relative to all others, we see that SPX improves on all models – both single-view- and fusion-based – for almost all tests. Sufficiently strong view models are required to compete with the market portfolio, but beyond the scope of this work. It is perhaps surprising that view- and fusion-based models net of transaction costs have net positive Sharpe ratios at all. Indeed, we estimate global Sharpe ratios between 0.10 (PW) and 0.34 (ICI) for the entirety of the 29-year period.

Closing remark. The results presented offer preliminary evidence supporting fusion-based methods within a Black-Litterman framework subsuming a relatively complex investment strategy net of transaction costs. Fusion-based methods result in a weighted-average performance respectful of uncertainty estimates. These methods could certainly be ex-ante preferred when all views are to be incorporated into an investment decision, in the case that the best view model is unknown (and/or as in our presented work, when the method constituting the best view model is both largely unpredictable, and changing through time). We have supporting evidence for a rare 'free lunch' in financial investing by way of model fusion, essentially amounting to a benefit we could call 'model diversification'. This could be of further application in alternative trading domains, perhaps with more sophisticated underlying view models, and we hope to inspire readers to consider their practical use.

4.6 Conclusion and next steps

In this work, we further characterise view uncertainty within a Bayesian Black-Litterman framework. In doing so, we offer optimal portfolio weights for a case of Black-Litterman in an Arbitrage Pricing Theory setting. Considering the Bayesian Black-Litterman model via an equivalent state space representation, we are inspired by the information fusion literature for combining correlated prediction mean and uncertainty estimates streamed from diverse sources. Hence, we demonstrate prediction and uncertainty fusion for multiple, simultaneous views within the Black-Litterman framework, and describe consistent

uncertainty estimation within financial investing contexts. In our empirical work, we demonstrate the utility of our approach for BL-APT. Future practical work could be based on more realistic investment strategies, not the least incorporating methods of drawdown control.

Chapter 5

Noise filtering for statistical arbitrage trading

5.1 Introduction

Statistical arbitrageurs seek trading opportunities whereby an asset price (or return) is dislocated from a measure of fair value to which it is assumed it will revert. Such dislocation needs to be sufficiently large so as to be profitably traded – at least, in expectation. These strategies are documented as originating within the financial services in the late 1970s and early 1980s, within both the hedge fund and investment banking industries [156]. To this day, statistical arbitrage strategies are of popular practical application, and strategy techniques and insights have inspired a deep academic literature. A comprehensive review, for work dated up till 2016, can be found in [157].

Indeed, much of the academic literature on statistical arbitrage ultimately concerns estimating a fair value model for an asset spread or portfolio, and/or modelling and analysing the stochastic dynamics of the dislocation about fair value. This paper is mostly concerned with the former. To our knowledge, we are the first authors to present and assess a conditional factor model for estimating fair value in the context of statistical arbitrage. The model assumes that future returns are predictable given (i) the asset-wise factor exposures for the universe of assets of interest, and (ii) a latent factor vector that is estimable by regression in the asset cross-section. Further, we assume the dynamic nature of the factor vector, suggesting a state space framework for the joint modelling of factors and asset returns.

Our fair value model is closest in spirit to existing work that presents explanatory models of returns based on Principal Component Analysis (PCA) [158, 159]. In these

papers, the principal components (and their loadings) that are deemed to have sufficiently high explanatory power are an analogue to our model's factor exposures (and factor vectors). However, the papers differ significantly in many key workflow assumptions relative to ours, most importantly with respect to data set construction, asset feature modelling, and the trading strategy assumed. We posit, critically, that the details of each of these steps can have a significant impact on the estimated performance statistics. Hence we clearly describe our methods in the interest of ease of interpretability and reproducibility, and present a range of performance data based on competing sets of reasonable assumptions. Owing to transparency, we also find two key, well-known works with which we can directly compare our results; this is less commonly found possible for academic quantitative finance papers that depend on strategy backtests. In any case, we further describe these comparable papers, and other recent work in the spirit of ours, in Section 5.2 below. This summary mostly covers work not contained in the aforementioned review [157], due to their recency.

In Section 5.3, we present further the state space framework as it relates to the conditional factor model. We model returns as a noisy observation of the latent factor vectors, and describe the well-known Kalman Filter algorithm for the case of a linear model, and the Unscented Kalman Filter algorithm for the case the model specified is non-linear. In Section 5.4 we review details of the data set construction, and the factor dynamics model. We also discuss the particulars of the trading strategy, which amounts to the management of a long-short beta neutral equity portfolio. We present the specification of the trading rule, realistic transaction cost assumptions, the hedging of market risk, and the calculation of strategy returns and bankroll management.

Our results are presented in Section 5.5. We show that the conditional factor state space model is useful for trading mid-frequency statistical arbitrage in US equity markets. This is true with respect to a meaningful benchmark model and a simpler ordinary least squares-based approach, and to the results given in other well-known academic studies calculated over similar data sets. We conclude in Section 5.6, and given that the model presents as a compelling starting point for building a strategy at larger scale, we include

notes for both academic and practitioner future work.

5.2 Existing literature

We collect and describe studies relating to statistical arbitrage trading, for work mostly post the review period of [157]. Unsurprisingly, many recent methods for approaching the problem subsume modern machine learning techniques. These approaches mostly concern pairs trading. For example, surrendering performance interpretability, the work of [160] presents a spectrum of techniques – PCA for dimension reduction of raw data, clustering as a strategy for pairs identification, and forecasting with deep learning seeking better trade entry points. Similarly, [161], but without the lattermost step. Many authors appeal to dimension reduction when a large universe of portfolio assets are under consideration; this often amounts to an application of PCA. (Non-PCA based) Factor models are an alternative with a strong theoretical grounding, and are presented (amongst many other introductory ideas) in the well-known work of [162]. Such models are often characterised by a collection of hand-constructed theoretically justified features called 'factors' that to correlate to price or returns time series of the target assets. An insightful empirical demonstration of both applied PCA and factor models can be found in [158]; also of note, the paper jointly trades a multivariate portfolio of assets – beyond targeting pairs only. The approach of [163] is to model returns via a statistical factor model, with an Ornstein-Uhlenbeck process for the error term. Novelty is introduced at the level of trade execution, formulated via a stochastic control model, though only applied on synthetic data. Approaches to optimizing mean-reverting multivariate portfolios, subject to either budget or leverage constraints, and influenced by Modern Portfolio Theory, are found in [164, 165]. An approach to multivariate statistical arbitrage modelling beyond factor models is found in [166], that presents a framework for both linear and non-linear relationship modelling via vine copulas.

State space models combine knowledge of the data process with noisy measurement data; a focus of this work is to consider their utility for statistical arbitrage applications. Hence our eventual contribution – a filtering approach to multivariate statistical arbitrage,

scalable to large portfolios. Indeed, state space models for finance have been reviewed in [167, 168]. The articles include applications to stochastic volatility modelling, and the modelling of the term structures of commodity prices and interest rates. Applications to pairs trading are relevant, and were introduced by the seminal work of Elliott [169], who models a latent pair spread, and noisy observations thereof, in a state space setting. Their results are presented for synthetic data. The work of [170, 171] extends the research by introducing time-varying parameters to the model, hence improving model flexibility, and testing the theory in realistic online trading settings. The Elliott model is also generalised somewhat for richer spread dynamics in [172], which designs a novel trading strategy based on model-estimated probabilities of large dislocations mean reverting. Further generalisations are apparent in the recent article [173] which presents a quasi-Monte Carlo estimation algorithm for the state space framework presented. This allows for estimating models with richer specifications, designed to express stylized features of financial data, including non-Gaussianity and heteroskedastic variance.

It is natural to incorporate (dynamic) factor models into a state space framework, as in [174], which also models stochastic volatility within the state variable; similarly, [175]. Arguments for dynamic factor modelling for statistical arbitrage are given by [159], though the approach does not model volatility (beyond a white noise process). PCA for factor mining, and prediction (of note, at the level of asset prices, rather than returns) is a heavily exploited feature of the trading strategy.

The studies related to statistical arbitrage are summarised in Table 5.1 below, and include information on the underlying market data analysed, by asset class, time scale and date range. ‘SS’ denotes modelling in a state space framework, ‘Target’ is given by ‘pair’ for pair-trading examples, and ‘MV’ for multi-variate portfolios, while ‘Dim. red.’ denotes a dimension reduction technique is a feature of the modelling approach.

Author (Year)	SS	Target	Dim. red.	Asset class	Date range	Time scale
[157] (2017)		pair		-	-	-
[160] (2020)		pair	✓	CMDTY	2009–2019	intraday – 5min
[161] (2023)		MV	✓	Equities – Global	2011–2021	daily
[162] (2004)		pair	✓	-	-	-
[158] (2010)		MV	✓	Equities – US	1997–2007	daily
[163] (2019)		MV	✓	-	-	-
[164] (2018)		MV		Equities – US	2012–2014	daily
[165] (2019)		MV		Equities – US	2010–2014	daily
[166] (2016)		MV		Equities – US	1992–2015	daily
[169] (2005)	✓	pair		-	-	-
[170] (2011)	✓	pair		Equities – US; CMDTY	1980–2008	daily
[171] (2010)	✓	pair		Equities – US	1997–2005	daily
[172] (2016)	✓	pair		Equities – US, BR	2011–2013	daily
[173] (2021)	✓	pair		Equities – US, TW, HK	2012–2019	daily
[159] (2016)	✓	MV	✓	Equities – US	1989–2011	daily

Table 5.1: Literature themed around statistical arbitrage, toward applications of state space models for multivariate portfolios. Note 'CMDTY' is used to abbreviate the word 'Commodities'.

5.3 A dynamic model of returns and factors

State space modelling. Consider the case of discrete time increments $k = 0, 1, ...$ Then a linear Gaussian state space model is given by the time-varying system of equations:

$$\boldsymbol{x}_{k+1} = \boldsymbol{A}_k \boldsymbol{x}_k + \boldsymbol{\epsilon}_k^{\boldsymbol{x}}, \tag{5.1}$$

$$\boldsymbol{y}_k = \boldsymbol{B}_k \boldsymbol{x}_k + \boldsymbol{\epsilon}_k^{\boldsymbol{y}}. \tag{5.2}$$

Here $\boldsymbol{\epsilon}_k^{\boldsymbol{x}}$ and $\boldsymbol{\epsilon}_k^{\boldsymbol{y}}$ model independent zero-mean white-noise processes with respective covariances $\boldsymbol{\Psi}_k$ and $\boldsymbol{\Omega}_k$, assumed to be known functions of time. The matrices $\boldsymbol{A}_k$ and $\boldsymbol{B}_k$ are also assumed to be known functions of time. We call Equations (5.1) and (5.2) the 'state' and 'measurement' equations, respectively. The state $\boldsymbol{x}_k$ is assumed to be a latent random variable, of which $\boldsymbol{y}_k$ denotes a noisy measurement at time k. Given the system parameters, probabilistic beliefs about the state can be inferred, and these beliefs can be updated given measurement data.

Kalman filter. By the classic Kalman filter (KF) algorithm, given an initial state $\boldsymbol{x}_0$, and the state and measurement equations above, the filter distribution for $\boldsymbol{x}_k|\boldsymbol{y}_{1:k}$ at time

$k > 0$ can be shown to be $\sim MVN(\boldsymbol{\mu}_k^f, \boldsymbol{\Sigma}_k^f)$ where

$$\boldsymbol{\mu}_k^f = \boldsymbol{\mu}_k^p + \boldsymbol{G}_k(\boldsymbol{y}_k - \boldsymbol{B}_k\boldsymbol{\mu}_k^p), \tag{5.3}$$

$$\boldsymbol{\Sigma}_k^f = (\boldsymbol{I} - \boldsymbol{G}_k\boldsymbol{B}_k)\boldsymbol{\Sigma}_k^p. \tag{5.4}$$

These equations describe the mean and covariance in terms of the 'Kalman gain' $\boldsymbol{G}$ given by:

$$\boldsymbol{G}_k := \boldsymbol{\Sigma}_k^p\boldsymbol{B}_k'(\boldsymbol{B}_k\boldsymbol{\Sigma}_k^p\boldsymbol{B}_k' + \boldsymbol{\Omega}_k)^{-1}.$$

Also, the predictive distribution for $\boldsymbol{x}_{k+1}|\boldsymbol{y}_{1:k}$ at time k can be shown to be $\sim MVN(\boldsymbol{\mu}_k^p, \boldsymbol{\Sigma}_k^p)$ where

$$\boldsymbol{\mu}_k^p = \boldsymbol{A}_k\boldsymbol{\mu}_k^f, \tag{5.5}$$

$$\boldsymbol{\Sigma}_k^p = \boldsymbol{A}_k\boldsymbol{\Sigma}_k^f\boldsymbol{A}_k' + \boldsymbol{\Psi}_k. \tag{5.6}$$

Unscented Kalman Filter. Generalising Equations (5.1) and (5.2), we consider the non-linear system dynamics

$$\boldsymbol{x}_{k+1} = f(\boldsymbol{x}_k; \boldsymbol{A}_k) + \boldsymbol{\epsilon}_k^{\boldsymbol{x}}, \tag{5.7}$$

$$\boldsymbol{y}_k = g(\boldsymbol{x}_k; \boldsymbol{B}_k) + \boldsymbol{\epsilon}_k^{\boldsymbol{y}}. \tag{5.8}$$

Here f and g may be non-linear functions of the state and parameters. An extension of the Kalman filter was developed based on the unscented transform [176], a method for calculating statistics of a random variable that has undergone a non-linear transformation. The idea behind the algorithm – called the Unscented Kalman Filter (UKF) – is to first draw sigma points, which are sample points drawn from the support of an underlying probability distribution. The sigma points are passed through a non-linear function, so that by the unscented transform the means and covariances of the predictive and filter distributions are estimable; indeed, they are calculable as functions of (weighted) sample means and (cross-)covariances of the transformed points.

In keeping with the approach of [177], so for N the state dimension, at initialisation we define vectors of weights $\boldsymbol{w}^c, \boldsymbol{w}^m$ of size $(2N + 1)$, and select a set of sigma points

denoted $\mathcal{X}$[1]. Each sigma point is transformed on passing through f, yielding the set of transformed sigma points $\mathcal{Y} = f(\mathcal{X})$. Then the predictive distribution for $\boldsymbol{x}_{k+1}|\boldsymbol{y}_{1:k}$ at time k can be shown to be $\sim MVN(\boldsymbol{\mu}_k^p, \boldsymbol{\Sigma}_k^p)$ where

$$\boldsymbol{\mu}_k^p = \sum_{i=0}^{2N} w_i^m \mathcal{Y}_{i,k}, \tag{5.9}$$

$$\boldsymbol{\Sigma}_k^p = \sum_{i=0}^{2N} w_i^c (\mathcal{Y}_{i,k} - \boldsymbol{\mu}_k^p)(\mathcal{Y}_{i,k} - \boldsymbol{\mu}_k^p)^T + \boldsymbol{\Psi}_k. \tag{5.10}$$

Next we calculate $\mathcal{Z} := g(\mathcal{Y})$, and $\hat{\boldsymbol{y}}_k^- := \sum_{i=0}^{2N} w_i^m \mathcal{Z}_{i,k}$. Then the filter distribution for $\boldsymbol{x}_k|\boldsymbol{y}_{1:k}$ at time k can be shown to be $\sim MVN(\boldsymbol{\mu}_k^f, \boldsymbol{\Sigma}_k^f)$ where

$$\boldsymbol{\mu}_k^f = \boldsymbol{\mu}_k^p + \boldsymbol{G}_k(\boldsymbol{y}_k - \hat{\boldsymbol{y}}_k^-), \tag{5.11}$$

$$\boldsymbol{\Sigma}_k^f = \boldsymbol{\Sigma}_k^p - \boldsymbol{G}_k \boldsymbol{P}_{y,k} \boldsymbol{G}_k^T. \tag{5.12}$$

In analogy with the Kalman gain defined above, here we have $\boldsymbol{G}$ given by:

$$\boldsymbol{P}_{y,k} := \sum_{i=0}^{2N} w_i^c (\mathcal{Z}_{i,k} - \hat{\boldsymbol{y}}_k^-)(\mathcal{Z}_{i,k} - \hat{\boldsymbol{y}}_k^-)^T, \tag{5.13}$$

$$\boldsymbol{P}_{xy,k} := \sum_{i=0}^{2N} w_i^c (\mathcal{Y}_{i,k} - \boldsymbol{\mu}_k^p)(\mathcal{Z}_{i,k} - \hat{\boldsymbol{y}}_k^-)^T, \tag{5.14}$$

$$\boldsymbol{G}_k := \boldsymbol{P}_{xy,k} \boldsymbol{P}_{y,k}^{-1}. \tag{5.15}$$

A state space conditional factor model. We consider a conditional factor model, as inspired by the Arbitrage Pricing Theory (APT) of Ross [10]. For a single-period investment setting, we assume the predictive model

$$\boldsymbol{r}_{k+1} = \boldsymbol{X}_k \boldsymbol{f}_k + \boldsymbol{\epsilon}_k^r, \qquad \boldsymbol{\epsilon}_k^r \sim N(0, \boldsymbol{D}_k). \tag{5.16}$$

Here $\boldsymbol{r}_{k+1}$ is the n-dimensional random vector containing the cross-section of excess returns for n equities over the next time increment $[k, k+1]$; $\boldsymbol{X}_k$ is a non-random $n \times m$ matrix of observable factor exposures estimated at k, for m the number of explanatory asset factors; $\boldsymbol{D}_k$ is a covariance matrix for the returns, assumed to be diagonal; and $\boldsymbol{f}_k$ is

[1]There are various suggestions within the literature for how one might choose the weights and sigma points; we keep with the presentation of [177] for our implementation.

an m-dimensional latent factor vector that we estimate at each time step by cross-sectional regression.

We write the state space conditional factor model

$$\boldsymbol{f}_{k+1} = g(\boldsymbol{f}_k; \boldsymbol{\theta}_k^f) + \boldsymbol{\epsilon}_k^{\boldsymbol{f}}, \tag{5.17}$$

$$\boldsymbol{r}_{k+1} = \boldsymbol{X}_k \boldsymbol{f}_k + \boldsymbol{\epsilon}_k^{\boldsymbol{r}}, \tag{5.18}$$

assuming dynamics in the latent factor vector. The state equation implies the dynamic predictability of $\boldsymbol{f}$ via some known function g. Recently, non-linear models for factor evolution have been explored in [178].

Of course, a popular modern choice for g might be one of the various machine learning-based time series models. Such models could depend on a recent history of the state variable, rather than just the current state. In the case of the factor model above, we can expand the state space to include lags in f. Consider the one-dimensional case where

$$f_k = g(f_{k-1}, \dots, f_{k-M}; \boldsymbol{w}) + v_k.$$

The function g, for example, can be estimated by a time series model, with v_k being its aleatoric uncertainty estimate. For this single factor case, per [177], the state space model can be written:

$$\boldsymbol{f}_{k+1} = \boldsymbol{g}(\boldsymbol{f}_k; \boldsymbol{w}) + \boldsymbol{B}\boldsymbol{\epsilon}_k^{\boldsymbol{f}},$$

$$\begin{bmatrix} f_{k+1} \\ f_k \\ \vdots \\ f_{k-M+2} \end{bmatrix} = \begin{bmatrix} g(f_k, \dots, f_{k-M+1}; \boldsymbol{w}) \\ \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & \ddots & 0 & \vdots \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} f_k \\ \vdots \\ f_{k-M+1} \end{bmatrix} \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} v_k,$$

$$r_{k+1} = \begin{bmatrix} 1 & 0 & \dots & 0 \end{bmatrix} f_k + \epsilon_k^r.$$

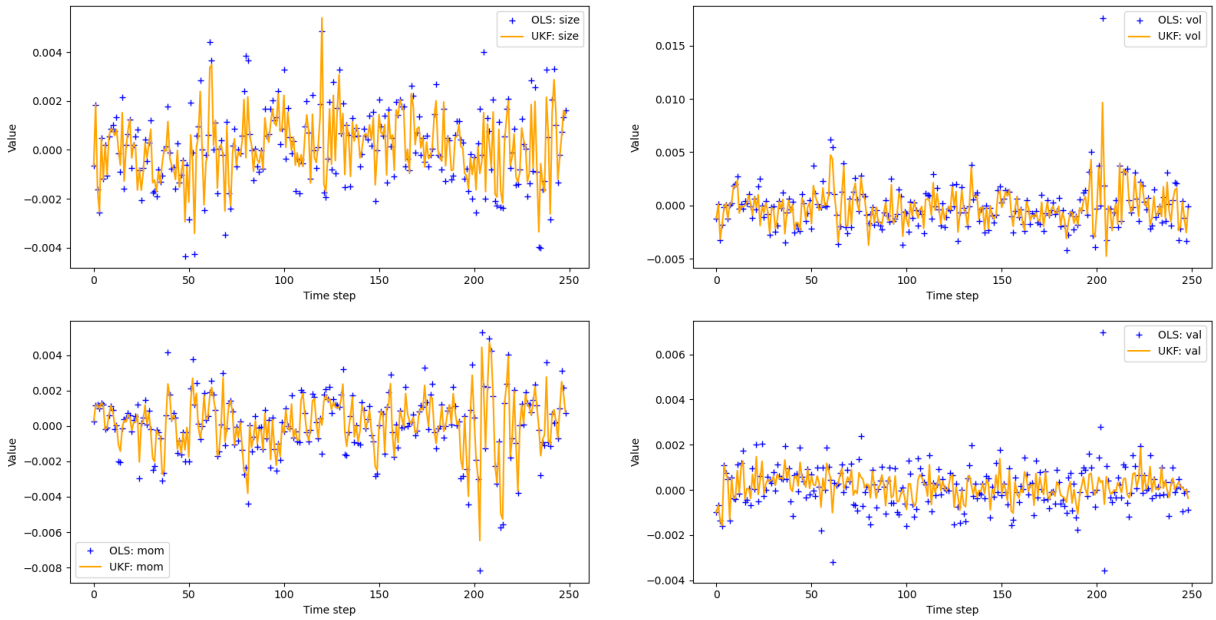

Figure 5.1: Factor risk premia as estimated by a conditional factor model using (i) ordinary least squares ('OLS'); and (ii) an underlying non-linear state space model ('UKF'). Four factors are displayed – size, volatility, momentum, and value, for the year 1997 (which was arbitrarily chosen).

We demonstrate an application of the model for four latent factors, assumed uncorrelated, showing a sample of model output in Figure 5.3 above. This compares with a predictive model for returns as a function of factors estimated by ordinary least squares. We provide further details of the model specification as part of the following section on experimental methods and strategy.

5.4 Experimental methods and strategy

Our statistical arbitrage scheme has three non-trivial components – data set construction and feature creation, data modelling, and specification of the trading strategy. Each of these components can have a significant bearing on performance results. In Section 5.3 we introduced the data model, and we present details of the particulars here. We also address the other two components in this section, commenting on the data and the trading strategy.

Data. The data set construction is the same as used in [179], with the exception that the period is daily rather than bi-monthly. From that work, recall that the data is sourced from CRSP and IBES databases, with access granted via the Wharton Research Data Service. The raw data is daily US equity market data for the years 1992-2021 inclusive. On each trading day, we select the top N greatest stocks across all available US equities sorted by market capitalisation, for $N = 2000$. The data set is filtered for USD denominated common stock only – there are no closed end funds, REITs, ETFs, unit trusts, depository receipts, warrants etc. We also collected the S&P500 (excess) return time series, stock market capitalisations, and earnings per share, adjusted for splits.

In that we experiment with factor models, the data set requires the creation of our 5 chosen risk factors: market beta, size, volatility, momentum, and value. Calculating size is relatively straightforward in CRSP: the calculation is the product of shares outstanding and the daily close price. Value was determined by analysing IBES data, and could be merged back to the CRSP database based on the CUSIP identifier. We note that risk factor construction is often non-trivial. Momentum was codable using CRSP data and the definitions of [153]; we calculated the daily compounded 12 month returns sans the most recent 1 month of data. We created the market Beta and volatility factors using the relevant CRSP data and the details of [105]. Indeed, over a two-year rolling window we estimate a collection of linear functions by regressing each asset's daily excess returns against the S&P500 excess returns. The market Beta is the regression gradient, while the volatility factor is set to be the regression mean-square error.

State space model specification. The state space models described in Section 5.3 require the specification of the state transition function g, and the state and measurement covariances, denoted $\boldsymbol{F}$ and $\boldsymbol{D}$. In the case of the linear model, we choose g to be the identity function[2], and estimate a diagonal state covariance for $\boldsymbol{f}_{k+1} - g(\boldsymbol{f}_k; \boldsymbol{\theta}^f_k)$ based on a rolling (lookback) window of size 20. For the non-linear model, we estimate a multi-layer perceptron model for g, that takes the previous history of f on a window of size 10, and

[2]The choice of g is in analogy with the state space model underlying the pairs trading application of [171].

that predicts f at the next time-step, and well as the aleatoric uncertainty of f, which we set as an estimate of $\boldsymbol{F}$. To be clear, this estimate is a weighted average of the individual uncertainty estimates collected when calculating the state update equation for each sigma point. For reference, an image of the neural network architecture is given by Figure 5.2. This model is in the spirit of [180], and in this simplified setting we implement a multi-layer perceptron with 32 nodes followed by a dropout layer with rate 0.1, and L_2 regularisation with parameter fixed at 10^{-5} chosen by cross-validation on data from the earliest years. Finally, for both the linear and the non-linear model, our estimate for $\boldsymbol{D}$ is a diagonal covariance gleaned from estimating Eq. (5.16) via ordinary least squares regression.

Benchmark models. We utilise a benchmark model with the trading strategy mechanics as outlined below, but with the key difference that the spread S is defined by the rolling 10-day sum total returns for each asset against the mean total return over a rolling 10-day window. This benchmark is discussed in [181], and is intuitive for considering reversals of returns. To be clear, this benchmark strategy is with reference to the total return series for each stock only; that is, it makes no reference to any underlying factors.

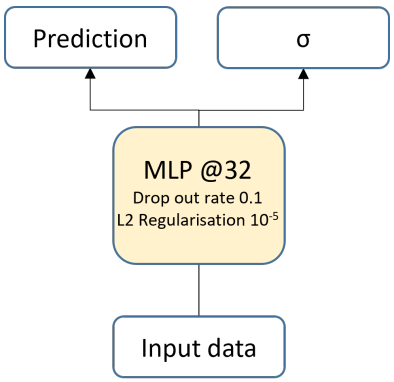

Figure 5.2: Suggestive neural network architecture for a non-linear latent factor vector state evolution model. The model outputs state prediction and aleatoric uncertainty estimates given an input historic time series.

Trading strategy. The state space model-inspired trading strategy is a long/short strategy, hedged to be beta neutral. The strategy depends on the spread defined by the rolling sum total returns in excess of the rolling sum filtered total returns – as estimated using the models of Section 3 – for each asset. That is, for each time k, r_k, X_k and X_{k-1} are observed, so the filter estimate for f_{k-1}, denoted by $f_{k-1}^{(\text{filt})}$, is calculable. Hence we can calculate $r_k^{(\text{filt})} = X_k f_{k-1}^{(\text{filt})}$. The spread, or (sum) error deviation, $\sum(r_k - r_k^{(\text{filt})})$ is the target for our trading strategy, and we implicitly assume that it is

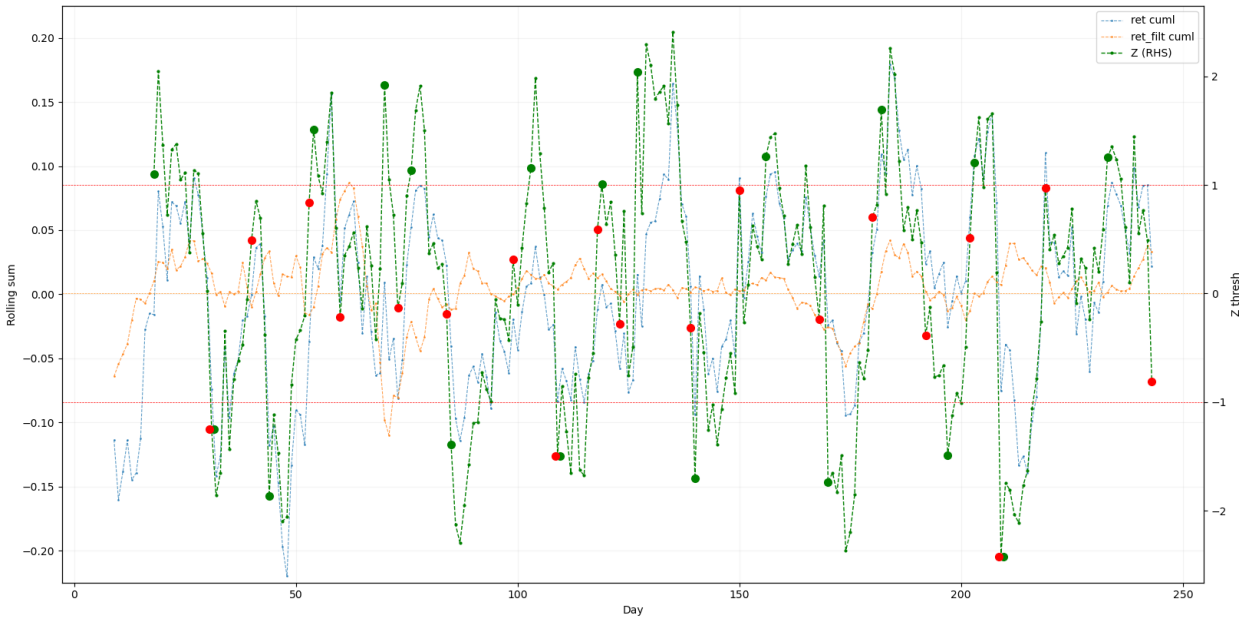

Figure 5.3: Trades are entered when the Z score deviation of the difference between returns and filtered returns passes a high (absolute) threshold; these points are indicated by the large green dots. Trades are closed at the first time the deviation has crossed the origin, as indicated by the large red dots.

mean-reverting.

Being arbitrary, the rolling window size for calculating the spread, denoted WS, is a target parameter for performance sensitivity analysis. Given a trading threshold L (S), a trade is entered long (short) the first time the Z score of spread divergences is below (above) the negative (positive) values of L (S). The trade is exited the first time the Z score reverts to, or through, the level 0, with trades 'closed out' on the last trading day of the year otherwise. The quantity L (S) should represent a sufficiently large deviation such that expected excess returns are not dominated by transaction costs. All position sizes long or short are set to a constant notional value. To assist intuition, the mechanics of the strategy are outlined diagrammatically in Figure 5.3 above, for the choice $L/S = 1.0/1.0$, for a target asset over the course of a year.

There are two other minor assumptions to the trading strategy that we state for

completeness. Firstly, when entering a trade, we assume that we can execute at the market close price. Secondly, at time t, if our database contains no close price for a stock at $t+1$, we assume that time t is the last trading day for the stock, and that this is known to market. Hence, we close any existing position in the stock at the time t close price, and do not enter a new trade at t in the event that the trade entry criteria is met.

Transaction costs. In our primary experiments transaction costs are set to 5 basis points (bps) per trade entry and per trade exit, making 10 bps for a round-trip trade in a single underlying (be it a single stock or index). This is consistent with the works of [166, 158] and chosen for ease of comparison; these works are important benchmarks for multivariate statistical arbitrage in US equities. Further, we assume that we can always borrow to short, and that the transaction costs subsume any cost of borrowing. This deviates to some extent from reality, where not all stocks can be shorted all the time, and where some stocks incur a 'hard to borrow' fee, particularly when in high demand. Unless specified otherwise, all returns discussed for stocks and indices are total returns, and short positions pay the risk-free rate assuming funding is received at the time of shorting, and able to be invested in the riskless asset. We test the robustness of our strategies to alternative levels of transaction costs; this is discussed further in the Results section below.

Hedging market risk. We hedge our trading strategy to be beta neutral. Our simple approach is to calculate the market beta of our portfolio of long and short positions each trade period, and to maintain a position of dollar value P in the S&P500 index, where

$$P = -\beta_{port}\Pi,$$

thus, providing the hedge. The term Π denotes the market dollar value of the portfolio to be hedged (and is defined further below). A positive value of P indicates a net long postion in the index as the hedge, while a negative value indicates a short position. The term β_{port} denotes the weighted-average beta of the portfolio given by

$$\beta_{port} = \frac{\boldsymbol{\beta}\boldsymbol{D}^T}{\Pi}.$$

The term $\boldsymbol{D}$ is an N-vector of position sizes whose i^{th} entry, consistent with the trading

strategy notes above, takes the value 1,-1 or 0 for a long, short, or nil position in stock i. The term $\boldsymbol{\beta}$ is an N-vector of corresponding stock beta estimates.

Implicit in the hedging strategy is the assumption that we can trade long or short positions in the S&P 500 index and earn or pay the total return of the index each period. In reality, the investor would take a position in the futures contract written on the index, which has a myriad of practical implications. Regardless, we do not consider these further, and so, for example, we ignore the spot / futures basis that could induce additional slippage.

Strategy returns. We assume a 100% leverage portfolio so that for each dollar invested in the portfolio, a position of up to a dollar's worth long and / or a dollar's worth short can be entered. Therefore, the market cost for l longs and s shorts is $\Pi = \max(l, s)$. We calculate (excess, unhedged) portfolio returns for a time period $[k, k + dt]$ thus:

$$\Delta\Pi_k/\Pi_k = \frac{\sum_i (R^l_{i,k} - r_k dt) + \sum_j (R^s_{j,k} - r_k dt)}{\Pi_k} + \frac{r_k dt S_k}{\Pi_k} - \frac{TC \cdot 1 \cdot (\Delta \boldsymbol{D}_k)^{|\cdot|}}{\Pi_k}.$$

Here $R^l_{i,k}$ ($R^s_{j,k}$) denotes the return over the period for a long (short) position held, r_k denotes the (annualised) risk-free rate of return, TC denotes the transaction cost, and $\boldsymbol{X}^{|\cdot|}$ defines an operator returning a vector of elementwise absolute values of the input $\boldsymbol{X}$. Hence the terms on the right-hand side of the summation account for, respectively, the excess return of the long and short positions, the risk free rate earned on the proceeds of the short sales deposited in a money account, and the transaction costs for changes to the portfolio holdings[3].

5.5 Results

In this section, we present performance statistics for the state space factor models of Section 5.3, relative to our benchmarks and the long-only market portfolio. For readers inclined, it is possible to replicate the results given the descriptions of the data and methodology as outlined in Section 5.4.

[3]The calculation is similar to that found in [158], and is equal when $\Pi = s$. Otherwise, our return is slightly more conservative for the case $\Pi = l$, though this has negligible impact on the final value calculated.

We find that the multivariate statistical arbitrage strategy has shown reasonable performance through time, which is consistent with the expectations for a mid-frequency strategy with reasonable capacity. Performance is best during the first half of our 29-year sample, and we conjecture that the impressive returns have diminished due to increased market competition and efficiency through time. Also, we find sufficient differences between strategy performance that supports the use of a state space conditional factor model for multi-variate statistical arbitrage, instead of other less complex - albeit relatively powerful - models. Despite all strategies showing relative underperformance in the most recent years, practitioners may find latitude for improvement by further modifying or optimizing components of the strategy; we briefly describe some preliminary ideas for research direction at the end of this section.

5.5.1 Strategy performance

The earlier years: 1993–2007. Given the data set and assumed fair value models, per the methods described in Section 5.4 there is an implied sensitivity of any estimated performance statistics to the key assumptions underlying the trading strategy. Not the least, this includes the setting of parameters denoting (i) the trade entry threshold Z score; (ii) the size of the returns window; and (iii) the level of transaction costs. We set the parameters of (i) and (ii) based on strategy performance over the 'earlier years' of 1993-2007. We assume a transaction cost level of 5 bps for each trade in all results that follow, unless otherwise specified. Choosing the parameters in this way also allows for a reasonable performance comparison between our strategy relative to results given by other well-known academic works. Finally, the parameter choices are maintained when we assess strategy aggregate performance, and performance conditioned on later years only.

For various combinations of Z score-based entry thresholds for the long and short legs (denoted L and S), and various window sizes (denoted WS), we show average Sharpe ratio and maximum drawdown statistics in Figure 5.4, on the next page. Drawdowns are calculated on a 252-day rolling window. We find utility in solving for the trade entry

parameters for the long and short legs separately, making no assumption that the same cut off should be optimal for both. We test the entry level to balance entering a sufficient number of trades, but not entering too early (or late) such that the trade is not held too long (or missed entirely). As expected, we find firstly that as the threshold decreases, the number of trades increases to the extent that small thresholds have bad economics owing to large turnover and cumulative transaction costs. Window size is similarly tested, for sizes of 1, 5 and 10, corresponding to daily, weekly, and fortnightly rolling windows.

Per Figure 5.4, we notice two clear clusters of data, whereby (i) one exhibits the highest average Sharpe ratio, but also exhibits amongst the worst for maximum drawdowns; and (ii) the other is a cluster with good average Sharpe ratio, but amongst the strategies with the better of the maximum drawdowns. It is reasonable to suggest that different investors might have different preferences for either cluster, so we extend our study for recent years with respect to them. The first cluster is characterised by models with a window size of 5, and long/short entry thresholds in $\{0.5,1.0,1.5,2.0 : L < S\}$. The second cluster is the same except that the window size is 10. For both clusters, the state space models tend to dominate both benchmark models with respect to both statistics of interest. However, the performance between the state space models is less noticeable, though is such that the KF tends to have the higher Sharpe, for worse drawdowns, relative to the UKF, and for all choices of cutoff. Regardless of an investor's preference, these differences are relatively small for this experiment.

One also notices from Figure 5.4 that the strategies that have a lower entry threshold for the longs, and relatively larger entry threshold for the shorts, tend to exhibit the better performance statistics. It is the case that the longs outperform the shorts for this data sample[4], with the shorts also more likely to have a negative year of performance. The effect of increasing an entry threshold is to reduce the number of trades entered on that long or short leg; this reduces relatively positive or negative returns, hence the observation. We leave the question of why the longs may more often outperform the shorts as future

[4]Recently, other authors have analysed long-short equity factor portfolios, albeit in an alternate strategy setting, and made the same outperformance observation [182].

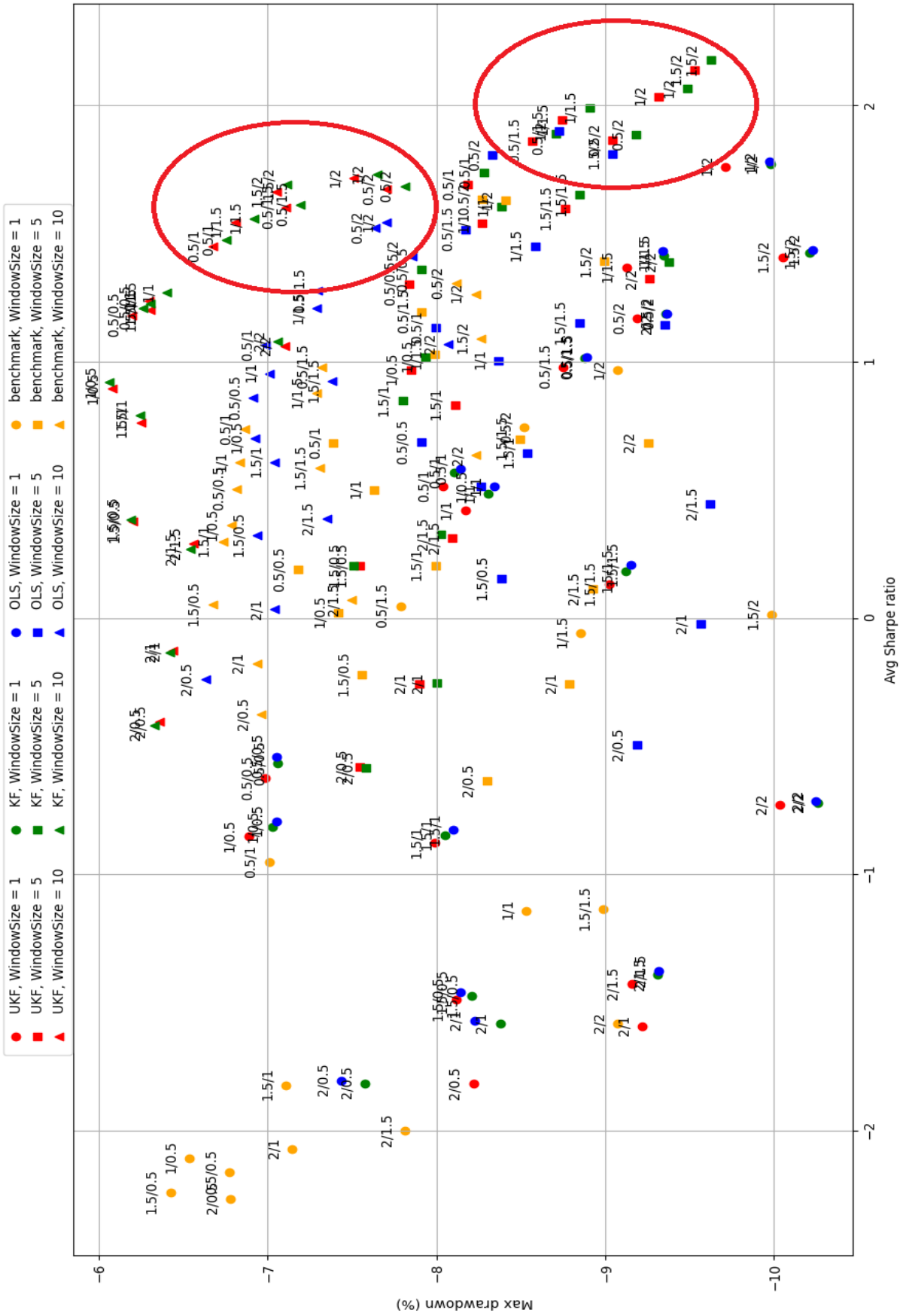

Figure 5.4: Average Sharpe ratio over the 15 years from 1993-2007 versus maximum drawdown for a collection of models underlying the trading strategy.

Table 5.2: Tabulated average annual Sharpe ratios by date range, for our conditional factor model (hedged UKF, L/S = 1.5/2.0, WS=5) and two leading external methods. Note I: All methods assume transaction costs of 5bps per trade. Note II: * results mean 'excluding 1992 due to lack of data'.

Strategy	Date range	Avg ann Sharpe [158, 166]	Avg ann Sharpe Ours	TR SPX	Notes (Models for cited work)
Ours	1993-2021	-	**1.29**	0.87	-
[158]	1997-2007	1.44	**1.61**	0.47	PCA based
	2003-2007	0.90	**1.01**	0.78	PCA based
	1997-2007	1.10	**1.61**	0.47	ETF based
	2003-2007	**1.51**	1.01	0.78	ETF based – incorporating additional volume feature
[166]	1992-2015	0.75-1.12	**1.67***	0.73*	Vine based. Range across 4 alt. trade selection methods

work, and include no assumption as to what leg may outperform for any period within our backtest; this could possibly be a source of improvement of the strategy alpha.

Our results can be reasonably compared with other well-known results from the literature. Our approach has a similar data set to the others (daily US equities data, albeit with different universe sizes), but most notably different models for fair value – in [158] factors are modelled based on Principal Component Analysis, with an additional ETF-based model, and in [166] asset relationships are modelled with Vine copulas. Our trading strategy is similar to [158], though we do not refine a trade exit threshold; we have less in common with [166], who refine their trading rules with many more steps relative to our approach. Importantly, across all papers, the transaction cost assumption is equal. For the papers cited, it is easiest to compare average annual Sharpe ratios over the date ranges given in the cited paper. We tabulate the results of this comparison in Table 5.2 above, and include the total (excess) return of the S&P 500 over the same period. Our results are similar in magnitude to [158], and we conjecture that daily US equity mean reversion could be captured by strategies using a variety of underlying fair value models over the considered date range. Superficially, there appears to be some outperformance in our approach, though this claim would benefit from comparison given outcomes of a

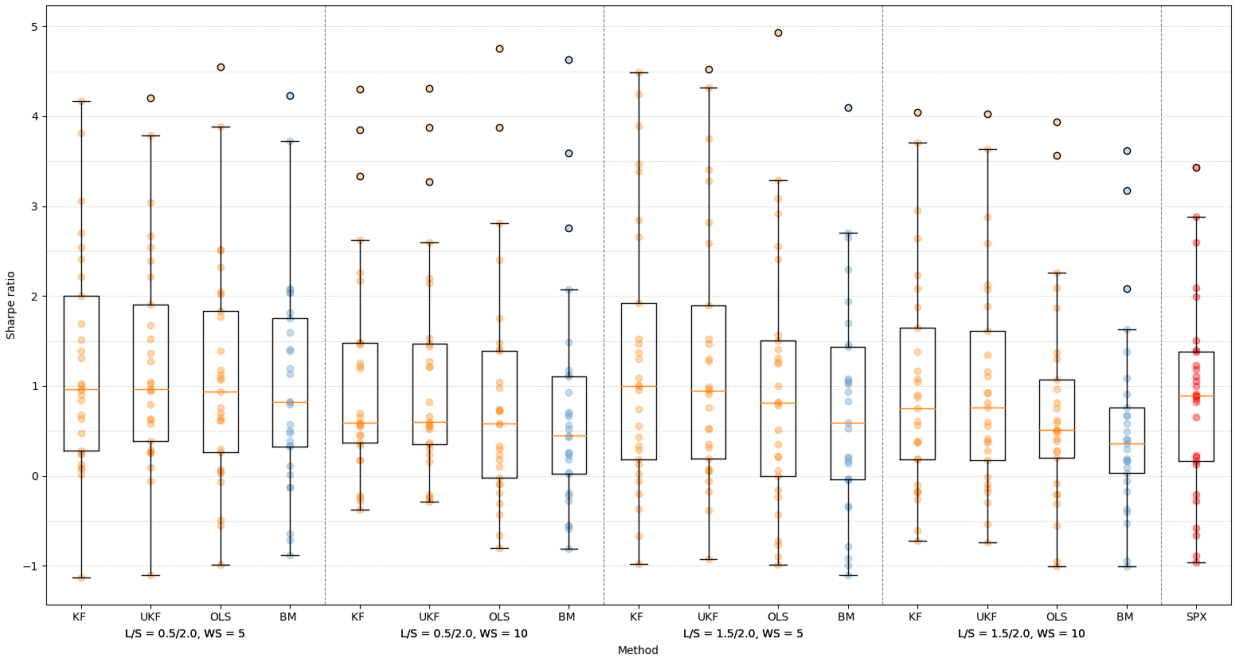

Figure 5.5: Sharpe ratio distribution over 29-years for 4 models underlying the trading strategy outlined in Section 5.3, grouped by long/short entry Z score criteria, and lookback window size for cumulative returns. The performance of the S&P 500 with respect to total (excess) return is also included for reference.

controlled experiment. On the other hand, though the average Sharpe ratios appear similar, our methods do show sufficient variation in Sharpe ratio year-on-year. For example, the PCA based method shows severe performance degradation, and negative Sharpes, in later years. On the other hand, our approach had no negative years between 1997 and 2007. The outperformance of our approach is more striking against [166], and with many degrees of freedom in the formulation of the copula-based strategy it is hard to conjecture exactly why this may be the case.

Aggregate performance. In Figure 5.5 below, we depict the annual Sharpe ratio distribution over 29-years by way of box and whisker plots, for the 4 underlying models. Further, we show the performance across 4 sets of parameter choices, which include window sizes of 5 and 10, and long/short entry thresholds of 0.5/2.0 and 1.5/2.0. We also show a plot for the Sharpe ratio of total (excess) return of the S&P 500.

Modelwise, the KF and the UKF outperform relative to OLS and BM across all but one groupwise comparisons of 1st, 2nd and 3rd quantiles, and averages, of annual Sharpe ratios. On the other hand, any UKF under/outperformance over the KF is not apparent, so that the additional computational complexity of the UKF is not justified for this particular application. If stronger non-linear relationships between variables were observed, or expected, given additional training data, the UKF may still find use. The OLS approach induces satisfactory results, with the (U)KF appearing to be a worthwhile refinement, that is, the introduction of uncertainty quantification has a benefit. The nature of the outperformance appears to vary depending on the parameter settings of the trading strategy.

In Table 5.3, we show the results of an exploration of robustness of our strategy to the transaction cost assumption. As mentioned, we implemented our strategies assuming a 5 bps (10 bps round-trip) transaction cost for each asset traded. We show Sharpe ratio percentile levels for transaction costs of 0,5,10 and 15 bps for the UKF 0.5/2.0 across window sizes. The yellow highlighted data are for the 5 bps transaction cost level. Clearly this is a sweet spot – and while the quoted performance can sustain a slightly higher transaction cost assumption, there is an obvious performance decay such that at 15 bps of transaction costs, the median annual Sharpe ratio over the sample realises negative. Finally, we should mention that the cost of short selling may have been significantly more than estimated in this work, if the capacity to borrow to sell was even available, particularly so for the earlier years of study. On the other hand,

		Transaction costs (bps)			
		0	5	10	15
WS 5	25th	1.11	0.39	-0.08	-0.76
	50th	1.39	0.99	0.57	-0.09
	75th	2.75	1.91	1.14	0.70
WS 10	25th	0.80	0.35	-0.10	-0.66
	50th	1.09	0.59	0.26	-0.17
	75th	1.90	1.47	1.05	0.83

Table 5.3: Sharpe ratio percentiles by transaction cost and window size levels for UKF 0.5/2.0.

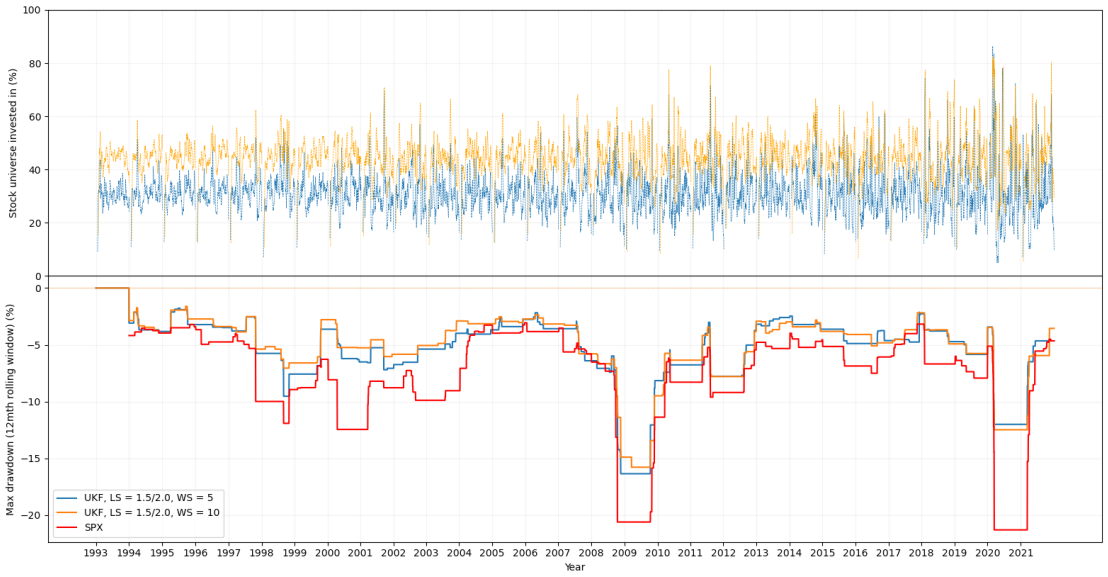

Figure 5.6: Top: Percentage of the stock universe (N=2000) invested in, long or short, at a given time. Bottom: Strategy rolling window (252-day) drawdown versus the long only market portfolio; transaction cost = 5bps.

practitioners may have strategies for cheapening transaction costs when trading. Given these latter points, and other cited academic studies, we think our primary transaction cost assumption is reasonable.

The bottom panel of Figure 5.6 shows rolling 252-day drawdown for 2 UKF based strategies differing on window size. In the years up till 2007 the strategies largely avoid the worst of the drawdowns realised by the market portfolio, which has drawdowns touching -10% or worse on 3 occasions. In these cases, our strategies realise drawdowns between -7 and -5 %, except for the strategy with window size of 5 that realises a -9.5% drawdown on one of the three occasions. Post-2007, the two severe market drawdowns of 2008 and 2020, both being worse than -20%, are matched by strategy drawdowns about -16 and - 12%, respectively. The first of these drawdowns, within the years of the Great Financial Crisis, was to the chagrin of many real-world strategies, such that we find it less of a concern that

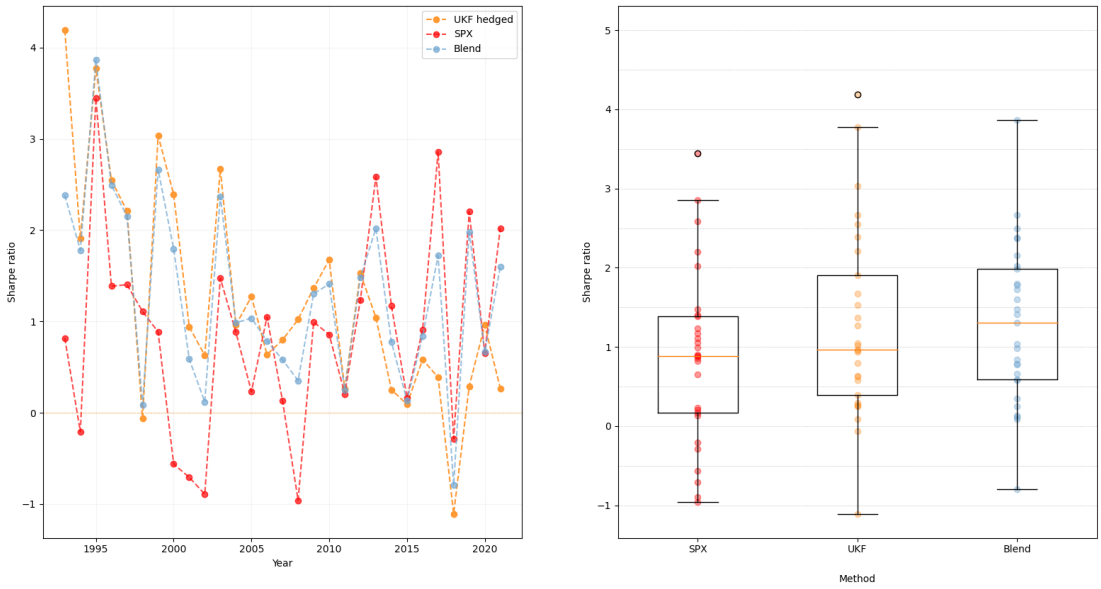

Figure 5.7: Sharpe ratio distribution over 29-years for a UKF model-based trading strategy as outlined in Section 5.3, the S&P 500 with respect to total (excess) return, and a portfolio of a weighted combination of the two.

our strategy did not register greater outperformance during this period. For intuition, we also depict the proportion of assets under consideration that are invested in for all times, in the top panel of Figure 5.6. We notice the obvious empirical fact that with a larger window size, more assets are held. There is an annual seasonality apparent, whereby at the end of each year positions are closed out, and at the start of the year there are spikes for new trades re-entered, as expected by the definition of our trading strategy.

It is reasonable to assume that the beta neutral statistical arbitrage strategy has zero or low (absolute) correlation of returns to the long only market portfolio. Hence, in theory, a blended investment taking a position in the strategy and the long only portfolio could show an outperformance (with respect to Sharpe ratio) relative to both [101]. We implement such a blended strategy by weighting the statistical arbitrage strategy and the long only portfolio as

$$w_i = \frac{S_i}{S_L + S_{LS}}, \qquad S_i := \frac{\mu_i}{\sigma_i^2}, \qquad i \in \{\mathrm{L}, \mathrm{LS}\},$$

where the weights w depend on the mean μ and variance σ^2 of the long portfolio L or long-short portfolio LS returns, optimizing for maximum Sharpe ratio [101, 180]. The true values of the mean and variance parameters are unknown, and for demonstration purposes we take the straightforward approach of setting the means and variances equal in the first year, and equal to their historical values on a rolling window of size (up to) 10 otherwise. We find that the results are not particularly sensitive to (reasonable) values of the choice of rolling window size. Images summarising the Sharpe ratio performance are given in Figure 5.7. Of note, given the right-hand side chart, we notice considerable performance improvement with respect to the blended strategy for the first, second and third quartiles, supporting the theoretical result, and offering a practical hint for deploying such a strategy.

Recent performance: 2008–2022. We notice from the left-hand side image of Figure 5.7 that the strategy has not performed so well in the most recent years, particularly for the years 2015-2019. We conjecture that this could be due to a lower volatility market regime during this time, increased sophistication of market participants and with it competition for alpha / eroded edge, or otherwise due to limitations derived from some underlying assumption of our strategy. Overcoming these challenges is likely of interest to the motivated academic or practitioner, and we offer some avenues for future work in the following conclusion.

5.6 Conclusion

We study multivariate statistical arbitrage under a conditional factor model augmented in a state space framework for US equities trading. Our study shows the modelling approach can yield compelling performance statistics over a 29-year period, an absolute basis and relative to a reasonable benchmark, and relative to a model estimating returns as a function of factors via ordinary least squares. However, for our experimental set up, we did not find that the non-linear state space model was justified relative to the linear case; though it was no worse, any outperformance benefit does not contrast well with the increased modelling complexity introduced. When we compare our results to existing attempts in the

literature, we find that our approach compares well. We also show empirical evidence that blending our investment capital between the long-short strategy and the long-only market portfolio as a function of each strategies estimated mean and variance yields a strategy with improved annual Sharpe ratio, consistent with theory. All results are with respect to reasonable level of transaction costs.

There is the scope for academic and practitioner work on the strategy presented in this paper, particularly with respect to exploring the potential for improvement on the period of strategy underperformance in recent years. We conclude with some final remarks:

- Regarding the data underlying the quantitative strategy, the addition of alternative data beyond factor data and returns time series may improve the predictive power of the model. With respect to the data set we constructed, and by the recent work of [183], it is the case that estimating submodels over clusters of data – for example, stocks grouped by sector – could lead to significant performance improvement.
- The recent work of [178, 184] consider alternative specifications of dynamic factor models for investing, including those with non-linearities at the level of the measurement equation. Such expressions could be handled by our UKF-based framework, and present as a reasonable target for improving performance.
- Recall our brief discussion on related literature in Section 5.2, whereby almost all work cited made use of data on a daily time scale. The literature would be bolstered by analysis completed on alternative time frames, particularly on intraday time scales. This is arguably harder, not the least in terms of data set curation. On the other hand, the recent work of [185] provides evidence of intraday return predictability as a function of factors using machine learning; this could support refining reactive entry and exit rules within the trading strategy.
- Modelling techniques subsuming regime filters could support our statistical arbitrage trading strategy, with recent evidence to this end for a pairs trading application given in [186].
- Finally, and requiring more expert / domain knowledge, it would be interesting to

understand trades entered that are directly mappable to a market phenomenon, such as a stock going ex-dividend, or being influenced by index rebalancing. This is interesting in the context of determining whether the quantitative strategy's profitable trades are explainable in terms of a well-known market effect, or else is supportive of the underlying factor-based theory.

Chapter 6

Conclusion

6.1 Summary remarks

This thesis is written in support of novel Bayesian machine learning-based approaches to financial trading and investment.

In the introduction of Chapter 1, we described a relevant end-to-end workflow for developing quantitative strategies relating to economic edge. The workflow had three chief components essential to both academic and industry research practices: (i) data collection and suitable preprocessing, (ii) data-driven edge modelling, and (iii) the strategic implementation of a terminal strategy designed to optimally capture edge. The overarching objective of this thesis was to examine the role of uncertainty estimation within this workflow. This led to three project contributions to the academic literature, which leveraged a breadth of ideas from the existing literature as presented in Chapter 2.

The first project was presented in Chapter 3 and concerned strategy position sizing. We developed a model for price change prediction within a high-frequency domain. This extended an existing state-of-the-art deep learning model architecture to a case of multi-variate, correlated futures price data. Considering our objective, we engineered our model to output prediction uncertainty estimates, and showed how such uncertainties were useful for weighting position sizes within a trading strategy. This work also provided conceptual motivation for a subset of the themes contained in Chapter 4.

Indeed, in the project work of Chapter 4, the Bayes-based consideration of uncertainty manifested in multiple interesting ways. As a proxy for investor 'views', we modelled equity factor predictions using various machine learning methods, and characterised view uncertainty within a Black-Litterman framework. In developing terminal strategy, we could find optimal portfolio weights for the case of a hierarchical Bayesian Black-Litterman

model crafted in an Arbitrage Pricing Theory setting ('BL-APT'). Thirdly, inspired by the information fusion literature for combining correlated prediction mean and uncertainty estimates streamed from diverse sources, we argued for the data fusion of multiple views held simultaneously about the same portfolio, and demonstrated how this fits within the Black-Litterman framework. We performed extensive empirical work to demonstrate the utility of our theory for BL-APT, achieving reasonable results net of a realistic transaction cost model.

In Chapter 5, we proceeded to reuse the data set constructed in the modelling of Chapter 4, this time in novel pursuit of an alternative edge – a multivariate statistical arbitrage under a conditional factor model, augmented within a Bayesian state space framework. Our filtering model yielded compelling performance statistics over a 29-year period, on an absolute basis, with respect to a reasonable benchmark, and comparable to alternative well-known approaches from the literature where directly comparable. Secondarily, we did not find evidence within our experimental set up that a non-linear specification of the state space model was justified relative to the linear case. Though the non-linear framework was no worse, any outperformance benefit did not contrast well with the increased modelling complexity introduced. Finally, we showed preliminary empirical evidence that blending a pool of investment capital between our long-short statistical arbitrage strategy and the long-only market portfolio, as a function of each strategies estimated mean and variance, yields a combined strategy with improved annual Sharpe ratio, consistent with theory. Again, all results were with respect to reasonable assumptions around transaction costs.

6.2 Contrast and recommendations

The project work explored settings where a market edge could be hypothesised and potentially earned under a reasonable trading or investing strategy. Thus far, this summary makes clear not only the content, but also the sequential nature of the inspiration that lead to the creation of each project. In this final subsection, we offer insights contrasting the core projects, and preliminary recommendations for researchers or practitioners seeking to

either improve the work or find extended avenues of exploration. This is in addition to the suggested 'future work' of previous chapters.

Of course, as is often characteristic of research more generally, in developing the projects, numerous ideas were discarded at varying points of failure within the workflow development cycle. This is a likely reality for those seeking to discover, model and potentially capture market edge via quantitative approaches. On the other hand, our success in discovering sufficiently interesting strategies for capturing market edge depended on the broad exploration that generated such failures. Consequently, the projects demonstrate diversity in that they include multiple asset classes (US equities and interest rate futures), strategy themes (lead-lag trading, longer-term investing and statistical arbitrage), and time scales of trade execution (intraday, daily and semimonthly). Still, the projects can be improved or extended in a variety of ways, or benefit further from cross-applying some of their contained techniques.

In fact, on critical review of Chapter 3 we would certainly recommend that practitioners improve on the strategy exploring the potential for higher backtest-gleaned net risk-adjusted returns. That is, the transaction cost analysis offered implies that the framework is not likely outright tradeable in a production environment with sufficiently high Sharpe – without, at least, further modification. Such modification is conceptually simple – one could, for example, increase the cutoff parameter defined before data set construction, so that the strategy might trade futures moves more likely to overcome a higher transaction cost hurdle, hence improving risk-adjusted returns. Another idea would be to update the strategy to target and trade futures spreads or butterflies (essentially spreads of spreads), rather than trade particular single contracts directionally. This could help in hedging exposure to 'parallel' moves of the futures curve, or uncover alternative idiosyncratic edge. Also, the framework devised could have been utilised for its benefit as an estimated 'gross' signal only, in support of or as input to some other existing strategy. Such strategies are those of interest, say, for intraday-to-mid frequency interest rate relative value hedge funds, and include classic statistical arbitrage or lead-lag plays between a market interest rate

curve and some measure of fair value. An overarching strategy may be based on a slower signal, whereby the gross signal of our project, calculated on a faster time scale, supports superior trade entries and exits, or otherwise improves the trading strategy within some book. Put more succinctly: a signal that is less compelling under reasonable transaction cost assumptions can still be a useful component of an ensemble of signals (that may induce a profitable trading strategy).

These comments, of course, overlap with the ideas given in Chapter 5. The factor-based statistical arbitrage strategy presents as a reasonable starting point net of transaction costs, and showed genuine relative outperformance in all but the most recent few years. Although this presents as a compelling strategy, it could be further improved for today's competitive market environment, most obviously, by incorporating more of the current state-of-the-art in modern statistical arbitrage. Many 'future work' ideas for assessing and trading the idiosyncratic error about fair value are given in Chapter 5, but for practical industrial purposes incorporating a richer initial data set is likely amongst the 'lowest hanging fruit'. This includes the inclusion of data at different time scales, but also data beyond merely US equity returns, including the various alternative data now available in the marketplace, and cross-asset signal or return data, and even cross-regional data from global time zones. The ideas of generation and fusion of multiple signals of Chapter 4 could also be of benefit. The project's concepts remain immediately useful, at least, for hedge funds in intraday-to-mid frequency statistical arbitrage. We note that it likely requires sufficient deep, creative work to yield higher, more persistent Sharpe ratios – say, from the estimated recent 0.5-1.0-like range to a more desirable 1.0-2.0. We suspect that a successful system demonstrating such performance would be relatively enduring – despite the strategy theme being known and explored for well over 30 years.

Finally, the project of Chapter 4 presents a compelling proof of concept to (i) clarify the Black-Litterman framework – a Bayes-based portfolio optimisation method incorporating investor views – including with a factor model underlying (given the enduring contemporary popularity and practicality of factors); and to (ii) incorporate signal fusion

in the context of multiple return mean and (co-)variance estimates. For practicality – that is, relative to available suitable data – this strategy concept was demonstrated in an investment management setting. Using fair implementation assumptions, including respect of transaction costs, this strategy has amongst the least compelling Sharpe ratio, however it trades on the longest time scale and, in practice, arguably offers amongst the best capacity. It would be difficult to implement this framework for genuine investment, however the overarching techniques could definitely be applied in broader domains with respect to more meaningful sources of edge, for example, within hedge funds with a focus on statistical arbitrage.

To their benefit, all strategies presented have fairly robust and simple explanations as to why they work and are expected to be of utility. However, they do harbour significant and non-trivial design and implementation challenges within their end-to-end development – namely across the three dimensions of data set construction, modelling, and portfolio optimization or trade sizing. These dimensions can be expanded on with seemingly arbitrarily high complexity – to the extent improvements support the global strategy. Evidently, such is the competitive and ongoing quest to unveil and capture market edge.

Appendix A

A.0.1 Variational inference

Variational inference is a common technique in statistical inference for estimating intractable distributions. The approach to inference is to approximate an unknown posterior distribution $p(\boldsymbol{\theta}|D)$ with a tractable, parameterised distribution $q(\boldsymbol{\theta})$, such that the Kullback-Leibler (KL) divergence between $q(\boldsymbol{\theta})$ and $p(\boldsymbol{\theta}|D)$ is minimised. That is, one finds an optimal q^* such that

$$q^*(\boldsymbol{\theta}) = \underset{q(\boldsymbol{\theta})\in F}{\arg\min}\, KL(q(\boldsymbol{\theta})||p(\boldsymbol{\theta}|D)) = \underset{q(\boldsymbol{\theta})\in F}{\arg\min}\, \mathbb{E}\frac{\log q(\boldsymbol{\theta})}{\log p(\boldsymbol{\theta}|D)}.$$

Here F denotes a set of candidate densities for q.

By the above specification, the estimation problem is presented as one of optimisation: $q(\boldsymbol{\theta})$ depends on hyperparameters that are optimised in an iterative way, with respect to minimising a divergence measure. Samples can then be efficiently extracted from $q(\boldsymbol{\theta})$ and used for approximate analysis.

Adding a further complication, the objective function above is not computable owing to the KL divergence term depending on $p(D)$ in the denominator. An alternative, equivalent (up to a constant), and solvable optimisation problem can be expressed via the ELBO [187], defined as

$$\text{ELBO}(q) := \mathbb{E}\log p(\boldsymbol{\theta}, D) - \mathbb{E}\log q(\boldsymbol{\theta}). \tag{A.1}$$

With a modicum of algebra it can be shown that minimising the Kullback-Leibler divergence is equivalent to minimising the negative ELBO, since

$$\log p(D) = KL(q(\boldsymbol{\theta})||p(\boldsymbol{\theta}|D)) + \text{ELBO}(q),$$

and $\log p(D)$ does not depend on q.

In mean field variational inference it is assumed that q is a fully-factorised Gaussian – $q(\boldsymbol{\theta}) \sim N(\boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q)$ – such that elements of $\boldsymbol{\theta}$ are mutually independent and have unique

density factors in q. The optmisation problem can be solved via a coordinate ascent algorithm, whereby the density factors are optimised iteratively, all else held constant[188]. Alternative, often complex approaches to variational inference exist, but are beyond the scope of this document.

Compared to, say, sampling methods (given enough time to converge), variational inference is biased. However, variational inference can be less computationaly demanding. That is to say, at least, that in choosing to implement either sampling methods or variational inference there is usually a trade-off between approximation accuracy and scalability.

Appendix B

B.0.1 Choosing alpha – sketch intuition

Assume we are given N trading opportunities $\{X_i\}$ with normally distributed and uncorrelated returns, parameterised by means and variances given by (μ_i, σ_i^2) for each choice i. Also assume these parameters take strictly positive values, and that we trade each opportunity an equal number of times. We seek to trade an optimal amount α_i of X_i to maximises the Sharpe ratio S of the return. We write

$$S = \frac{\sum \alpha_j \mu_j}{\sqrt{\sum \alpha_j^2 \sigma_j^2}}.$$

Taking the derivative of S with respect to α_i yields

$$\frac{dS}{d\alpha_i} = \frac{\mu_i \sum_{j \neq i} \alpha_j^2 \sigma_j^2 - \alpha_i \sigma_i^2 \sum_{j \neq i} \alpha_j \mu_j}{(\sum \alpha_j^2 \sigma_j^2)^{3/2}}$$

which equals 0 for

$$\alpha_i^* = \frac{\mu_i / \sigma_i^2}{\sum_{j \neq i} \alpha_j^* \mu_j / \sum_{j \neq i} \alpha_j^{2*} \sigma_j^2}.$$

One can see that a solution for each i is given by

$$\alpha_i^* = \frac{\mu_i}{\sigma_i^2}.$$

Taking the second derivative of S with respect to α_i yields

$$\frac{d^2 S}{d\alpha_i^2} = \frac{-\sigma_i^2 (\sum_{j \neq i} \alpha_j \mu_j \sum \alpha_j^2 \sigma_j^2 + 3\alpha_i \mu_i \sum_{j \neq i} \alpha_j^2 \sigma_j^2 - 3\alpha_i^2 \sigma_i^2 \sum_{j \neq i} \alpha_j \mu_j)}{(\sum \alpha_j^2 \sigma_j^2)^{5/2}}.$$

Substituting α_i^* we have that

$$\begin{aligned}
\frac{d^2 S}{d\alpha_i^2}\Big|_{\alpha_i = \alpha_i^*} &= \frac{-\sigma_2^2 \Big(\sum_{j \neq i} \mu_j^2/\sigma_j^2 \sum \mu_j^2/\sigma_j^2 + 3\mu_i^2/\sigma_i^2 \sum_{j \neq i} \mu_j^2/\sigma_j^2 - 3\mu_i^2/\sigma_i^2 \sum_{j \neq i} \mu_j^2/\sigma_j^2 \Big)}{(\sum \mu_j^2/\sigma_j^2)^{5/2}} \\
&= \frac{-\sigma_2^2 \cdot \sum_{j \neq i} \mu_j^2/\sigma_j^2}{(\sum \mu_j^2/\sigma_j^2)^{3/2}} \\
&< 0,
\end{aligned}$$

hence for this optimal α_i^* the Sharpe ratio is indeed maximised.

B.0.2 Modelling notes

Initial exploration. We compare a suite of models for predictive performance on out-of-sample data to develop intuition about suitable rates curve model architectures. This is independent of considering any uncertainty measures. Listing these models by increasing complexity, we explore simple and exponentially-weighted averaging, ordinary least squares, PCA, single- and multi-layer perceptrons, CNN, LSTM, CNN-LSTM, and finally a CNN-LSTM Inc in the spirit of [11]. We evaluate model fit by comparing mean-squared error and Huber loss [189] on our validation data, whereby the proceeding observations were broadly consistent. The first clear result is that for the $t + 1$ prediction horizon, there is a clear trend in significantly decreasing prediction error as model complexity increases. Further, the CNN-LSTM Inc always outperforms every other model for both error types, which complements the results of [11]. Secondly, each model broadly displays larger error for larger prediction horizons, which accords with intuition. Considering the $t + 100$ prediction horizon, there is such little variation in error across rows that it is clear that not only do we we have a poor ability to forecast price action relatively far into the future, but all models appear to be of about as little utility as each other. An inflection point for these observations seems apparent at the $t + 10$ prediction horizon.

CNN-LSTM Inc model details. The model architecture is as shown in Figure 3.2, and we follow our reference closely, since we have no strict advantage in selecting this beyond intution. A key difference for our setting, however, is that we do not include any information from the order book beyond Level 1, and the spatial component analogue for our model is equal to adjacent asset prices on the curve, rather then levels of limit order book data on the single asset. Another difference is that our reference work's initial 3 convolutional layers, that convolve price and volume levels at different depths of the limit order book, are not relevant in our case, and so are excluded. To recapitulate some of the other details, we use 6 convolutional layers of 16 filters each and with kernel sizes, respectively, (1,2), (4,1), (4,1), (1,8), (4,1), (4,1). All layers have a leaky ReLU output with parameter $\alpha = 0.01$. Same padding is applied to the latter two layers. The inception layer is a 2x3 tower whereby

the input layer consists of a MaxPooling layer with pool size (3,1) and stride of 1, and 2 convolutional layers with kernel size (1,1) and 32 filters. The output layer of the inception network is 3 more convolutional layers with kernel sizes (1,1), (3,1) and (5,1) respectively, all with 32 filters. All inception modules have same padding, and the convolutional layers have ReLU activation. The LSTM layers have 64 units, with a subsequent dense layer, and linear output activation function. Finally, for replicability sake we add that the MLP model uses ReLU activations at all dense layers but the last, where a linear activation function is instead implemented.

Bayesian linear baseline model details. We follow the Bayesian framework for multioutput regression outlined in Appendix A.2 of [190] in estimating a suitable baseline model for comparing our results. Similarly to Section 2.2 of that paper, we model $\boldsymbol{Y}_t \sim \text{MVN}(\boldsymbol{\mu}_t, \boldsymbol{\Sigma})$ where $\boldsymbol{\mu}_t = \boldsymbol{D}_t\boldsymbol{\beta}$ is an affine function. [Hence we have adjusted our earlier definition of $\boldsymbol{D}_t$ by appending a constant term.] We make the natural assumption that the prior distribution for $\boldsymbol{\Sigma}$ is inverse-Wishart, that is, $\boldsymbol{\Sigma} \sim W^{-1}(\boldsymbol{\Omega}, \nu_0)$. We further assume that the prior for $\boldsymbol{\beta}|\boldsymbol{\Sigma}$ is matrix normal, that is, $\boldsymbol{\beta}|\boldsymbol{\Sigma} \sim N_{(100c+1)\times c}(\boldsymbol{\beta}_0, \boldsymbol{\Sigma}, \boldsymbol{\Sigma}_0)$. Hence, the posterior predictive of $\boldsymbol{Y}_t$ given n historical data observations $\mathcal{D} := (\boldsymbol{D}, \boldsymbol{Y}) = \{(\boldsymbol{D}_i, \boldsymbol{Y}_i) : 1 \leq i \leq n\}$ and a single new observation $\boldsymbol{D}_t$ can be shown to be a multivariate student distribution:

$$\boldsymbol{Y}_t|\boldsymbol{D}_t, \mathcal{D} \sim T((\boldsymbol{\Omega} + \boldsymbol{A}^*).C^{-1}, n + \nu_0),$$

for matrices $\boldsymbol{A}^* = \boldsymbol{Y}^T\boldsymbol{Y} + \boldsymbol{\beta}_0^T\boldsymbol{\Sigma}_0^{-1}\boldsymbol{\beta}_0 - (\boldsymbol{D}^T\boldsymbol{D}\hat{\boldsymbol{\beta}} + \boldsymbol{\Sigma}_0^{-1}\boldsymbol{\beta}_0)^T(\boldsymbol{D}^T\boldsymbol{D} + \boldsymbol{\Sigma}_0^{-1})^{-1}(\boldsymbol{D}^T\boldsymbol{D}\hat{\boldsymbol{\beta}} + \boldsymbol{\Sigma}_0^{-1}\boldsymbol{\beta}_0)$ and $\hat{\boldsymbol{\beta}} = (\boldsymbol{D}^T\boldsymbol{D})^{-1}\boldsymbol{D}^T\boldsymbol{Y}$, and scalars $C^{-1} = 1 + \boldsymbol{D}_t(\boldsymbol{D}^T\boldsymbol{D} + \boldsymbol{\Sigma}_0^{-1})^{-1}\boldsymbol{D}_t^T$, and $\nu_0 = n_0 - (c + (100c + 1)) + 1$. Hence, the required output prediction and uncertainty estimates can be written

$$\widehat{\mathbb{E}[\boldsymbol{Y}_t|\boldsymbol{D}_t, \mathcal{D}]} = \boldsymbol{D}_t(\boldsymbol{D}^T\boldsymbol{D} + \boldsymbol{\Sigma}_0^{-1})^{-1}(\boldsymbol{D}^T\boldsymbol{Y} + \boldsymbol{\Sigma}_0^{-1}\boldsymbol{\beta}_0), \qquad \text{and}$$

$$\widehat{\text{Var}[\boldsymbol{Y}_t|\boldsymbol{D}_t, \mathcal{D}]} = \frac{1}{n + \nu_0 - 2}(\boldsymbol{\Omega} + \boldsymbol{A}^*).C^{-1}.$$

Bayesian OLS

		Base	Al+Ep
	7	-	-
Full	8	0.90	0.85
$\boldsymbol{\Sigma}$	9	2.23	1.46
	10	2.51	1.95
	11	1.32	0.90
	12	2.69	2.93
	Cuml	1.39	1.10

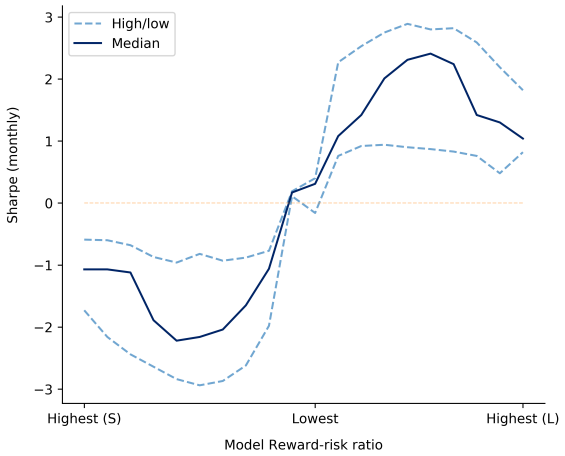

Figure B.1: Left: Monthly Sharpe performance for baseline Bayesian OLS model, for comparison with Table 4.3, for estimating the full covariance matrix. While the base strategy appears to show comparable performance to the more sophisticated models, it does not appear to generally improve taking uncertainty into account. Right: Median out-of-sample model predicted risk-reward ratio (for months 8 to 12 of out-of-sample data) versus monthly Sharpe, for a diagonal covariance MLP. The greatest long (short) ratio is shown to the right (left). Here, out-of-sample Sharpe performance for the short positions is multiplied by -1.

Appendix C

C.0.1 Further APT calculations

This section supplements the calculations of Section 4.2.3 with additional detail. In determining the expectation and covariance for $\boldsymbol{r}|\boldsymbol{q}$ in the case of unknown mean but known covariance, given Equation (4.12) we collect and expand the terms in the exponent up to a factor of $-\frac{1}{2}$, complete the square in $\boldsymbol{f}$, and integrate what is recognisable as a Gaussian integral:

$$\begin{aligned}(\boldsymbol{r}-\boldsymbol{X}\boldsymbol{f})^T\boldsymbol{D}^{-1}(\boldsymbol{r}-\boldsymbol{X}\boldsymbol{f})+(\boldsymbol{f}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{f}-\boldsymbol{\mu})&\\ =\boldsymbol{r}^T\boldsymbol{D}^{-1}\boldsymbol{r}-2\boldsymbol{f}^T\boldsymbol{X}^T\boldsymbol{D}^{-1}\boldsymbol{r}&\\ +(\boldsymbol{X}\boldsymbol{f})^T\boldsymbol{D}^{-1}\boldsymbol{X}\boldsymbol{f}+\boldsymbol{f}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{f}-2\boldsymbol{f}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}+\boldsymbol{\mu}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}&\\ =\boldsymbol{r}^T\boldsymbol{D}^{-1}\boldsymbol{r}+\boldsymbol{f}^T\boldsymbol{H}\boldsymbol{f}-2\boldsymbol{\beta}\boldsymbol{f}+\boldsymbol{\mu}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}&\end{aligned}$$

where $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ denote the mean and covariance for $\boldsymbol{f}|\boldsymbol{q}$, $\boldsymbol{H}:=\boldsymbol{X}^T\boldsymbol{D}^{-1}\boldsymbol{X}+\boldsymbol{\Sigma}^{-1}$ and $\boldsymbol{\beta}:=\boldsymbol{X}^T\boldsymbol{D}^{-1}\boldsymbol{r}+\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}$. Completing the square,

$$\boldsymbol{f}^T\boldsymbol{H}\boldsymbol{f}-2\boldsymbol{f}^T\boldsymbol{\beta}=(\boldsymbol{f}-\boldsymbol{\nu})^T\boldsymbol{H}(\boldsymbol{f}-\boldsymbol{\nu})-\boldsymbol{\nu}^T\boldsymbol{H}\boldsymbol{\nu},\qquad\boldsymbol{\nu}=\boldsymbol{H}^{-1}\boldsymbol{\beta},$$

whereby

$$\int\exp\Big(-\frac{1}{2}(\boldsymbol{f}-\boldsymbol{\nu})^T\boldsymbol{H}(\boldsymbol{f}-\boldsymbol{\nu})\Big)d\boldsymbol{f}=\sqrt{\frac{(2\pi)^k}{|\boldsymbol{H}|}}.$$

Therefore

$$\begin{aligned}p(\boldsymbol{r}|\boldsymbol{q})\propto\exp\Big(-\frac{1}{2}\Big[\boldsymbol{r}^T\boldsymbol{D}^{-1}\boldsymbol{r}+\boldsymbol{\mu}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}-\boldsymbol{\nu}^T\boldsymbol{H}\boldsymbol{\nu}\Big]\Big)&\\ =\exp\Big(-\frac{1}{2}\Big[\boldsymbol{r}^T\boldsymbol{D}^{-1}\boldsymbol{r}+\boldsymbol{\mu}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}-\boldsymbol{\beta}^T\boldsymbol{H}^{-1}\boldsymbol{\beta}\Big]\Big),&\end{aligned}\tag{C.1}$$

and we retrieve the exponent of the Gaussian for $\boldsymbol{r}|\boldsymbol{q}$. Inspired by [105], we next make use of the following Lemma:

Lemma 1 *If a multivariate normal random variable $\boldsymbol{\theta}$ has density $p(\boldsymbol{\theta})$ and $-2\log p(\boldsymbol{\theta})=\boldsymbol{\theta}'\boldsymbol{H}\boldsymbol{\theta}-2\boldsymbol{\nu}'\boldsymbol{\theta}+$ (terms without $\boldsymbol{\theta}$) then var$(\boldsymbol{\theta})=\boldsymbol{H}^{-1}$ and $\mathbb{E}\boldsymbol{\theta}=\boldsymbol{H}^{-1}\boldsymbol{\nu}$.*

Collecting quadratic terms in $\boldsymbol{r}$ from the exponent of Equation (C.1):

$$\begin{aligned} -\boldsymbol{\beta}^T\boldsymbol{H}^{-1}\boldsymbol{\beta} + \boldsymbol{r}^T\boldsymbol{D}^{-1}\boldsymbol{r} &= -[\boldsymbol{X}^T\boldsymbol{D}^{-1}\boldsymbol{r} + \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}]^T\boldsymbol{H}^{-1}[\boldsymbol{X}^T\boldsymbol{D}^{-1}\boldsymbol{r} + \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}] + \boldsymbol{r}^T\boldsymbol{D}^{-1}\boldsymbol{r} \\ &= -\boldsymbol{r}^T\boldsymbol{D}^{-1}\boldsymbol{X}\boldsymbol{H}^{-1}\boldsymbol{X}^T\boldsymbol{D}^{-1}\boldsymbol{r} + \boldsymbol{r}^T\boldsymbol{D}^{-1}\boldsymbol{r} + \text{lower-order terms in } \boldsymbol{r} \\ &= \boldsymbol{r}^T[-\boldsymbol{D}^{-1}\boldsymbol{X}\boldsymbol{H}^{-1}\boldsymbol{X}^T\boldsymbol{D}^{-1} + \boldsymbol{D}^{-1}]\boldsymbol{r} + \text{lower-order terms in } \boldsymbol{r} \end{aligned} \tag{C.2}$$

Making use of the lemma, we have that the posterior predictive distribution $p(\boldsymbol{r}|\boldsymbol{q})$ is multivariate normal with covariance

$$\begin{aligned} \text{var}(\boldsymbol{r}|\boldsymbol{q}) &= [-\boldsymbol{D}^{-1}\boldsymbol{X}\boldsymbol{H}^{-1}\boldsymbol{X}^T\boldsymbol{D}^{-1} + \boldsymbol{D}^{-1}]^{-1}, \\ &= [\boldsymbol{D}^{-1} - \boldsymbol{D}^{-1}\boldsymbol{X}[\boldsymbol{X}^T\boldsymbol{D}^{-1}\boldsymbol{X} + [(\boldsymbol{V}^{-1} + \boldsymbol{\Omega}^{-1})^{-1} + \boldsymbol{F}]^{-1}]^{-1}\boldsymbol{X}^T\boldsymbol{D}^{-1}]^{-1}. \end{aligned}$$

Futher, we collect the linear terms in $\boldsymbol{r}$ from Equation (C.2):

$$-\boldsymbol{r}^T\boldsymbol{D}^{-1}\boldsymbol{X}^T\boldsymbol{H}^{-1}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} - \boldsymbol{\mu}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{H}^{-1}\boldsymbol{X}^T\boldsymbol{D}^{-1}\boldsymbol{r} = -2\boldsymbol{\mu}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{H}^{-1}\boldsymbol{X}^T\boldsymbol{D}^{-1}\boldsymbol{r},$$

so that again by the lemma

$$\begin{aligned} \mathbb{E}(\boldsymbol{r}|\boldsymbol{q}) &= \text{var}(\boldsymbol{r}|\boldsymbol{q})\boldsymbol{D}^{-1}\boldsymbol{X}\boldsymbol{H}^{-1}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} \\ &= \text{var}(\boldsymbol{r}|\boldsymbol{q})\boldsymbol{D}^{-1}\boldsymbol{X}[\boldsymbol{X}^T\boldsymbol{D}^{-1}\boldsymbol{X} + [(\boldsymbol{V}^{-1} + \boldsymbol{\Omega}^{-1})^{-1} + \boldsymbol{F}]^{-1}]^{-1} \\ &\quad \cdot [(\boldsymbol{V}^{-1} + \boldsymbol{\Omega}^{-1})^{-1} + \boldsymbol{F}]^{-1}(\boldsymbol{V}^{-1} + \boldsymbol{\Omega}^{-1})^{-1}(\boldsymbol{V}^{-1}\boldsymbol{\xi} + \boldsymbol{\Omega}^{-1}\boldsymbol{q}). \end{aligned}$$

C.0.2 View fusion under model-based forecasts

Brief overview of view models

To yield the results of Section 4.5 above we first implement three diverse view models. For completeness, we provide the implementation details below.

We implement a Gaussian Process model using scikit-learn in Python [191]. The kernel function is given by the sum of a radial basis function and a white-noise kernel; see, for example, [192] for context on Gaussian Processes for timse-series modelling. We re-fit the model on a semimonthly basis on a rolling data window of 1 year. The predict method is used to generate prediction estimates and an accompanying (total) uncertainty estimate. The uncertainty can be decomposed

into epistemic and aleatoric components by setting the latter as the noise-level parameter estimated for the white-noise kernel. By Equation (2.6) above, the epistemic uncertainty is then the difference between the total uncertainty and the aleatoric uncertainty estimates.

We implement a boosted regression model making use of CatBoost [80]. We re-fit the model on a semimonthly basis on a rolling data window of 2 years. The model yields an aleatoric uncertainty estimate owing to modelling this uncertainty directly via the loss function. An estimate for epistemic uncertainty is approximated by the variance of the underlying ensemble of predictions. Further details of this method can be found in [193].

We estimate an ARIMA model for view generation as a well-known baseline model. Further, this model naturally yields an aleatoric uncertainty estimate. To approximate epistemic uncertainty for this model, we are inspired by Algorithm 3 of [194]. We note that our method is somewhat ad hoc, and has subtle differences: we estimate prediction error variance on a recent out-of-sample lookback window of size 6 months, estimating epistemic uncertainty as the difference between this term and today's aleatoric uncertainty estimate, to a minimum of 1e-8. We re-fit the model on a semimonthly basis on a rolling data window of 1 year.

C.0.3 Performance results by year

The following pages contain performance metrics for market total return (in excess of the risk-free rate), against a collection of Black-Litterman models with varying underlying prediction methods to generate and/or fuse views, over the 29-year observation period. Methods sans the index allocation are normalised by return volatility to equal the ex-post annual return volatility of the market index.

Table C.1: Performance metrics for a collection of Black-Litterman models with varying underlying prediction methods, 1993-1998.

1993						
Portfolio method	Cuml Ret	Ret Vol	Sharpe	IR	Sortino	Max DD
S&P 500 (TR)	7.11	8.59	0.82	-1.94	1.21	-8.84
ARIMA	-17.50	8.59	-2.04	-2.68	-2.01	-16.43
GP	-15.42	8.59	-1.79	-2.60	-1.78	-15.90
Boost	-9.14	8.59	-1.06	-1.84	-1.18	-10.40
PW	-12.04	8.59	-1.40	-2.20	-1.44	-12.37
CU	-11.11	8.59	-1.29	-1.98	-1.47	-13.29
CI	-20.11	8.59	-2.34	-2.81	-2.22	-18.60
ICI	-16.51	8.59	-1.92	-2.46	-1.82	-16.14
1994						
Portfolio method	Cuml Ret	Ret Vol	Sharpe	IR	Sortino	Max DD
S&P 500 (TR)	-2.03	9.80	-0.21	-1.42	-0.28	-8.91
ARIMA	20.00	9.80	2.04	1.73	5.65	-18.76
GP	21.39	9.80	2.18	1.78	5.91	-19.92
Boost	12.53	9.80	1.28	1.04	2.73	-13.63
PW	13.90	9.80	1.42	1.16	3.07	-14.73
CU	20.65	9.80	2.11	1.74	5.76	-19.13
CI	18.24	9.80	1.86	1.59	6.13	-17.71
ICI	19.94	9.80	2.03	1.70	8.08	-18.73
1995						
Portfolio method	Cuml Ret	Ret Vol	Sharpe	IR	Sortino	Max DD
S&P 500 (TR)	26.80	7.81	3.43	-2.57	5.72	-24.15
ARIMA	-12.74	7.81	-1.63	-4.48	-1.67	-14.11
GP	-11.30	7.81	-1.45	-4.33	-1.52	-13.81
Boost	-15.45	7.81	-1.98	-4.56	-2.06	-14.93
PW	-21.30	7.81	-2.73	-5.36	-2.46	-20.11
CU	-11.64	7.81	-1.49	-4.53	-1.57	-14.14
CI	-13.29	7.81	-1.70	-4.87	-1.77	-14.72
ICI	-12.45	7.81	-1.59	-4.78	-1.67	-14.07
1996						
Portfolio method	Cuml Ret	Ret Vol	Sharpe	IR	Sortino	Max DD
S&P 500 (TR)	16.31	11.75	1.38	-1.60	1.96	-18.96
ARIMA	14.18	11.75	1.21	-0.33	2.78	-19.34
GP	9.49	11.75	0.81	-0.59	1.65	-16.73
Boost	10.90	11.75	0.93	-0.49	1.62	-18.25
PW	9.34	11.75	0.79	-0.60	1.34	-19.69
CU	20.40	11.75	1.74	0.02	4.69	-22.35
CI	25.06	11.75	2.13	0.29	6.07	-25.21
ICI	21.15	11.75	1.80	0.06	4.78	-23.72
1997						
Portfolio method	Cuml Ret	Ret Vol	Sharpe	IR	Sortino	Max DD
S&P 500 (TR)	25.34	18.09	1.40	-1.04	2.04	-23.74
ARIMA	26.74	18.09	1.48	-0.05	3.39	-21.90
GP	31.28	18.09	1.73	0.17	3.71	-25.04
Boost	32.70	18.09	1.81	0.22	4.57	-26.85
PW	30.50	18.09	1.69	0.13	3.69	-26.26
CU	28.95	18.09	1.60	0.05	3.75	-23.06
CI	23.15	18.09	1.28	-0.22	2.61	-18.85
ICI	28.57	18.09	1.58	0.04	3.58	-22.90
1998						
Portfolio method	Cuml Ret	Ret Vol	Sharpe	IR	Sortino	Max DD
S&P 500 (TR)	22.47	20.25	1.11	-0.51	1.57	-22.93
ARIMA	12.62	20.25	0.62	-0.34	0.91	-16.82
GP	10.57	20.25	0.52	-0.38	0.73	-16.72
Boost	5.64	20.25	0.28	-0.58	0.41	-19.44
PW	15.04	20.25	0.74	-0.24	1.06	-15.90
CU	29.10	20.25	1.44	0.22	2.41	-27.24
CI	17.15	20.25	0.85	-0.18	1.24	-16.74
ICI	26.14	20.25	1.29	0.12	2.20	-24.05

Table C.2: Performance metrics for a collection of Black-Litterman models with varying underlying prediction methods, 1999-2004.

1999						
Portfolio method	Cuml Ret	Ret Vol	Sharpe	IR	Sortino	Max DD
S&P 500 (TR)	16.15	18.04	0.90	-3.27	1.34	-14.82
ARIMA	-28.79	18.05	-1.60	-1.72	-1.71	-27.22
GP	-21.79	18.05	-1.21	-1.58	-1.40	-22.85
Boost	19.04	18.05	1.06	-0.12	1.64	-25.24
PW	-9.48	18.05	-0.53	-1.13	-0.68	-20.84
CU	-3.66	18.05	-0.20	-0.94	-0.27	-21.42
CI	-19.29	18.05	-1.07	-1.39	-1.27	-20.96
ICI	-32.13	18.05	-1.78	-1.89	-1.89	-30.89
2000						
Portfolio method	Cuml Ret	Ret Vol	Sharpe	IR	Sortino	Max DD
S&P 500 (TR)	-12.80	22.18	-0.58	-0.50	-0.81	-20.07
ARIMA	-5.65	22.17	-0.25	-0.10	-0.33	-26.38
GP	-8.89	22.17	-0.40	-0.23	-0.51	-26.27
Boost	-29.64	22.17	-1.34	-0.83	-1.55	-38.33
PW	-10.81	22.17	-0.49	-0.32	-0.64	-27.58
CU	1.46	22.17	0.07	0.15	0.09	-24.08
CI	6.15	22.17	0.28	0.34	0.39	-23.62
ICI	-6.63	22.17	-0.30	-0.12	-0.38	-27.23
2001						
Portfolio method	Cuml Ret	Ret Vol	Sharpe	IR	Sortino	Max DD
S&P 500 (TR)	-14.12	21.51	-0.67	-2.25	-0.92	-30.91
ARIMA	36.09	21.51	1.68	1.72	2.57	-35.22
GP	36.18	21.51	1.68	1.67	2.60	-34.88
Boost	25.65	21.51	1.19	1.34	1.64	-30.70
PW	23.71	21.51	1.10	1.28	1.48	-29.85
CU	21.26	21.51	0.99	1.24	1.39	-26.00
CI	35.16	21.51	1.63	1.66	2.62	-33.08
ICI	34.26	21.51	1.59	1.64	2.52	-33.28
2002						
Portfolio method	Cuml Ret	Ret Vol	Sharpe	IR	Sortino	Max DD
S&P 500 (TR)	-23.24	25.99	-0.89	-1.43	-1.28	-33.81
ARIMA	26.07	25.99	1.00	1.49	1.53	-34.82
GP	33.51	25.99	1.29	1.73	2.10	-37.44
Boost	31.67	25.99	1.22	1.65	1.99	-36.37
PW	33.17	25.99	1.28	1.68	2.09	-37.85
CU	4.75	25.99	0.18	0.83	0.24	-30.08
CI	22.01	25.99	0.85	1.37	1.28	-31.67
ICI	26.35	25.99	1.01	1.48	1.57	-35.08
2003						
Portfolio method	Cuml Ret	Ret Vol	Sharpe	IR	Sortino	Max DD
S&P 500 (TR)	25.64	17.04	1.50	-2.42	2.32	-28.43
ARIMA	-23.61	17.03	-1.39	-2.48	-1.53	-23.77
GP	-27.81	17.03	-1.63	-2.70	-1.73	-26.21
Boost	-31.93	17.03	-1.87	-2.76	-1.95	-30.85
PW	-31.18	17.03	-1.83	-2.80	-1.86	-29.23
CU	-20.25	17.03	-1.19	-2.60	-1.28	-22.22
CI	-20.35	17.03	-1.19	-2.32	-1.36	-21.51
ICI	-21.49	17.03	-1.26	-2.39	-1.40	-22.16
2004						
Portfolio method	Cuml Ret	Ret Vol	Sharpe	IR	Sortino	Max DD
S&P 500 (TR)	9.78	11.07	0.88	-1.37	1.28	-12.49
ARIMA	14.26	11.07	1.29	0.26	2.62	-15.21
GP	15.63	11.07	1.41	0.35	3.08	-16.01
Boost	12.73	11.07	1.15	0.16	2.41	-14.50
PW	14.71	11.07	1.33	0.30	3.02	-14.74
CU	12.98	11.07	1.17	0.17	2.22	-15.10
CI	12.22	11.07	1.10	0.13	2.18	-14.11
ICI	16.73	11.07	1.51	0.43	3.21	-15.57

Table C.3: Performance metrics for a collection of Black-Litterman models with varying underlying prediction methods, 2005-2010.

2005						
Portfolio method	Cuml Ret	Ret Vol	Sharpe	IR	Sortino	Max DD
S&P 500 (TR)	2.37	10.26	0.23	-3.66	0.33	-9.87
ARIMA	3.65	10.27	0.36	-0.24	0.52	-9.98
GP	-0.49	10.27	-0.05	-0.57	-0.07	-11.40
Boost	1.49	10.27	0.15	-0.42	0.22	-11.54
PW	0.10	10.27	0.01	-0.53	0.01	-12.06
CU	1.58	10.27	0.15	-0.44	0.23	-10.15
CI	1.62	10.27	0.16	-0.41	0.22	-10.65
ICI	1.23	10.27	0.12	-0.45	0.17	-9.78

2006						
Cuml Ret	Ret Vol	Sharpe	IR	Sortino	Max DD	
10.49	10.02	1.05	-3.14	1.59	-12.93	
-3.92	10.02	-0.39	-1.17	-0.58	-15.33	
-2.43	10.02	-0.24	-1.03	-0.36	-14.76	
-4.06	10.02	-0.41	-1.14	-0.58	-15.78	
-6.36	10.02	-0.63	-1.33	-0.87	-16.78	
3.40	10.02	0.34	-0.59	0.61	-13.37	
-2.07	10.02	-0.21	-1.02	-0.32	-14.31	
1.52	10.02	0.15	-0.74	0.27	-12.77	

2007						
Portfolio method	Cuml Ret	Ret Vol	Sharpe	IR	Sortino	Max DD
S&P 500 (TR)	2.07	15.96	0.13	-4.06	0.17	-10.66
ARIMA	-1.06	15.96	-0.07	-0.33	-0.10	-12.74
GP	-1.52	15.96	-0.10	-0.35	-0.14	-13.27
Boost	-3.80	15.96	-0.24	-0.48	-0.35	-15.23
PW	1.83	15.96	0.11	-0.21	0.18	-14.36
CU	-13.65	15.96	-0.86	-0.87	-1.09	-17.35
CI	-3.35	15.96	-0.21	-0.44	-0.31	-11.66
ICI	-0.99	15.96	-0.06	-0.31	-0.08	-9.66

2008						
Cuml Ret	Ret Vol	Sharpe	IR	Sortino	Max DD	
-39.34	40.88	-0.96	-0.77	-1.31	-47.78	
17.00	40.88	0.42	1.12	0.60	-42.30	
46.05	40.88	1.13	1.69	1.83	-40.38	
39.48	40.88	0.97	1.34	1.56	-43.34	
56.22	40.88	1.38	1.81	2.61	-46.29	
67.57	40.88	1.65	2.27	3.13	-51.49	
47.33	40.88	1.16	1.79	2.16	-42.41	
60.76	40.88	1.49	2.33	2.91	-49.16	

2009						
Portfolio method	Cuml Ret	Ret Vol	Sharpe	IR	Sortino	Max DD
S&P 500 (TR)	27.09	27.22	1.00	-0.83	1.45	-41.01
ARIMA	17.54	27.22	0.64	-0.13	1.13	-21.65
GP	7.78	27.22	0.29	-0.43	0.48	-18.56
Boost	-33.40	27.22	-1.23	-1.76	-1.51	-35.25
PW	-9.72	27.22	-0.36	-1.00	-0.55	-19.77
CU	19.15	27.22	0.70	-0.08	1.30	-21.64
CI	25.99	27.22	0.95	0.12	1.52	-31.85
ICI	27.58	27.22	1.01	0.17	1.74	-27.00

2010						
Cuml Ret	Ret Vol	Sharpe	IR	Sortino	Max DD	
15.47	18.02	0.86	-1.81	1.23	-19.54	
-41.11	18.02	-2.28	-2.30	-2.19	-37.28	
-42.99	18.02	-2.38	-2.42	-2.33	-38.98	
-33.99	18.02	-1.89	-2.07	-1.97	-33.55	
-36.86	18.02	-2.04	-2.17	-2.06	-34.57	
-36.30	18.02	-2.01	-2.14	-2.03	-35.08	
-37.32	18.02	-2.07	-2.14	-1.98	-34.19	
-38.24	18.02	-2.12	-2.16	-1.99	-33.98	

Table C.4: Performance metrics for a collection of Black-Litterman models with varying underlying prediction methods, 2011-2016.

Portfolio method	Cuml Ret	Ret Vol	Sharpe	IR	Sortino	Max DD
	2011					
S&P 500 (TR)	4.78	23.24	0.21	0.83	0.28	-18.64
ARIMA	-29.76	23.24	-1.28	-1.08	-1.35	-34.76
GP	-29.17	23.24	-1.26	-1.03	-1.33	-33.83
Boost	-8.30	23.24	-0.36	-0.41	-0.45	-18.13
PW	-8.41	23.24	-0.36	-0.42	-0.48	-16.08
CU	-15.96	23.24	-0.69	-0.56	-0.80	-29.10
CI	-34.31	23.24	-1.48	-1.19	-1.53	-32.76
ICI	-26.26	23.24	-1.13	-0.92	-1.18	-33.43
	2012					
S&P 500 (TR)	15.63	12.73	1.24	-0.23	1.89	-14.23
ARIMA	22.54	12.73	1.77	0.46	3.23	-19.67
GP	29.53	12.73	2.32	0.86	5.28	-25.51
Boost	28.89	12.73	2.27	0.83	5.22	-25.12
PW	25.43	12.73	2.00	0.65	4.23	-24.06
CU	15.64	12.73	1.23	0.09	1.79	-15.94
CI	19.15	12.73	1.50	0.28	2.45	-18.43
ICI	18.62	12.73	1.46	0.25	2.45	-16.55
	2013					
S&P 500 (TR)	28.68	11.05	2.60	-1.62	3.96	-22.78
ARIMA	2.61	11.05	0.24	-1.74	0.34	-11.82
GP	-4.37	11.05	-0.40	-2.45	-0.51	-16.53
Boost	7.30	11.05	0.66	-1.47	0.96	-8.04
PW	-1.12	11.05	-0.10	-2.36	-0.14	-14.72
CU	-11.24	11.05	-1.02	-2.96	-1.17	-19.21
CI	4.64	11.05	0.42	-1.57	0.63	-9.25
ICI	-3.97	11.05	-0.36	-2.40	-0.47	-15.92
	2014					
S&P 500 (TR)	13.48	11.33	1.19	1.09	1.67	-18.27
ARIMA	8.11	11.33	0.72	-0.35	1.74	-10.11
GP	7.28	11.33	0.64	-0.42	1.33	-8.94
Boost	16.40	11.33	1.45	0.18	3.85	-15.84
PW	14.34	11.33	1.27	0.05	3.14	-14.43
CU	1.81	11.33	0.16	-0.79	0.29	-7.87
CI	6.04	11.33	0.53	-0.47	1.26	-9.70
ICI	7.34	11.33	0.65	-0.39	1.43	-9.50
	2015					
S&P 500 (TR)	2.57	15.46	0.17	0.91	0.23	-12.04
ARIMA	29.50	15.46	1.91	1.56	2.82	-24.32
GP	25.13	15.46	1.62	1.34	2.52	-24.29
Boost	28.98	15.46	1.87	1.47	3.17	-26.31
PW	28.45	15.46	1.84	1.56	2.77	-26.20
CU	31.35	15.46	2.03	1.83	2.96	-26.59
CI	27.01	15.46	1.75	1.45	2.56	-22.93
ICI	22.50	15.46	1.46	1.12	2.42	-23.22
	2016					
S&P 500 (TR)	11.94	13.07	0.91	-1.32	1.30	-20.80
ARIMA	-0.11	13.07	-0.01	-0.90	-0.01	-11.81
GP	0.75	13.07	0.06	-0.84	0.11	-12.31
Boost	-3.25	13.07	-0.25	-1.08	-0.39	-10.62
PW	-11.84	13.07	-0.91	-1.60	-1.22	-12.92
CU	-1.96	13.07	-0.15	-1.01	-0.25	-12.55
CI	2.62	13.07	0.20	-0.73	0.38	-13.80
ICI	6.23	13.07	0.48	-0.52	1.03	-13.95

Table C.5: Performance metrics for a collection of Black-Litterman models with varying underlying prediction methods, 2017-2021.

2017						
Portfolio method	Cuml Ret	Ret Vol	Sharpe	IR	Sortino	Max DD
S&P 500 (TR)	19.16	6.67	2.89	-1.04	4.55	-17.04
ARIMA	-1.08	6.67	-0.16	-2.14	-0.24	-7.76
GP	0.06	6.67	0.01	-2.03	0.02	-5.84
Boost	-3.86	6.67	-0.58	-2.50	-0.79	-10.74
PW	-7.71	6.67	-1.16	-2.99	-1.41	-12.44
CU	-0.88	6.67	-0.13	-2.16	-0.20	-6.45
CI	2.51	6.67	0.38	-1.75	0.61	-6.02
ICI	4.71	6.67	0.71	-1.48	1.32	-4.78

2018						
Portfolio method	Cuml Ret	Ret Vol	Sharpe	IR	Sortino	Max DD
S&P 500 (TR)	-4.81	17.03	-0.28	-0.77	-0.37	-19.81
ARIMA	4.96	17.03	0.29	0.50	0.40	-14.97
GP	-0.59	17.03	-0.03	0.26	-0.04	-14.22
Boost	6.59	17.03	0.39	0.61	0.56	-16.52
PW	13.39	17.03	0.79	0.87	1.17	-17.03
CU	7.20	17.03	0.42	0.57	0.57	-12.90
CI	2.18	17.03	0.13	0.37	0.17	-12.58
ICI	-5.87	17.03	-0.34	0.05	-0.44	-14.05

2019						
Portfolio method	Cuml Ret	Ret Vol	Sharpe	IR	Sortino	Max DD
S&P 500 (TR)	26.03	12.45	2.09	-0.79	3.02	-24.34
ARIMA	22.69	12.45	1.37	-0.16	3.17	-22.74
GP	20.72	12.45	1.66	-0.30	3.69	-21.68
Boost	23.24	12.45	1.87	-0.15	4.15	-20.27
PW	16.83	12.45	1.35	-0.50	3.33	-19.62
CU	17.75	12.45	1.43	-0.50	2.73	-19.82
CI	23.50	12.45	1.89	-0.15	3.68	-21.02
ICI	19.15	12.45	1.54	-0.42	2.47	-19.28

2020						
Portfolio method	Cuml Ret	Ret Vol	Sharpe	IR	Sortino	Max DD
S&P 500 (TR)	22.51	34.37	0.65	-2.26	0.90	-41.22
ARIMA	17.36	34.37	0.51	-0.17	0.61	-38.09
GP	-1.95	34.37	-0.06	-0.59	-0.07	-36.61
Boost	26.22	34.37	0.76	0.03	1.01	-39.32
PW	22.99	34.37	0.67	-0.04	0.86	-37.70
CU	3.84	34.37	0.11	-0.47	0.13	-36.68
CI	8.55	34.37	0.25	-0.36	0.29	-36.89
ICI	8.74	34.37	0.25	-0.36	0.30	-36.94

2021						
Portfolio method	Cuml Ret	Ret Vol	Sharpe	IR	Sortino	Max DD
S&P 500 (TR)	26.10	13.07	2.00	1.55	2.97	-23.86
ARIMA	-5.78	13.07	-0.46	-1.65	-0.60	-12.93
GP	-2.48	13.07	-0.20	-1.47	-0.27	-10.21
Boost	6.64	13.07	0.53	-0.81	0.84	-12.56
PW	-5.24	13.07	-0.42	-1.55	-0.53	-12.61
CU	-0.34	13.07	-0.03	-1.37	-0.04	-9.50
CI	-1.12	13.07	-0.09	-1.34	-0.13	-9.80
ICI	-2.15	13.07	-0.17	-1.40	-0.24	-10.50

Bibliography

[1] M. Villarroel, "thesis_template." `https://github.com/maurovm/thesis_template`, 2023.

[2] G. E. Box, "Robustness in the Strategy of Scientific Model Building," in *Robustness in Statistics*, pp. 201–236, Academic Press, 1979.

[3] H. Markowitz, "Portfolio Selection," *The Journal of Finance*, vol. 7, no. 1, pp. 77–91, 1952.

[4] W. F. Sharpe, "The Sharpe Ratio," *The Journal of Portfolio Management*, vol. 21, no. 1, pp. 49–58, 1994.

[5] J. L. Kelly, "A new interpretation of information rate," *IEEE Information Theory Society*, vol. 2, pp. 185–189, 1956.

[6] C. R. Harvey, S. Rattray, and O. V. Hemert, *Strategic Risk Management*. Wiley, 2021.

[7] E. F. Fama, "Efficient Capital Markets: A Review of Theory and Empirical Work," *The Journal of Finance*, vol. 25, no. 2, pp. 383–417, 1970.

[8] A. Menkveld, "High frequency trading and the new market makers," *Journal of Financial Markets*, vol. 16, no. 4, pp. 712–740, 2013.

[9] M. Baron, J. Brogaard, H. Björn, and A. Kirilenko, "Risk and Return in High-Frequency Trading," *Journal of Financial and Quantitative Analysis*, vol. 54, no. 3, pp. 993–1024, 2019.

[10] S. A. Ross, "The arbitrage theory of capital asset pricing," *Journal of Economic Theory*, vol. 13, no. 3, pp. 341–360, 1976.

[11] Z. Zhang, S. Zohren, and S. Roberts, "DeepLOB: Deep Convolutional Neural Networks for Limit Order Books," *IEEE Transactions on Signal Processing*, vol. 67, 2018.

[12] F. Black and R. B. Litterman, "Asset Allocation: Combining Investor Views with Market Equilibrium," *The Journal of Fixed Income*, vol. 1, no. 2, pp. 7–18, 1991.

[13] B. M. Henrique, V. A. Sobreiro, and H. Kimura, "Literature review: Machine learning techniques applied to financial market prediction," *Expert Systems with Applications*, vol. 124, p. 226–251, 2019.

[14] J. Hull, *Options, futures, and other derivatives*. Upper Saddle River, NJ [u.a.]: Pearson Prentice Hall, 6. ed., pearson internat. ed ed., 2006.

[15] W. F. Sharpe, "Capital asset prices: A theory of market equilibrium under conditions of risk," *The Journal of Finance*, vol. 19, no. 3, pp. 425–442, 1964.

[16] E. F. Fama and K. R. French, "The cross-section of expected stock returns," *The Journal of Finance*, vol. 47, no. 2, pp. 427–465, 1992.

[17] E. F. Fama and K. R. French, "Common risk factors in the returns on stocks and bonds," *Journal of Financial Economics*, vol. 33, no. 1, pp. 3–56, 1993.

[18] M. M. Carhart, "On persistence in mutual fund performance," *The Journal of Finance*, vol. 52, no. 1, pp. 57–82, 1997.

[19] M. Farboodi, D. Singal, L. Veldkamp, and V. Venkateswaran, "Valuing financial data," Working Paper 29894, National Bureau of Economic Research, 3 2022.

[20] M. Evans and T. Swartz, "Methods for Approximating Integrals in Statistics with Special Emphasis on Bayesian Integration Problems," *Statistical Science*, vol. 10, no. 3, pp. 254 – 272, 1995.

[21] A. Kendall and Y. Gal, "What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision?," in *Advances in Neural Information Processing Systems*, vol. 30, pp. 5574–5584, Curran Associates, Inc., 2017.

[22] Y. Kwon, J.-H. Won, B. J. Kim, and M. C. Paik, "Uncertainty quantification using Bayesian neural networks in classification: Application to biomedical image segmentation," *Computational Statistics & Data Analysis*, vol. 142, no. C, 2020.

[23] E. Hüllermeier and W. Waegeman, "Aleatoric and epistemic uncertainty in machine learning: an introduction to concepts and methods.," *Machine Learning*, vol. 110, pp. 457–506, 2021.

[24] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. The MIT Press, 11 2005.

[25] S. J. Roberts, M. Osborne, M. Ebden, S. D. Reece, N. J. Gibson, and S. Aigrain, "Gaussian processes for time-series modelling," *Philosophical transactions of the Royal Society. Series A.*, vol. 371, p. 20110550, 2012.

[26] N. F. James Hensman and N. D. Lawrence, "Gaussian Processes for Big Data," in *UAI'13: Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence*, p. 282–290, 2013.

[27] X. S. Haitao Liu, Yew-Soon Ong and J. Cai, "When Gaussian Process Meets Big Data: A Review of Scalable GPs," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, pp. 4405–4423, 2018.

[28] R. H. Shumway and D. S. Stoffer, "Chapter 6 State-Space Models," in *Time Series Analysis and Its Applications: With R Examples* (2nd, ed.), Springer Texts in Statistics, pp. 324–411, Springer New York, 2006.

[29] E. D. Blanchard, A. Sandu, and C. Sandu, "Parameter estimation method using an extended Kalman Filter," in *Proceedings of the Joint North America, Asia-Pacific ISTVS Conference and Annual Meeting of Japanese Society for Terramechanics*, pp. 1–14, 2007.

[30] W. McCulloch and W. Pitts, "A Logical Calculus of Ideas Immanent in Nervous Activity," *Bulletin of Mathematical Biophysics*, vol. 5, pp. 115–133, 1943.

[31] F. Rosenblatt, "The Perceptron: a Probabilistic Model for Information Storage and Organization in the Brain," *Psychological Review*, pp. 65–386, 1958.

[32] D. J. C. MacKay, *Information Theory, Inference & Learning Algorithms*. Cambridge University Press, 2002.

[33] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of Control, Signals and Systems*, vol. 2, pp. 303–314, 1989.

[34] K. Hornik, M. Stinchcombe, and H. White, "Multilayer Feedforward Networks Are Universal Approximators," *Neural Networks*, vol. 2, pp. 359–366, 1989.

[35] K.-H. Tan and B. P. Lim, "The artificial intelligence renaissance: deep learning and the road to human-level machine intelligence," *APSIPA Transactions on Signal and Information Processing*, vol. 7, pp. 1–19, 2018.

[36] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[37] J. Schmidhuber, "Deep Learning in Neural Networks: An Overview," *Neural Networks*, vol. 61, pp. 85–117, 2015.

[38] Y. Lecun, K. Kavukcuoglu, and C. Farabet, "Convolutional Networks and Applications in Vision," pp. 253–256, 5 2010.

[39] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition." `https://arxiv.org/abs/1409.1556`, 2014. [Online; last accessed 20-May-2019].

[40] S. Bai, J. Z. Kolter, and V. Koltun, "An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling." `http://arxiv.org/abs/1803.01271`, 2018. [Online; last accessed 02-Jan-2020].

[41] J. Donahue, L. A. Hendricks, M. Rohrbach, S. Venugopalan, S. Guadarrama, K. Saenko, and T. Darrell, "Long-term Recurrent Convolutional Networks for Visual Recognition and Description." `https://arxiv.org/abs/1411.4389`, 2014. [Online; last accessed 20-Jul-2019].

[42] Z. C. Lipton, "A critical review of recurrent neural networks for sequence learning." `https://arxiv.org/abs/1506.00019`, 2015. [Online; last accessed 02-Jan-2020].

[43] P. Werbos, "Backpropagation through time: what it does and how to do it," *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1550–1560, 1990.

[44] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, pp. 1735–1780, 1997.

[45] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to Sequence Learning with Neural Networks," in *Advances in Neural Information Processing Systems*, vol. 27, Curran Associates, Inc., 2014.

[46] P. Lara-Benítez, M. Carranza-García, and J. C. Riquelme, "An experimental review on deep learning architectures for time series forecasting," *International Journal of Neural Systems*, vol. 31, no. 3, 2021.

[47] S. Linnainmaa, "The representation of the cumulative rounding error of an algorithm as a taylor expansion of the local rounding errors," Master's thesis, Univ. Helsinki, 1970.

[48] S. Linnainmaa, "Taylor expansion of the accumulated rounding error," *BIT Numerical Mathematics volume*, vol. 16, no. 2, p. 146–160, 1976.

[49] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[50] J. Duchi, E. Hazan, and Y. Singer, "Adaptive Subgradient Methods for Online Learning and Stochastic Optimization," *Journal of Machine Learning Research*, vol. 12, no. 61, p. 2121–2159, 2011.

[51] D. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization." `https://arxiv.org/abs/1412.6980`, 2014. [Online; last accessed 02-Jan-2020].

[52] H. Robbins and S. Monro, "A Stochastic Approximation Method," *The Annals of Mathematical Statistics*, vol. 22, no. 3, pp. 400 – 407, 1951.

[53] S. Geman, E. Bienenstock, and R. Doursat, "Neural Networks and the Bias/Variance Dilemma," *Neural Computation*, vol. 4, no. 1, pp. 1–58, 1992.

[54] R. Tibshirani, "Regression Shrinkage and Selection via the Lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.

[55] A. E. Hoerl and R. W. Kennard, "Ridge Regression: Biased Estimation for Nonorthogonal Problems," *Technometrics*, vol. 12, no. 1, pp. 55–67, 1970.

[56] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors." `https://arxiv.org/abs/1207.0580`, 2012. [Online; last accessed 20-Jul-2019].

[57] H. S. Alex Labach and S. Valaee, "Survey of Dropout Methods for Deep Neural Networks." `https://arxiv.org/abs/1904.13310`, 2019. [Online; last accessed 20-Jul-2019].

[58] D. J. C. MacKay, "A practical Bayesian framework for backpropagation networks," *Neural Computation*, vol. 4, no. 3, p. 448–472, 1992.

[59] R. M. Neal, "Priors for Infinite Networks," in *Bayesian Learning for Neural Networks*, Chapman & Hall/CRC Handbooks of Modern Statistical Methods, pp. 29–53, Springer New York, 1996.

[60] C. Williams, "Computing With Infinite Networks," in *Advances in Neural Information Processing Systems*, vol. 9, pp. 295–301, MIT Press, 1996.

[61] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight Uncertainty in Neural Networks," in *Proceedings of the 32nd International Conference on Machine Learning*, vol. 37, pp. 1613–1622, Proceedings of Machine Learning Research, 7 2015.

[62] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes." `https://arxiv.org/abs/1312.6114`, 2013. [Online; last accessed 02-Jan-2020].

[63] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning." `https://arxiv.org/abs/1506.02142`, 2016. [Online; last accessed 20-Jul-2019].

[64] Y. Gal and Z. Ghahramani, "A Theoretically Grounded Application of Dropout in Recurrent Neural Networks," in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, p. 1027–1035, 2016.

[65] Y. Gal and Z. Ghahramani, "Bayesian Convolutional Neural Networks with Bernoulli Approximate Variational Inference." `https://arxiv.org/abs/1506.02158`, 2016. [Online; last accessed 02-Jan-2020].

[66] W.-Y. Loh, "Fifty Years of Classification and Regression Trees," *International Statistical Review*, vol. 82, pp. 329–348, December 2014.

[67] L. Breiman, J. Friedman, C. Stone, and R. Olshen, *Classification and Regression Trees*. Taylor & Francis, 1984.

[68] L. Breiman, "Bias, Variance , And Arcing Classifiers," *Technical Report 460, Statistics Department, University of California*, 4 1996.

[69] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996.

[70] L. G. Valiant, "A Theory of the Learnable," *Communications of the ACM*, vol. 27, p. 1134–1142, 11 1984.

[71] R. E. Schapire, "The strength of weak learnability," *Machine Learning*, vol. 5, no. 2, pp. 197–227, 1990.

[72] H. Drucker and C. Cortes, "Boosting Decision Trees," in *Advances in Neural Information Processing Systems*, vol. 8, MIT Press, 1995.

[73] Y. Freund and R. E. Schapire, "Experiments with a New Boosting Algorithm," in *Proceedings of the Thirteenth International Conference on International Conference on Machine Learning*, p. 148–156, Morgan Kaufmann Publishers Inc., 1996.

[74] H. Drucker, "Improving Regressors Using Boosting Techniques," *Proceedings of the 14th International Conference on Machine Learning*, 08 1997.

[75] L. Breiman, "Arcing the edge." `https://api.semanticscholar.org/CorpusID:14849468`, 1997. [Online; last accessed 12-Jan-2022].

[76] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *The Annals of Statistics*, vol. 29, no. 5, pp. 1189–1232, 2001.

[77] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, p. 785–794, Association for Computing Machinery, 2016.

[78] D. Nielsen, "Tree Boosting With XGBoost - Why Does XGBoost Win "Every" Machine Learning Competition?," Master's thesis, Norwegian University of Science and Technology, 2016.

[79] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "Lightgbm: A highly efficient gradient boosting decision tree," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, (Red Hook, NY, USA), p. 3149–3157, Curran Associates Inc., 2017.

[80] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, "CatBoost: Unbiased Boosting with Categorical Features," in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, p. 6639–6649, Curran Associates Inc., 2018.

[81] S. M. Lundberg and S.-I. Lee, "A Unified Approach to Interpreting Model Predictions," in *Advances in Neural Information Processing Systems*, vol. 30, Curran Associates, Inc., 2017.

[82] M. H. Shaker and E. Hüllermeier, "Aleatoric and Epistemic Uncertainty with Random Forests," in *Advances in Intelligent Data Analysis XVIII*, pp. 444–456, Springer International Publishing, 2020.

[83] T. Duan, A. Anand, D. Y. Ding, K. K. Thai, S. Basu, A. Ng, and A. Schuler, "NGBoost: Natural Gradient Boosting for Probabilistic Prediction," in *Proceedings of the 37th International Conference on Machine Learning*, vol. 119, pp. 2690–2700, Proceedings of Machine Learning Research, 7 2020.

[84] A. Malinin, L. Prokhorenkova, and A. Ustimenko, "Uncertainty in Gradient Boosting via Ensembles." `https://arxiv.org/abs/2006.10562`, 2021. [Online; last accessed 07-Mar-2022].

[85] B. Fitzenberger, "The moving blocks bootstrap and robust inference for linear least squares and quantile regressions," *Journal of Econometrics*, vol. 82, no. 2, pp. 235–287, 1998.

[86] H. M. Markowitz, *Portfolio Selection: Efficient Diversification of Investments*. Yale University Press, 1959.

[87] V. K. Chopra and W. T. Ziemba, "The Effect of Errors in Means, Variances, and Covariances on Optimal Portfolio Choice," *The Journal of Portfolio Management*, vol. 19, no. 2, pp. 6–11, 1993.

[88] O. Ledoit and M. Wolf, "Honey, I Shrunk the Sample Covariance Matrix," *The Journal of Portfolio Management*, vol. 30, no. 4, pp. 110–119, 2004.

[89] P. Jorion, "Bayes-Stein Estimation for Portfolio Analysis," *The Journal of Financial and Quantitative Analysis*, vol. 21, no. 3, pp. 279–292, 1986.

[90] A. Moreira and T. Muir, "Volatility-managed portfolios," *The Journal of Finance*, vol. 72, no. 4, pp. 1611–1643, 2017.

[91] E. Qian, "Risk Parity Portfolios: Efficient Portfolios through True Diversification," tech. rep., PanAgora Asset Management White Paper, 9 2005.

[92] M. Lopez de Prado, "Building diversified portfolios that outperform out of sample:," *The Journal of Portfolio Management*, vol. 42, pp. 59–69, 07 2016.

[93] V. Demiguel, L. Garlappi, and R. Uppal, "Optimal Versus Naive Diversification: How Inefficient is the 1/N Portfolio Strategy?," *Review of Financial Studies*, vol. 22, no. 5, p. 1915–1953, 2009.

[94] N. Gârleanu and L. H. Pedersen, "Dynamic trading with predictable returns and transaction costs," *The Journal of Finance*, vol. 68, no. 6, pp. 2309–2340, 2013.

[95] O. Ledoit and M. Wolf, "Markowitz portfolios under transaction costs," Tech. Rep. 420, Department of Economics - University of Zurich, Oct. 2022.

[96] M. H. A. Davis and A. R. Norman, "Portfolio Selection with Transaction Costs," *Mathematics of Operations Research*, vol. 15, no. 4, pp. 676–713, 1990.

[97] M. S. Lobo, M. Fazel, and S. P. Boyd, "Portfolio optimization with linear and fixed transaction costs," *Annals of Operations Research*, vol. 152, pp. 341–365, 2007.

[98] T. I. Jensen, B. T. Kelly, S. Malamud, and L. H. Pedersen, "Machine Learning and the Implementable Efficient Frontier." `https://ssrn.com/abstract=4187217`, 2022. [Online; last accessed 10-Aug-2022].

[99] A. Frazzini, R. Israel, and T. J. Moskowitz, "Trading Costs." `https://ssrn.com/abstract=3229719`, 2018. [Online; last accessed 10-Aug-2022].

[100] M. Choey and A. S. Weigend, "Nonlinear trading models through Sharpe Ratio maximization," in *International Journal of Neural Systems*, pp. 417–431, World Scientific, 1997.

[101] J. S. Brush, "Comparisons and Combinations of Long and Long/Short Strategies," *Financial Analysts Journal*, vol. 53, no. 3, pp. 81–89, 1997.

[102] J. Walters, "The Black-Litterman Model in Detail." `https://ssrn.com/abstract=1314585`, 2014. [Online; last accessed 10-Aug-2022].

[103] S. Satchell and A. Scowcroft, "A demystification of the Black–Litterman model: Managing quantitative and traditional portfolio construction," *Journal of Asset Management*, vol. 1, pp. 138–150, 2000.

[104] H. S. Martin van der Schans, "Time-Dependent Black–Litterman," *Journal of Asset Management*, vol. 18, pp. 371–387, 2017.

[105] P. N. Kolm and G. Ritter, "On the Bayesian interpretation of Black–Litterman," *European Journal of Operational Research*, vol. 258, no. 2, pp. 564–572, 2017.

[106] H. A. Latané, "Criteria for Choice Among Risky Ventures," *Journal of Political Economy*, vol. 67, no. 2, pp. 144–155, 1959.

[107] E. Thorp, "The Kelly Criterion in Blackjack, Sports Betting, and the Stock Market," *Handbook of Asset and Liability Management*, vol. 1, 12 2008.

[108] J. Baz and H. Guo, "An asset allocation primer: connecting Markowitz, Kelly and Risk Parity," *PIMCO Quantitative Research*, 2017.

[109] P. A. Samuelson, "The "Fallacy" of Maximizing the Geometric Mean in Long Sequences of Investing or Gambling," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 68, no. 10, pp. 2493–2496, 1971.

[110] D. Bernoulli, "Specimen Theoriae Novae de Mensura Sortis. Translation by L. Sommer: "Exposition of a new theory on the measurement of risk"," *Econometrica*, vol. 22, no. 1, pp. 23–36, 1954 (original text 1738).

[111] I. Moscati, "Retrospectives: How Economists Came to Accept Expected Utility Theory: The Case of Samuelson and Savage," *Journal of Economic Perspectives*, vol. 30, pp. 219–36, May 2016.

[112] K. Ito, "On stochastic differential equations," *Memoirs of the American Mathematical Society*, no. 4, 1951.

[113] M. Spitznagel, *Safe Haven: Investing for Financial Storms*. Wiley, 2021.

[114] B. Donnelly, *The Art of Currency Trading: A Professional's Guide to the Foreign Exchange Market*. Wiley, 2019.

[115] J. Chen, W. Chen, C. Huang, S. Huang, and A. Chen, "Financial Time-Series Data Analysis Using Deep Convolutional Neural Networks," in *2016 7th International Conference on Cloud Computing and Big Data (CCBD)*, pp. 87–92, 2016.

[116] S. Borovkova and I. Tsiamas, "An ensemble of LSTM neural networks for high-frequency stock market classification," *Journal of Forecasting*, vol. 38, no. 6, pp. 600–619, 2019.

[117] A. Kondratyev, "Learning Curve Dynamics with Artificial Neural Networks." `https://ssrn.com/abstract=3041232`, 2018. [Online; last accessed 1-Apr-2019].

[118] J. Gonzalvez, E. Lezmi, T. Roncalli, and J. Xu, "Financial Applications of Gaussian Processes and Bayesian Optimization." `https://arxiv.org/pdf/1903.04841.pdf`, 2019. [Online; last accessed 02-Jan-2020].

[119] C. Leibig, V. Allken, M. Ayhan, P. Berens, and S. Wahl, "Leveraging uncertainty information from deep neural networks for disease detection," *Scientific Reports*, vol. 7, 2017.

[120] A. D. Cobb, M. D. Himes, F. Soboczenski, S. Zorzan, M. D. O'Beirne, A. G. Baydin, Y. Gal, S. D. Domagal-Goldman, G. N. Arney, and D. Angerhausen, "An Ensemble of Bayesian Neural Networks for Exoplanetary Atmospheric Retrieval." `https://arxiv.org/abs/1905.10659`, 2019. [Online; last accessed 20-Apr-2020].

[121] Z. Zhang, S. Zohren, and S. Roberts, "BDLOB: Bayesian Deep Convolutional Neural Networks for Limit Order Books." `https://arxiv.org/abs/1811.10041`, 2018. [Online; last accessed 20-Jul-2019].

[122] B. Lim, S. Zohren, and S. Roberts, "Enhancing Time-Series Momentum Strategies Using Deep Neural Networks," *The Journal of Financial Data Science*, vol. 1, 2019.

[123] N. Towers and A. N. Burgess, "Implementing trading strategies for forecasting models," in *Proceedings of Computational Finance*, The MIT Press, 1999.

[124] J. Gatheral and R. Oomen, "Zero-Intelligence Realized Variance Estimation," *Finance and Stochastics*, vol. 14, pp. 249–283, 2010.

[125] D. Nix and A. Weigend, "Estimating the mean and variance of the target probability distribution," in *Proceedings of 1994 IEEE International Conference on Neural Networks*, pp. 55–60, IEEE, 1994.

[126] G. Dorta, S. Vicente, L. Agapito, N. Campbell, and I. Simpson, "Structured Uncertainty Prediction Networks," in *IEEE Conference on Computer Vision and Pattern Recognition 2018*, IEEE, 2018.

[127] R. L. Russell and C. Reale, "Multivariate Uncertainty in Deep Learning." `https://arxiv.org/abs/1910.14215`, 2019. [Online; last accessed 03-May-2020].

[128] F. Chollet *et al.*, "Keras." `https://github.com/fchollet/keras`, 2015.

[129] Y. Gal, J. Hron, and A. Kendall, "Concrete Dropout," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, p. 3584–3593, 2017.

[130] A. D. Cobb, A. G. Baydin, A. M., and S. J. Roberts, "Introducing an Explicit Symplectic Integration Scheme for Riemannian Manifold Hamiltonian Monte Carlo." `https://arxiv.org/abs/1910.06243`, 2019. [Online; last accessed 10-May-2020].

[131] S. Giglio, B. Kelly, and D. Xiu, "Factor models, machine learning, and asset pricing," *Annual Review of Financial Economics*, vol. 14, no. 1, pp. 337–368, 2022.

[132] F. J. Fabozzi, S. M. Focardi, and P. N. Kolm, "Incorporating Trading Strategies in the Black-Litterman Framework," *The Journal of Trading*, vol. 1, no. 2, pp. 28–37, 2006.

[133] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012.

[134] P. Date and K. Ponomareva, "Linear and non-linear filtering in mathematical finance: A review," *IMA Journal of Management Mathematics*, vol. 21, 06 2010.

[135] D. Willner, C.-B. Chang, and K.-P. Dunn, "Kalman filter algorithms for a multi-sensor system," *1976 IEEE Conference on Decision and Control including the 15th Symposium on Adaptive Processes*, pp. 570–574, 1976.

[136] B. Khaleghi, A. Khamis, F. O. Karray, and S. N. Razavi, "Multisensor data fusion: A review of the state-of-the-art," *Information Fusion*, vol. 14, no. 1, pp. 28–44, 2013.

[137] F. Castanedo, "A review of data fusion techniques," *The Scientific World Journal*, 2013.

[138] J. K. Uhlmann, *Dynamic Map Building and Localization : New Theoretical Foundations*. PhD thesis, University of Oxford, 1995.

[139] E. O. Thorp, "The Kelly Criterion in Blackjack, Sports Betting, and the Stock Market," *Handbook of Asset and Liability Management*, vol. 1, pp. 385–428, 2006.

[140] S. Reece and S. Roberts, "Generalised Covariance Union: A Unified Approach to Hypothesis Merging in Tracking," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 46, 2010.

[141] A. P. Dempster, *Elements of continuous multivariate analysis*. Addison-Wesley Pub. Co., 1969.

[142] P. S. Maybeck, *Stochastic models, estimation, and control, Vol. 1*. Academic Press, New York, 1979.

[143] Y. Bar-Shalom, "On the track-to-track correlation problem," *IEEE Transactions on Automatic Control*, vol. 26, no. 2, pp. 571–572, 1981.

[144] S. J. Julier and J. K. Uhlmann, "A non-divergent estimation algorithm in the presence of unknown correlations," in *Proceedings of the 1997 American Control Conference*, vol. 4, pp. 2369–2373, IEEE, 1997.

[145] L. Chen, P. Arambel, and R. Mehra, "Fusion under unknown correlation - covariance intersection as a special case," in *Proceedings of the Fifth International Conference on Information Fusion. FUSION 2002.*, vol. 2, pp. 905–912, 2002.

[146] B. Noack, J. Sijs, M. Reinhardt, and U. D. Hanebeck, "Decentralized data fusion with inverse covariance intersection," *Automatica*, vol. 79, pp. 35–41, 2017.

[147] J. Sijs and M. Lazar, "State fusion with unknown correlation: Ellipsoidal intersection," *Automatica*, vol. 48, no. 8, pp. 1874–1878, 2012.

[148] B. Noack, J. Sijs, and U. Hanebeck, "Inverse covariance intersection: New insights and properties," in *Proceedings of the 20th International Conference on Information Fusion*, 2017.

[149] J. K. Uhlmann, "Covariance consistency methods for fault-tolerant distributed data fusion," *Information Fusion*, vol. 4, pp. 201–215, 2003.

[150] A. Kuntsevich and F. Kappel, "SolvOpt: The Solver For Local Nonlinear Optimization Problems," *Institute for Mathematics, University of Graz*, 1997.

[151] O. Bochardt, R. Calhoun, J. K. Uhlmann, and S. J. Julier, "Generalized information representation and compression using covariance union," *2006 9th International Conference on Information Fusion*, pp. 1–7, 2006.

[152] S. J. Julier, J. K. Uhlmann, and D. Nicholson, "A method for dealing with assignment ambiguity," *Proceedings of the 2004 American Control Conference*, vol. 5, pp. 4102–4107 vol.5, 2004.

[153] C. S. Asness, T. J. Moskowitz, and L. H. Pedersen, "Value and Momentum Everywhere," *The Journal of Finance*, vol. 68, no. 3, pp. 929–985, 2013.

[154] B. Efron, "Better bootstrap confidence intervals," *Journal of the American Statistical Association*, vol. 82, no. 397, pp. 171–185, 1987.

[155] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics Bulletin*, vol. 1, no. 6, pp. 80–83, 1945.

[156] E. Thorp, "Statistical Arbitrage – Part II," *Wilmott*, pp. 48–49, 11 2004.

[157] C. Krauss, "Statistical Arbitrage Pairs Trading Strategies: Review and Outlook," *Journal of Economic Surveys*, vol. 31, no. 2, pp. 513–545, 2017.

[158] M. Avellaneda and J.-H. Lee, "Statistical arbitrage in the US equities market," *Quantitative Finance*, vol. 10, no. 7, pp. 761–782, 2010.

[159] S. M. Focardi, F. J. Fabozzi, and I. K. Mitov, "A new approach to statistical arbitrage: Strategies based on dynamic factor models of prices and their performance," *Journal of Banking & Finance*, vol. 65, pp. 134–155, 2016.

[160] S. Sarmento and N. Horta, "Enhancing a Pairs Trading strategy with the application of Machine Learning," *Expert Systems with Applications*, vol. 158, 5 2020.

[161] F. Gatta, C. Iorio, D. Chiaro, F. Giampaolo, and S. Cuomo, "Statistical arbitrage in the stock markets by the means of multiple time horizons clustering," *Neural Computing and Applications*, 2023.

[162] G. Vidyamurthy, *Pairs Trading: Quantitative Methods and Analysis*. John Wiley & Sons, Hoboken, N.J., 2004.

[163] J. Guijarro-Ordonez, "High-dimensional Statistical Arbitrage with Factor Models and Stochastic Control," *Applied Mathematical Finance*, vol. 26, no. 4, pp. 328–358, 2019.

[164] Z. Zhao and D. P. Palomar, "Mean-Reverting Portfolio With Budget Constraint," *IEEE Transactions on Signal Processing*, vol. 66, no. 9, pp. 2342–2357, 2018.

[165] Z. Zhao, R. Zhou, and D. P. Palomar, "Optimal Mean-Reverting Portfolio With Leverage Constraint for Statistical Arbitrage in Finance," *IEEE Transactions on Signal Processing*, vol. 67, no. 7, pp. 1681–1695, 2019.

[166] J. Stübinger, B. Mangold, and C. Krauss, "Statistical arbitrage with vine copulas," *Quantitative Finance*, vol. 18, no. 11, pp. 1831–1849, 2018.

[167] D. Lautier, A. Javaheri, and A. Galli, "Filtering in Finance," *Wilmott*, pp. 2–18, 05 2003.

[168] P. Date and K. Ponomareva, "Linear and non-linear filtering in mathematical finance: a review," *IMA Journal of Management Mathematics*, vol. 22, no. 3, pp. 195–211, 2011.

[169] R. Elliott, J. Van Der Hoek, and W. Malcolm, "Pairs trading," *Quantitative Finance*, vol. 5, no. 3, pp. 271–276, 2005.

[170] K. Triantafyllopoulos and G. Montana, "Dynamic modeling of mean-reverting spreads for statistical arbitrage," *Computational Management Science*, vol. 8, p. 23–49, 2011.

[171] T. Tsagaris, *Adaptive Regression Methods with Application to Streaming Financial Data*. PhD thesis, Imperial College, 2010.

[172] C. E. de Moura, A. Pizzinga, and J. Zubelli, "A pairs trading strategy based on linear state space models and the Kalman filter," *Quantitative Finance*, vol. 16, no. 10, pp. 1559–1573, 2016.

[173] G. Zhang, "Pairs trading with general state space models," *Quantitative Finance*, vol. 21, no. 9, pp. 1567–1587, 2021.

[174] Y. Han, "Asset Allocation with a High Dimensional Latent Factor Stochastic Volatility Model," *Review of Financial Studies*, vol. 19, no. 1, pp. 237–271, 2006.

[175] S. Chib, F. Nardari, and N. Shephard, "Analysis of high dimensional multivariate stochastic volatility models," *Journal of Econometrics*, vol. 134, no. 2, pp. 341–371, 2006.

[176] S. Julier and J. Uhlmann, "Unscented filtering and nonlinear estimation," *Proceedings of the IEEE*, vol. 92, pp. 401 – 422, 04 2004.

[177] E. A. Wan and R. van der Merwe, "The unscented Kalman filter for nonlinear estimation," in *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium*, pp. 153–158, IEEE, 2000.

[178] S. Gu, B. Kelly, and D. Xiu, "Autoencoder Asset Pricing Models," *Journal of Econometrics*, vol. 222, no. 1, Part B, pp. 429–450, 2021.

[179] T. Spears, S. Zohren, and S. Roberts, "View Fusion Vis-à-Vis a Bayesian Interpretation of Black–Litterman for Portfolio Allocation," *The Journal of Financial Data Science*, vol. 5, no. 3, pp. 23–49, 2023.

[180] T. Spears, S. Zohren, and S. Roberts, "Investment Sizing with Deep Learning Prediction Uncertainties for High-Frequency Eurodollar Futures Trading," *The Journal of Financial Data Science*, vol. 3, no. 1, pp. 57–73, 2021.

[181] A. W. Lo and A. C. MacKinlay, "When are Contrarian Profits Due to Stock Market Overreaction?," *The Review of Financial Studies*, vol. 3, no. 2, pp. 175–205, 1990.

[182] D. Blitz, G. Baltussen, and P. van Vliet, "When Equity Factors Drop Their Shorts," *Financial Analysts Journal*, vol. 76, no. 4, pp. 73–99, 2020.

[183] C. Howard, "Less is More? Reducing Biases and Overfitting in Machine Learning Return Predictions." `https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4497739`, 2023. [Online; last accessed 1-Aug-2023].

[184] P. Andreini, C. Izzo, and G. Ricco, "Deep Dynamic Factor Models," in *Working Papers 2023-08*, Center for Research in Economics and Statistics, 2023. [Online; last accessed 1-Aug-2023].

[185] S. Aleti, T. Bollerslev, and M. Siggaard, "Intraday Market Return Predictability Culled from the Factor Zoo." `https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4388560`, 2023. [Online; last accessed 1-Aug-2023].

[186] R. J. Elliott and R. Bradrania, "Estimating a regime switching pairs trading model," *Quantitative Finance*, vol. 18, no. 5, pp. 877–883, 2018.

[187] H. Attias, "A variational bayesian framework for graphical models," in *Proceedings of the 12th International Conference on Neural Information Processing Systems*, NIPS'99, (Cambridge, MA, USA), p. 209–215, MIT Press, 1999.

[188] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer-Verlag, 2006.

[189] P. J. Huber, "Robust Estimation of a Location Parameter," *Annals of Mathematical Statistics*, vol. 35, pp. 73–101, 1964.

[190] P. Lebrun, *Bayesian design space applied to pharmaceutical development*. PhD thesis, Université de Liège, Belgique, 2012.

[191] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[192] S. Roberts, M. Osborne, M. Ebden, S. Reece, N. Gibson, and S. Aigrain, "Gaussian processes for time-series modelling," *Philosophical transactions. Series A, Mathematical, physical, and engineering sciences*, vol. 371, p. 20110550, 02 2013.

[193] A. Ustimenko, L. Ostroumova Prokhorenkova, and A. Malinin, "Uncertainty in Gradient Boosting via Ensembles." `https://arxiv.org/abs/2006.10562`, 2020. [Online; last accessed 22-Apr-2021].

[194] S. Lahlou, M. Jain, H. Nekoei, V. Butoi, P. Bertin, J. Rector-Brooks, M. Korablyov, and Y. Bengio, "DEUP: Direct Epistemic Uncertainty Prediction." `https://arxiv.org/abs/2102.08501`, 2022. [Online; last accessed 04-Dec-2022].