ALGORITHMIC TRADING *and* REINFORCEMENT LEARNING

Robust methodologies for AI in finance

THOMAS SPOONER

July 2021

They say that "he who flies highest, falls farthest" – and who am I to argue?
But we can't forget that "he who doesn't flap his wings, never flies at all."
And with that, I'll stop trying to convince myself that I can't fail;
how dull the whole thing would be if that were true.

— Hunter S. Thompson [174]

The way you not fall off, is you remain a student of the game.
At all times, in anything that you doing.

— Freddie Gibbs [64]

COLOPHON

This document was typeset using the typographical look-and-feel `classicthesis` developed by André Miede and Ivo Pletikosić. The style was inspired by Robert Bringhurst's seminal book on typography "*The Elements of Typographic Style*". `classicthesis` is available here:

https://bitbucket.org/amiede/classicthesis/

Algorithmic Trading and Reinforcement Learning, Dedicated to the loving memory of my grandfather Robert “Papa” Pill.

1922 – 2014

ACKNOWLEDGMENTS

Taking on a PhD is an enormous undertaking, mentally, emotionally and, at times, physically. At the beginning, I'm not sure that I knew exactly to what extent nor what I was getting myself into. Catalysing naïve enthusiasm into a finished product is a non-trivial process and requires time and persistence. But, above all else, this thesis could not have been achieved without the support of a wonderful group of friends, family and colleagues. There is no doubt in my mind that this wouldn't have been possible without them.

First and foremost, I would like to thank my advisor and mentor **Rahul Savani** who has helped me beyond measure both in and out of academia. The patience and stoicism he has afforded me cannot be understated. His drive for excellence and precision has formed the foundations of my own principles as a researcher. And the opportunities and trust he has placed in me over these four years has been humbling. I am eternally grateful for the path you have now set me on. I only hope that I can repay the favour in kind as we continue to collaborate in the future. Alongside my advisor, **John Fearnley** was one of the first people whom I collaborated with — going back as far as my Master's project. His advice and support at these early stages of my development and during the PhD itself have been pivotal. I cannot thank him enough for this.

A number of other staff at the University of Liverpool have also helped me a great deal. On the academic side: my secondary advisor, **Karl Tuyls**; collaborator and inspiration, **Frans Oliehoek**; advisor, **Martin Gairing**. I have also greatly appreciated being a part of the Economics and Computation group. On the support side, I'd like to thank **Paddy Colleran**, **Alison Goodyear**, **Dave Shield**, and **Elaine Smith** for putting up with me these four years. I *literally* couldn't have done it without you all. My work with the Institute of Infection and Global Health (an enjoyable tangent from finance) was a wonderful experience as well. To **Matthew Baylis**, **Anne Jones** and **Joanne Turner**: it was a pleasure to work with you. I learnt a great deal about the value of interdisciplinary research, and the importance of collaboration. On a similar vein, I'd like to mention **Sam Devlin** and **Nolan Bard** for their company at conferences. I will always appreciate having friends who are willing to talk about things other than our work at these events.

Fellow students — **Greg**, **Argy**, **James**, **Jacopo**, alongside many others — provided invaluable emotional and technical support that I couldn't have done without. They're willingness to listen to my moaning and diatribes over the years will not be forgotten. Similarly, I would not have managed without the friendships and welcome distraction that the UoL Karate club provided me. **Ash**, **Josie**, **Harry**, **Matthew**, **Michael**, and again, many others — you've been great. I cannot wait to compete with you again or, for that matter, against you. Oss!

The various colleagues during my time as a consultant also taught me a great deal. Not only about industry, startups, the world of work and research, but also about humility and the importance of self reflection. My thanks go to **Andreas Koukorinis** for affording me the opportunity to learn from the amazing team that was at Stratagem. Many of these people are still good friends of mine today: **Simon**, **Caelin**, **Tony**, **Thomas** and **Nicolas**, to name some.

Now, in more recent times, I have began my new journey at JP Morgan's AI Research group. To **Sammy**, **Josh** and **Mahmoud**: I cannot thank you enough for seeing something in me at ICML and the opportunity you have given me. My team is wonderful, and finishing my thesis with your support has meant a great deal to me. **Sumitra**, you have been, and are, an incredible manager who has made me feel welcome and valued; I am truly excited about the future. Similarly, **Daniele**, you have made working with the London team a joy. There are so many other I want to thank, but in particular: **Nelson** and **Mengda**; **Danial** and **Cecilia**; **Prashant**, **Tucker** and **Manuela**.

Outside of work and research entirely, I have relied on many a person for moral and emotional support. To my friends **Oba** and **Eduardo**: you know exactly how much you mean to me. My life will forever be the better for meeting you both. This includes the rest of you too: **Bo**, **Justin** and **Ryan**. To poor **John** who's had to put up with me since undergraduate: you've been there consistently and unwaveringly, and I cannot thank you enough. I thank all my friends and amazing flat mates in London, both old and new: **Alan**, **Manon**, **Niall** and **Tash**; **Francis**, **Nadia** and **Rachel**. I cannot wait for what 2021 brings with the whole London crew, you're all wonderful people: **Sarah** (thanks for letting me couch surf), **James**, **Mathilde**, **Esteban**, **Adam**, **Lara** and **Lolly**. All my friends back home.

Finally, I would like to thank my parents — **Stuart** and **Chris** — who have supported me throughout the PhD and encouraged me to pursue my ambitions. Without them I know for certain that I wouldn't be where I am today. As a firm believer that nurture triumphs over nature, the responsibility and my gratitude is with you. And, of course, last but not least, my two sisters, **Molly** and **Florence**, who have been there for me on many occasions when I've needed advice. I love you all.

To everyone, mentioned above or not: *thank you.*

ABSTRACT

The application of reinforcement learning (RL) to algorithmic trading is, in many ways, a perfect match. Trading is fundamentally a problem of making decisions under uncertainty, and reinforcement learning is a family of methods for solving such problems. Indeed, many researchers have explored this space and, for the most, validated RL, its ability to find effective solutions and its importance in studying the behaviour of agents in markets. In spite of this, many of the methods available today fail to meet expectations when evaluated in *realistic environments*. There are a number of reasons for this: partial observability, credit assignment and non-stationary dynamics. Unlike video games, the state and action spaces are often unstructured and unbounded, which poses challenges around knowledge representation and task invariance. As a final hurdle, traders also need RL to be able to handle risk-sensitive objectives with solid human interpretation to be used reliably in practice. All of these together make for an exceptionally challenging domain that poses fascinating questions about the efficacy of RL and the techniques one can use to address these issues. This dissertation makes several contributions towards two core themes that underlie the challenges mentioned above. The first, *epistemic uncertainty*, covers modelling challenges such as misspecification and robustness. The second relates to *aleatoric risk* and safety in the presence of intrinsic randomness. These will be studied in depth, for which we summarise, below, the key findings and insights developed during the course of the PhD.

The first part of the thesis investigates the use of data and historical reconstruction as a platform for learning strategies in limit order book markets. The advantages and limitations of this class of model are explored and practical insights provided. It is demonstrated that these methods make minimal assumptions about the market's dynamics, but are restricted in terms of their ability to perform counterfactual simulations. Computational aspects of reconstruction are discussed, and a highly performant library provided for running experiments. The second chapter in this part of the thesis builds upon historical reconstruction by applying value-based RL methods to market making. We first propose an intuitive and effective reward function for both risk-neutral and risk-sensitive learning and justify it through variance analysis. Eligibility traces are shown to solve the credit assignment problem observed in past work, and a comparison of different state-of-the-art algorithms (each with different assumptions) is provided. We then propose a factored state representation which incorporates market microstructure and benefits from improved stability and asymptotic performance compared with benchmark algorithms from the literature.

In the second part, we explore an alternative branch of modelling techniques based on explicit stochastic processes. Here, we focus on policy gradient methods, introducing a family of likelihoods functions that are effective in trading domains and studying their properties. Four key problem domains are introduced along with their solution concepts and baseline methods. In the second chapter of part two, we use adversarial reinforcement learning to derive epistemically robust strategies. The market making model of Avellaneda and Stoikov [11] is recast as a zero-sum, two player game between the market maker, and the market. We study the theoretical properties of a one-shot projection, and empirically evaluate the dynamics of the full stochastic game. We show that the resulting algorithms are robust to discrepancies between train and test time price/execution dynamics, and that the resulting strategies dominate performance in all cases. The final results chapter addresses the intrinsic risk of trading and portfolio management by framing the problems explicitly as constrained Markov decision processes. A downside risk measure based on lower partial moments is proposed, and a tractable linear bound derived for application in temporal-difference learning. This proxy has a natural interpretation and favourable variance properties. An extension of previous work to use natural policy gradients is then explored. The value of these two techniques is demonstrated empirically for a multi-armed bandit and two trading scenarios. The results is a practical algorithm for learning downside risk-averse strategies.

CONTENTS

I PROLOGUE

II DATA-DRIVEN TRADING

IV EPILOGUE

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

LO	Limit Order
MM	Market Maker
MO	Market Order
NE	Nash Equilibrium
RL	Reinforcement Learning
TD	Temporal-Difference
ARL	Adversarial Reinforcement Learning
LOB	Limit Order Book
LPM	Lower Partial Moment
MDP	Markov Decision Process
RCPO	Reward Constrained Policy Optimisation
MMMW	Market-Making Multiplicative Weights
NRCPO	Natural Reward Constrained Policy Optimisation

Part I

PROLOGUE

1 INTRODUCTION

Finance is the most pervasive global industry today, and it is in the throws of a technological renaissance that is largely driven by the advent of machine learning. Indeed, many of the major banks are now investing heavily in artificial intelligence research, driven by the complexities observed in real-world problems. This is a powerful idea — historically speaking, some of the most important discoveries in science have come from solving practical problems rather than focusing on abstractions in isolation. Pasteurisation contributed to the germ theory of disease, Bayesian statistics and causal modelling challenged the frequentist dogma of Francis Galton and Karl Pearson, revolutionising our understanding of probability, and Robert Brown's study of Clarkia pulchella pollen lead to the development of the Wiener process which is at the core of stochastic processes today. All of these examples support a key conclusion: that a "top-down" approach to science is a vital element of research.

Perez [126] classified this as one of five key technological revolutions, likening it to the industrial revolution.

It is believed that Thomas Bayes first proposed conditional probability as a means of proving the existence of "the Deity" [14, 124].

One of the most relevant fields of finance where this is true is option pricing. There, the industry faced the question of how to uniquely value derivative products which were growing in prevalence during late 1900s. The contributions towards this pursuit of Fischer Black, Myron Scholes and Robert C. Merton (BSM) [23, 113] — as well as Louis Bachelier before them [12] — cannot be understated. Before this landmark, eponymous result, options were traded in a mostly heuristic fashion by experienced traders, and while it has been shown that these markets were comparably efficient to those we have nowadays (relative to the BSM value) [39, 117], the logic driving decision-making was somewhat opaque. Reliance on human operators is severely limiting and even dangerous in certain circumstances, much in the same way that human drivers pose undue risk to pedestrians. One would be right to conclude, then, that BSM paved the way towards increased automation in finance, replacing abstract intuition with concrete, verifiable methodology. Of course, option pricing is not the only domain in finance that has experienced a transition towards scientific rigour.

Trading has, over the last 30 years, become a powerhouse of computational ingenuity, and algorithmic trading now reigns supreme over the traditional methods of exchange. Yet, despite the advances in high-frequency trading platforms and their systematisation, trading as a whole is still exposed to two key forms of human bias:

(i) The classic form of bias is driven by *direct, human interaction in which the operator controls a large part of the process itself*; i.e. using machine learning to derive signals, but leaving the decision-making and implementation entirely to the trader.

(ii) The more subtle — and far more dangerous — source of human bias enters in the form of *mechanistic assumptions made during the development of a strategy*. Whether this be assuming that prices evolve in a particular way, or that transactions occur with a given probability, the result is anything but transparent and often leads to unexpected results. This problem can often be traced back to weak backtesting, or even the "brainchild" effect in which a proponent of a strategy has an emotional bias towards having it rolled out.

This presents a dichotomy. On the one hand, increased automation reduces the bias and risk due to explicit human error, but on the other hand algorithmic trading strategies are entirely dependent on the assumptions underlying the model, which were again chosen by a human. If these are incorrect, or the model parameters are poorly calibrated, then the discrepancy between the *expected market* and the *true market* — i.e. the epistemic uncertainty — will be great. Fixed strategies are also incapable of adapting to new market conditions that were not expressly anticipated in advance, a task which humans are very much capable of. What, then, can we do to minimise these sources of bias in the journey towards automation?

In this thesis, we explore the intersection of reinforcement learning (RL) and algorithmic trading. The emphasis will be on robustness, both in terms of model specification, and in terms of the objective used to define "optimality". In doing so, the aim is to provide an answer to the questions posed above by first building intuition — theoretical and empirical — and then proposing a suite of methods that address the limitations of past approaches. Unsurprisingly, we will take a "top-down" approach, posing problems, exploring the space of solutions and providing a practical guide to researchers and practitioners alike.

1.1 MOTIVATION

The vast majority of literature on trading over the last 20+ years — for market making, optimal execution and inter-temporal portfolio selection — relies on the theory of optimal control. Problems are described using stochastic processes and are solved using some variant of dynamic programming [31]. The solutions are often exact, but the underlying models themselves are only ever approximations of the true market dynamics. This choice of model is always constrained by solubility which limits and, arguably, introduces a fundamental bias in the space of research. One can't help but ask questions such as:

(i) What would the solutions look like if we had access to a more realistic model?

(ii) Is the language of mathematics currently equipped with the tools for exploring this question?

(iii) Can we remove the need to specify a model in the first place?

(iv) And is it possible to bridge the gap between analytical methods and simulation-based methods?

The answer to the latter two is, for the most part, yes, and it is precisely these themes that we will explore in the course of the thesis.

Another key motivator for this work is to address the distinct lack of simplicity in many related papers. To quote Einstein [56]:

> It can scarcely be denied that the supreme goal of all theory is to make the irreducible basic elements as simple and as few as possible without having to surrender the adequate representation of a single datum of experience.

This is a classic rendition of the principles underlying Occam's razor, and the variations therein of Claudius Ptolemy, Isaac Newton and Bertrand Russel, to name a few. Put in my own words: the hallmark of true science is in the distillation of complexity into simplicity. That is not to say that the process of discovery itself is easy — as it rarely is — but rather that the end result teach something new in an effective way. The more convoluted an approach becomes, the less insight it provides and the less practical the solution. This is incredibly important in top-down research where the purpose is to solve a well-defined problem. In this thesis, I aim to do exactly this: provide insight alongside concrete, performant solutions that are simple to implement without sacrificing value or merit.

1.1.1 *Pioneers*

Two papers in particular inspired the thought processes behind this work. The first was the pioneering work of Chan and Shelton [40] who, to the best of our knowledge, were the first to explore the use of reinforcement learning (RL) for systematic market making. Their paper proposed a simplified model of the market and experimented with a selection of state-of-the-art-at-the-time learning techniques. While the work suffered from the same limitations that we highlighted previously, their methodology was novel and paved the way towards greater application of approximate dynamic programming in this space. The second paper, published some 5 years after, was authored by Nevmyvaka, Feng, and Kearns [119] and focused instead on optimal execution. Unlike its predecessors, this work took a direct, data-driven approach, opting to use limit order book (LOB) reconstruction to replay past event sequences that were known to have occurred. They demonstrated remarkably good results and helped catapult RL into the mainstream attention of financiers. To this day, their work and methodology is considered a template for research in this area.

Together, these two papers set the stage for a now rapidly growing field of research. Not only did they demonstrate the power of RL, but they also highlighted just how interesting this domain is, and it's value for analysing the performance of our learning algorithms. Non-stationary environment dynamics, high variance and delayed reward signals, knowledge representation, model misspecification and epistemic uncertainty, to name a few, are all challenges that are intrinsic to the intersection of RL and algorithmic trading. Taken independently, each of these examples are known in the artificial intelligence community to be hard to address and have yet to be solved as a result. While in this thesis we do not consider all of these, we aim to address some of the important issues and build upon the innovative and influential contributions of Chan and Shelton, and Nevmyvaka, Feng, and Kearns, as well as the many others who have since moved this field and our understanding forwards significantly.

1.2 THE THESIS

The problems with existing work, as discussed above fall under two main topics:

(i) model discrepancy; and

(ii) validity and interpretability of objectives.

Here "validity" refers to the degree to which the solution matches human notions of risk-sensitivity. While addressing these core themes, we shall discover that the two topics can also be grouped under **epistemic** and **aleatoric** sources uncertainty, respectively. This gives rise to an axis of sorts, with the former on the x-axis, and the latter on the y-axis. The objective of this thesis is to cover all four quadrants of the matrix, and in so doing answer the following research questions:

	Aleatoric Uncertainty	
Epistemic Uncertainty	**Q1**: How do we exploit data in LOB reconstruction to minimise train-test model ambiguity?	**Q2**: What techniques are required to apply RL to realistic settings and promote risk-sensitivity via the reward function?
	Q3: Is is possible to derive epistemically robust strategies from improperly specified models?	**Q4**: Can risk-sensitive RL be extended to support human-interpretable objectives that aren't possible to specify in the reward?

The questions posed above are largely independent. For example, both **Q2** and **Q4** could be applied in either data-driven or model-driven learning. We maintain this separation, which gives rise to Part II and Part III of the thesis (each covering one column of the matrix), due to

(i) their suitability with respect to the class of algorithms used in each setting;

(ii) the monotonic increase in complexity of the contributions throughout the thesis, which largely follows the ordering of **Q1** to **Q4**; and

(iii) the chronology of the research conducted and (in some cases) published during the course of the PhD.

1.3 STRUCTURE

The first part of the thesis is focused on providing the technical background needed to understand the contributions presented herein. Chapter 2 first introduces the family of methods known of as reinforcement learning (RL) and the algorithms that we build upon for solving decision-making problems. This includes the theoretical foundations used to define value functions and optimality, and the two main subfields of prediction and control. Chapter 3 then covers algorithmic trading and quantitative finance, outlining a general framework for how to define portfolios, assets and strategies in both discrete- and continuous-time. We then contextualise the role played by RL as a modern approach to deriving strategies for many of the important problems studied in finance.

The remainder of the thesis is broken up into two mostly independent parts, one tackling data-driven approaches, the other covering model-driven approaches.

DATA-DRIVEN TRADING

Chapter 4 provides a thorough decomposition of the computational methods associated with LOB reconstruction. We begin by presenting a set of techniques for "replaying" historical events as they would have happened in the LOB. A family of indicators is then introduced for defining predictors of future market states. This sets the scene for engineering and evaluating trading strategies.

Chapter 5 explicates the use of RL for learning automated market making strategies in LOB markets. We propose and validate a novel reward function that promotes robust behaviour, and a factored knowledge representation to stabilise learning. A consolidation of these techniques is presented and shown to produce effective market making strategies, outperforming a recent approach from the online learning community.

MODEL-DRIVEN TRADING

Chapter 6 explores some of the key models in quantitative finance and the intersection with RL. Parallels are drawn between solution methods and a formalism presented for translating stochastic optimal control representations into the language of artificial intelligence and RL. This includes an analysis of policy distributions and their stability when used in trading applications with gradient-based methods.

Chapter 7 addresses the concern of epistemic risk in model-driven trading. The market making problem is re-framed as a zero-sum game between the market maker and the market. An instantiation of adversarial reinforcement learning (ARL) is then used to train robust strategies which outperform their traditional counterparts. We prove in several special cases that a one-shot variant of the full stochastic game exhibits Nash equilibria that correspond to those observed empirically.

Chapter 8 focuses on the problem of aleatoric risk in model-driven trading. A solution based on partial moments — a more "human" measure of risk — is proposed, theoretically justified and empirically validated on three benchmark domains. This takes the form of a bound on the general value function used to estimate risk, and a novel extension of risk-sensitive RL methods to use natural gradients.

The main content of the thesis concludes with Part IV which binds the contributions together, providing context and final remarks in Chapter 9. This final chapter includes a discussion of a number of key future research directions that would be of great interest and value to study going forward. An illustration of the flow of the thesis can be found in Figure 1.1.

1.4 PUBLISHED MATERIAL

1.4.1 *Papers*

The central themes and contributions put forward in this thesis were derived primarily from the following papers (published and preprint):

CHAPTER 5 Thomas Spooner, John Fearnley, Rahul Savani, and Andreas Koukorinis. 'Market Making via Reinforcement Learning'. In: *Proc. of AAMAS*. 2018, pp. 434–442

CHAPTER 7 Thomas Spooner and Rahul Savani. 'Robust Market Making via Adversarial Reinforcement Learning'. In: *Proc. of IJCAI*. Special Track on AI in FinTech. July 2020, pp. 4590–4596

CHAPTER 8 Thomas Spooner and Rahul Savani. 'A Natural Actor-Critic Algorithm with Downside Risk Constraints'. URL: https://arxiv.org/abs/2007.04203

Miscellaneous

Not all the research conducted during the course of the PhD fits within the scope of this thesis. Other projects, including the study of some game-theoretic aspects of

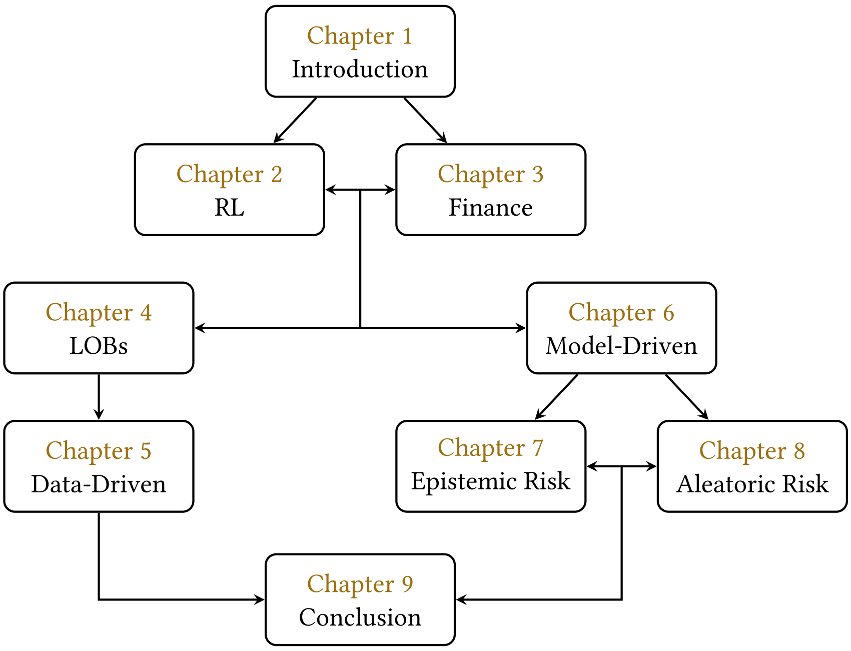


Figure 1.1: Chapter dependence diagram illustrating the suggested routes to be taken through the thesis. This diagram was heavily inspired by the excellent thesis of Grondman [73].

generative adversarial networks, as well as an application of Bayesian optimisation to epidemiological control, are mentioned here for posterity. The latter also yielded a publication which we mention below:

> Thomas Spooner, Anne E Jones, John Fearnley, Rahul Savani, Joanne Turner, and Matthew Baylis. 'Bayesian optimisation of restriction zones for bluetongue control'. In: *Scientific Reports* 10.1 (2020), pp. 1–18

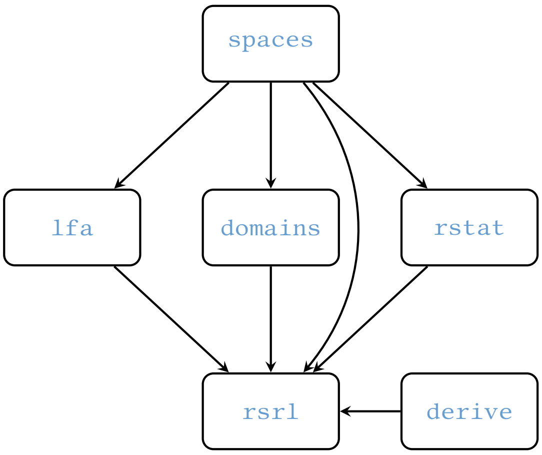


Figure 1.2: Dependency diagram for the ecosystem of crates developed during the course of the PhD.

1.4.2 *Code*

During the PhD, time was spent developing a robust ecosystem for reinforcement learning (RL) research in the Rust programming language [141]. What began as a side-project rapidly became something of an obsession and has since grown considerably. All of the results presented in Part III, for example, were generated using this suite of tools. To date, a total of six packages — amongst some other minor contributions — have been published on the Rust crate registry at https://crates.io with a total of $\sim 20\text{k}$ downloads. The core crate, `rsrl`, has even been featured in a book [21] on practical use of Rust for machine learning and on a data science podcast [140]. A dependency diagram between all six crates is illustrated in Figure 1.2, and a brief description of their purposes below:

`spaces` provides set/space primitives for defining machine learning problems akin to the `gym.spaces` module used ubiquitously in the Python community.

`lfa` is a set of native Rust implementations of linear function approximators. It includes various basis functions and highly efficient, type-generalised (i.e. trait-level) implementations.

`rstat` is a crate containing implementations of probability distributions and statistics in Rust, with integrated fitting routines, convolution support and mixtures.

`rsrl` is a fast, extensible reinforcement learning framework in Rust. It supports both value-based and policy gradient methods, as well as efficient code for both prediction and control. The framework revolves around an actor system, such that each agent can be deployed and interacted with through the same interface. It includes a number of sub-crates:

- **`domains`** contains toy domains for RL research with collection primitives for transitions and trajectories; this crate, too, is very similar to that of `gym`.
- **`derive`** is a core (under-the-hood) crate with procedural macros for simplifying many aspects of the `rsrl` codebase.

2 REINFORCEMENT LEARNING

2.1 MARKOV DECISION PROCESSES

Markov decision processes (MDPs) are a parsimonious framework for describing decision-making problems in which an *agent* interacts with an *environment* in order to achieve some goal. At each time $t \in \{0\} \cup \mathbb{N}$ the agent observes the current state of the system s_t and selects a new action to take a_t. At the next time step, $t+1$, the agent receives a scalar numerical reward r_{t+1}, and arrives in a new state s_{t+1}. This series of decisions and innovations gives rise to temporal sequences called *trajectories* (or *histories*) denoted by

$$h_{t:(t+n)} \doteq (s_t, a_t, s_{t+1}, \dots, s_{t+n-1}, a_{t+n-1}, s_{t+n}), \tag{2.1}$$

with $h_t \doteq h_{t:\infty}$. Formally, this description is known as an infinite-horizon MDP in discrete-time [134], comprising: a state space $s \in \mathcal{S}$, (state-dependent) action space $a \in \mathcal{A}(s) \subseteq \mathcal{A}$, set of rewards $r \in \mathcal{R} \subseteq \mathbb{R}$, and corresponding space of n-step trajectories given by the product $h_{t:(t+n)} \in \mathcal{H}_n = (\mathcal{S}, \mathcal{A})^n \times \mathcal{S}$, with $\mathcal{H} \doteq \mathcal{H}_\infty$.

Continuous-time MDPs exist but are considerably more involved. See e.g. Bertsekas and Tsitsiklis [19].

In this thesis we assume that the state space is continuous — i.e. $\mathcal{S} \subseteq \mathbb{R}^N$ for some $N > 0$ — but allow for either discrete or continuous action spaces as required. The behaviour of such an MDP is described by two key properties: an initial state distribution with density function $d_0 : \mathcal{S} \to \mathbb{R}_+$, for which we require the integral condition $\int_{\mathcal{S}} d_0(s)\, \mathrm{d}s = 1\ \forall\, s \in \mathcal{S}$; and a dynamics function $p : \mathcal{S} \times \mathcal{R} \times \mathcal{S} \times \mathcal{A} \to \mathbb{R}_+$ which defines a joint distribution over the set of successor states and rewards for any state-action pair. As with the initial state distribution, we require that the dynamics process is well-defined, such that

$$\int_{\mathcal{S}} \int_{\mathcal{R}} p(s', r \,|\, s, a)\, \mathrm{d}r\, \mathrm{d}s' = 1 \quad \forall a \in \mathcal{A}(s),\ s \in \mathcal{S}.$$

The probability of transitioning from a state s_{t-1} to a state s_t *in the region* $\mathcal{S}_t \subseteq \mathcal{S}$, following an action a_t, may thus be expressed as

$$\Pr\{s_t \in \mathcal{S}_t \,|\, s_{t-1} = s,\ a_{t-1} = a\} = \int_{\mathcal{S}_t} p(s_t \,|\, s, a)\, \mathrm{d}s_t, \tag{2.2}$$

where we denote by $p(s' \,|\, s, a)$ the *state-transition kernel* (see Klenke [92])

$$p(s' \,|\, s, a) = \int_{\mathcal{R}} p(s', r \,|\, s, a)\, \mathrm{d}r. \tag{2.3}$$

By virtue of $\mathcal{S}$ being continuous, the probability of transitioning to any single state s' is zero. See Section 5.1 of Dekking, Kraaikamp, Lopuhaä, and Meester [51].

This function maps each state-action pair to a probability measure defined on the state-space. A key property of (2.3) is that there is no direct dependence on states or actions preceding s and a. This means that the probability of transitioning into a new state s' is independent of the history (i.e. the path leading up to s), conditioned on the present,

$$\Pr\left\{s_t \in \mathcal{S}_t \,\middle|\, h_{0:(t-1)}\right\} = \Pr\{s_t \in \mathcal{S}_t \,|\, s_{t-1} = s,\ a_{t-1} = a\}. \tag{2.4}$$

This is known as the *Markov* property.

Hence Markov *decision process.*

From these core definitions, we can derive just about any quantity needed in RL. Of particular relevance are reward functions, which are defined as expectations over rewards with respect to the transition dynamics and policy distribution. For example,

$r : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ defines the average reward generated by a given state-action-next state triple:

$$r(s, a, s') \doteq \int_{\mathcal{R}} p\big(r, s' \,\big|\, s, a\big)\; r \; dr, \tag{2.5}$$

We can then define a function which maps state-action pairs to expected rewards, $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, by marginalising over the possible successor states, leading to the integral equation

$$r(s, a) \doteq \int_{\mathcal{S}} r(s, a, s')\, p\big(s' \,\big|\, s, a\big)\; ds' = \mathbb{E}_{s' \sim p(\cdot \,|\, s, a)}\big[r(s, a, s')\big]\,. \tag{2.6}$$

These quantities all feature prominently in the literature and are used as a local gauge of performance.

2.1.1 *Policies*

This generalises the notion of stationary deterministic policies since any function $\pi : \mathcal{S} \to \mathcal{A}$ may be represented as a Dirac distribution.

The *behaviour* of an agent acting in an MDP is characterised by a *policy* $\pi \in \Pi$. These functions can depend explicitly on time or even the entire history of states, actions and rewards. In this thesis we restrict our searches to the class of *stationary (Markovian), stochastic* policies $\Pi_s \subset \Pi$, such that $\pi : \mathcal{S} \to \Delta(\mathcal{A})$ for $\pi \in \Pi_s$, and $\Delta(\mathcal{A})$ denotes the set of probability measures on $\mathcal{A}$; i.e. when $\mathcal{A}$ is absolutely continuous, $\pi(a \,|\, s)$ is a probability density function over actions in a given state s. Policies of this kind do not depend on time (stationarity), and are randomised with respect to their input (stochasticity). It is also well known that, for a wide class of models — including discounted and average reward optimality criteria — there exists at least one deterministic optimal policy which is captured by Π_s [134]. For brevity, we will refer to these "stationary policies" as simply "policies" herein.

It is now possible to define a probability distribution over the space of trajectories by evaluating $\pi(a \,|\, s)$ at each step along a temporal sequence:

$$p\Big(h_{t:(t+n)} \,\Big|\, s_t, \pi\Big) = \prod_{k=0}^{n-1} p(s_{t+k+1} \,|\, s_{t+k}, a_{t+k})\; \pi(a_{t+k} \,|\, s_{t+k})\,. \tag{2.7}$$

Given the distribution of initial states $d_0(\cdot)$, we may also extend (2.7) above to express a distribution over histories as the product

$$p(h_0 \,|\, d_0, \pi) = d_0(s_0)\; p(h_0 \,|\, s_0, \pi)\,. \tag{2.8}$$

The explicit dependence on π is typically omitted from these expressions when it is clear from context, such that $p\Big(h_{t:(t+n)} \,\Big|\, s_t\Big) = p\Big(h_{t:(t+n)} \,\Big|\, s_t, \pi\Big)$. Similarly, we may now extend the reward function definitions, (2.6) and (2.5), to a function $r : \mathcal{S} \to \mathbb{R}$ by marginalising over the paths from one state to the next:

$$\begin{aligned} r_\pi(s) \doteq \int_{\mathcal{A}} r(s, a)\, \pi(a \,|\, s)\; da &= \mathbb{E}_{a \sim \pi(\cdot \,|\, s)}[r(s, a)]\,, \\ &= \mathbb{E}_{s \sim p(\cdot \,|\, s, a), a \sim \pi(\cdot \,|\, s)}\big[r(s, a, s')\big]\,. \end{aligned} \tag{2.9}$$

An important subclass of behaviours that features prominently in this thesis — and indeed the wider literature — is the set of stationary, *parameterised* policies $\Pi_{s,\theta} \doteq \Big\{\pi_\theta \in \Pi_s : \theta \in \mathbb{R}^{|\theta|}\Big\}$. A parameterised policy $\pi_\theta(a \,|\, s)$ is a continuously differentiable function with respect to the weights θ. While $\Pi_{s,\theta}$ does not enjoy the same theoretical performance guarantees as Π_s, there is overwhelming evidence that this is not generally a problem in practice [85]. Indeed the set $\Pi_{s,\theta}$ forms the foundation of the policy gradient methods covered in Section 2.4.2.

Clearly $\Pi_{s,\theta} \subset \Pi_s$ by construction.

2.1.2 *Stationary Distributions*

Take an MDP and a fixed policy, and define the function

$$p(s' \mid s) \doteq \int_{\mathcal{A}} p(s' \mid s, a)\ \pi(a \mid s)\ da. \tag{2.10}$$

This describes the probability of transitioning from a state s to a state s' in a single step, weighted by the likelihood of each path under π. With this definition we have essentially reduced the MDP-policy pair into a *Markov chain* (i.e. an uncontrolled sequence of states with Markovian transition dynamics). This is a crucial insight as it allows us to leverage the notion of *transition kernels* and *stationary distributions* in RL. Specifically, the n-step transition kernel — i.e. the probability of transition from a state s_t to a state s_{t+n} in exactly n steps — is given by the integral equation

$$K_\pi^n(s_t, s_{t+n}) \doteq \int_{\mathcal{A}_t} \int_{\mathcal{S}} \int_{\mathcal{A}_{t+1}} \cdots \int_{\mathcal{S}} \int_{\mathcal{A}_{t+n-1}} p\left(h_{t:(t+n)} \,\middle|\, s_t\right) da_t\, ds_{t+1}\, da_{t+1} \ldots da_{t+n-1}\, ds_{t+n-1}, \tag{2.11}$$

where the action integrals are being evaluated over the subsets $\mathcal{A}_t \doteq \mathcal{A}(s_t)$. Unrolling the trajectory distribution, we find that this expression reduces to an integral over products of 1-step kernels:

$$K_\pi^n(s_t, s_{t+n}) = \int_{\mathcal{S}} \cdots \int_{\mathcal{S}} \prod_{k=0}^{n-1} K_\pi(s_{t+k}, s_{t+k+1})\, ds_{t+1} \ldots ds_{t+n-1}$$

where $K_\pi(s, s') \doteq K_\pi^1(s, s') = p(s' \mid s)$. With some manipulation, this equation can be expressed in a recursive form,

$$K_\pi^{n+1}(s, s'') = \int_{\mathcal{S}} K_\pi^n(s, s') K_\pi(s', s'')\, ds', \tag{2.12}$$

which is a special case of the Chapman-Kolmogorov equations [134].

Equation 2.12 is important because it allows us to analyse, more rigorously, the long-term behaviour of the Markov chain induced by an MDP-policy pair. For any given start state $s_0 \sim d_0(\cdot)$, we are now able to define the probability of arriving at any other state after any number of time steps. As a result, we can also define the *average state distribution*,

$$d_\pi(s) \doteq \lim_{n \to \infty} \frac{1}{n} \int_{\mathcal{S}} d_0(s_0) \sum_{t=0}^{n} K_\pi^t(s_0, s)\, ds_0, \tag{2.13}$$

as the mean occupancy of a state, and, similarly, the *discounted average state distribution*

$$d_\pi^\gamma(s) \doteq \int_{\mathcal{S}} d_0(s_0) \sum_{t=0}^{\infty} \gamma^t K_\pi^t(s_0, s)\, ds_0, \tag{2.14}$$

where $0 < \gamma < 1$ is known as the discount factor. Note that (2.13) and (2.14) are both finite by construction, since $K_\pi^n(s, s')$ outputs probabilities (i.e. values in $[0, 1]$), and either $\gamma < 1$ or the MDP has an absorbing state (by assumption) [134]. These feature prominently in policy gradient literature as a natural way of defining optimisation objectives.

2.1.3 *Performance Criteria*

The goal of any agent in RL is to identify a policy that maximises the expected value of some function of the sequence of rewards generated while following said policy. This value can be expressed as

$$J(\pi) \doteq \mathbb{E}_{d_0,\pi}[g(h)] = \int_{\mathcal{H}} p(h \,|\, d_0, \pi)\; g(h)\, dh \tag{2.15}$$

where $g(\cdot)$ can be any bounded function of trajectories. This is typically given by either a discounted sum or an average over rewards to define a *"reward-to-go" objective*. In this thesis we only concern ourselves with the former, for which we define the n-step *return*, starting from time t, by the summation

Most results in RL can be easily translated between the two regimes.

$$G_{t:(t+n)} \doteq \sum_{k=0}^{n} r_{t+k+1} = r_{t+1} + G_{(t+1):(t+n)}, \tag{2.16}$$

and the *discounted return* by the geometric series

$$G^{\gamma}_{t:(t+n)} \doteq \sum_{k=0}^{n} \gamma^k r_{t+k+1} = r_{t+1} + \gamma G^{\gamma}_{(t+1):(t+n)}, \tag{2.17}$$

where $\gamma \in [0, 1]$ is the *discount rate*, $G^{\gamma}_t \doteq G^{\gamma}_{t:\infty}$ and $G^{\gamma} \doteq G^{\gamma}_0$; equivalent definitions for the undiscounted return are assumed. Note that, when $\gamma = 1$, Equation 2.17 is bounded only if there is a probability 1 of reaching an *absorbing/terminal* state. Such a state emits a reward zero and only transitions back to itself, regardless of the chosen action. The reward-to-go objective in this case now reduces to $J(\pi) = \mathbb{E}_{d_0,\pi}[G^{\gamma}]$, which may also be expressed in terms of the discounted state distribution,

The discount factor may actually be any function of state $\gamma(s)$, subsuming the constant form presented here.

$$J(\pi) = \mathbb{E}_{d^{\gamma}_{\pi},\pi}[r(s)] = \int_{\mathcal{S}} d^{\gamma}_{\pi}(s) \int_{\mathcal{A}(s)} \pi(a \,|\, s)\, r(s, a)\, da\, ds. \tag{2.18}$$

2.1.4 *Value Functions*

Value functions measure the total expected reward that would be accumulated under a policy at any a given state. They are an invaluable set of tools for evaluating the performance in an MDP and date back to the seminal work of Bellman [17], forming the backbone of dynamic programming and many of the techniques used in RL. Indeed, the fields of both prediction (Section 2.3) and control (Section 2.4) revolve around estimating these functions from interactions with an environment. The two main value functions used in RL are the *state-value function*

Taking the limit here stresses the need for convergent sequences if the value functions are to be well defined.

$$\begin{aligned} V^{\gamma}_{\pi}(s) &\doteq \lim_{n\to\infty} \mathbb{E}_{\pi}\left[G^{\gamma}_{t:(t+n)} \,\middle|\, s_t = s \right], \\ &= \lim_{n\to\infty} \mathbb{E}_{\pi}\left[\sum_{k=0}^{n} \gamma^k r(s_{t+k}, a_{t+k}) \,\middle|\, s_t = s \right], \end{aligned} \tag{2.19}$$

and the *action-value function*

$$\begin{aligned} Q^{\gamma}_{\pi}(s, a) &\doteq \lim_{n\to\infty} \mathbb{E}_{\pi}\left[G^{\gamma}_{t:(t+n)} \,\middle|\, s_t = s, a_t = a \right], \\ &= \lim_{n\to\infty} \mathbb{E}_{\pi}\left[\sum_{k=0}^{n} \gamma^k r(s_{t+k}, a_{t+k}) \,\middle|\, s_t = s, a_t = a \right], \end{aligned} \tag{2.20}$$

where the undiscounted value functions are denoted by $V_{\pi} \doteq V^1_{\pi}$ and $Q_{\pi} \doteq Q^1_{\pi}$, respectively. Both V^{γ}_{π} and Q^{γ}_{π} have the interpretation of being scalar potential functions over the spaces $\mathcal{S}$ and $\mathcal{S} \times \mathcal{A}$, respectively.

These are analogous to the potentials arising in such as electromagnetism (i.e. Maxwell's equations).

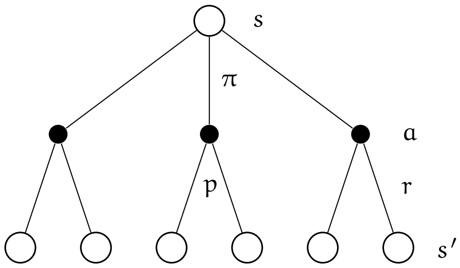


Figure 2.1: Backup diagram of $V_\pi(s)$ [162]. Empty nodes represent states, solid nodes represent actions, and paths correspond to transitions/agent decisions.

BELLMAN EQUATIONS An important property of the state- and action-value functions is that they can be defined recursively. In other words, the quantities V_π^γ and Q_π^γ may be written as functions of themselves at future states. This derives from the recursive nature of the return sequences (2.16) and (2.17), and the linearity of expectation. For example, the latter satisfies a consistency condition of the form:

$$Q_\pi^\gamma(s,a) = r(s,a) + \gamma \int_{\mathcal{S}} p(s' \mid s,a) \int_{\mathcal{A}(s')} \pi(a' \mid s')\; Q_\pi^\gamma(s',a')\,\mathrm{d}a'\,\mathrm{d}s', \tag{2.21}$$

$$= r(s,a) + \gamma \int_{\mathcal{S}} p(s' \mid s,a)\; V_\pi^\gamma(s')\,\mathrm{d}s',$$

$$= r(s,a) + \gamma \mathbb{E}\left[V_\pi^\gamma(s_{t+1}) \mid s_t = s, a_t = a\right].$$

Here we have written the action-value function as a combination of the expected reward (Equation 2.6) and the value at possible successor states, weighted under the policy π and transition kernel. The state-value function has a similar decomposition:

$$V_\pi^\gamma(s) = \int_{\mathcal{A}} \pi(a \mid s) \left(r(s,a) + \gamma \int_{\mathcal{S}} p(s' \mid s,a)\, V_\pi^\gamma(s')\,\mathrm{d}s' \right) \mathrm{d}a, \tag{2.22}$$

$$= \mathbb{E}_\pi[r(s,a_t) + \gamma V_\pi^\gamma(s_{t+1}) \mid s_t = s],$$

$$= r_\pi(s) + \gamma \mathbb{E}_\pi[V_\pi^\gamma(s_{t+1}) \mid s_t = s].$$

Equations 2.22 and 2.21 are known as the *Bellman equations*, and an illustration of the relationship for $V_\pi^\gamma(s)$ is given in Figure 2.1.

ADVANTAGE FUNCTION Another important quantity is the *advantage* function, defined as the difference between the action-value and the state-value,

$$A_\pi^\gamma(s,a) \doteq Q_\pi^\gamma(s,a) - V_\pi^\gamma(s). \tag{2.23}$$

This quantity features prominently in policy gradient methods (Section 2.4.2) due to the fact that it maintains the same ordering over actions,

$$Q_\pi^\gamma(s,a) \geqslant Q_\pi^\gamma(s,a') \iff A_\pi^\gamma(s,a) \geqslant A_\pi^\gamma(s,a') \quad \forall\, s \in \mathcal{S};\, a, a' \in \mathcal{A},$$

This is evident from the fact that (2.23) is a monotonic transformation.

and because the state-dependent offset has been removed, such that the expected value of the advantage function in any state s is zero; i.e.

$$\int_{\mathcal{A}} \pi(a \mid s)\; A_\pi^\gamma(s,a)\,\mathrm{d}a = 0,$$

which follows trivially from (2.21) and (2.22). As we shall see later, this function arises in many value-based control methods (Section 2.4.1), and leads to some favourable properties compared with (2.20) that improves learning efficiency in both policy gradient and actor-critic methods (Section 2.4.3).

2.1.5 *Optimality*

As stated in Section 2.1.3, the goal of any agent is to maximise some expected quantity derived from the sequences of rewards. In the case that we want to maximise the expected value of (2.17), we now see that the objective $J(\pi)$ reduces to

$$J(\pi) = \mathbb{E}_{d_0,\pi}[V^\gamma_\pi(s_0)]\,. \tag{2.24}$$

The importance of value functions should now be clear: they define a total partial ordering over policies, such that

$$\pi \succeq \pi' \iff V^\gamma_\pi(s) \geqslant V^\gamma_{\pi'}(s) \quad \forall s \in \mathcal{S}. \tag{2.25}$$

An *optimal policy* $\pi_\star$ is then defined such that the associated state- and action-value functions are maximal for all $s \in \mathcal{S}$ and $a \in \mathcal{A}(s)$:

$$V^\gamma_\star(s) \doteq \max_\pi V^\gamma_\pi(s) = V^\gamma_{\pi_\star}(s), \tag{2.26}$$

and

$$Q^\gamma_\star(s,a) \doteq \max_\pi Q^\gamma_\pi(s,a) = Q^\gamma_{\pi_\star}(s,a). \tag{2.27}$$

These are known as the *optimal state-value* and *optimal action-value functions*, respectively. For a large class of MDPs, it is well understood that there exists at least one *optimal* stationary policy satisfying these criteria, and that these optimal values are indeed unique; see e.g. Section 6.2 of Puterman [134]. What's more, at least one of these optimal behaviours must be deterministic. This, however, says nothing of policy uniqueness, and indeed there may be more than one policy producing $V^\gamma_\star$ and $Q^\gamma_\star$; i.e. it is a necessary but not sufficient condition. We refer to *any* optimal policy by $\pi_\star$.

BELLMAN OPTIMALITY EQUATIONS As in Section 2.1.4, we require that the optimal value functions be self-consist. For the state-value function this gives rise to a Bellman optimality equation of the form

$$\begin{aligned} V^\gamma_\star(s) &= \max_{a\in\mathcal{A}(s)} Q^\gamma_{\pi_\star}(s,a), \\ &= \max_{a\in\mathcal{A}(s)} \mathbb{E}\left[r(s,a) + \gamma V^\gamma_\star(s_{t+1}) \mid s_t = s, a_t = a\right], \end{aligned} \tag{2.28}$$

and for the action-value function, the condition

$$Q^\gamma_\star(s,a) = \mathbb{E}\left[r(s,a) + \gamma \max_{a'\in\mathcal{A}(s_{t+1})} Q^\gamma_\star(s_{t+1},a') \,\middle|\, s_t = s, a_t = a\right]. \tag{2.29}$$

These derive directly from the definitions (2.26) and (2.27), and the recursive property of value functions. In general, unique solutions to these equations can be shown to exist in both finite and continuous (state/action/time) MDPs under mild technical conditions [118, 134].

OPTIMAL POLICIES It is clear from (2.28) and (2.29) that an optimal policy is one which chooses actions yielding the highest value in every state $s \in \mathcal{S}$. Given the action-value function $Q^\gamma_\star(s,\cdot)$ and a state s, the choice over which action to select reduces to an optimisation problem over $\mathcal{A}(s)$. Note that we need only consider a single step of the MDP, and thus the problem is essentially one of one-step search. A policy that acts according to this metric is said to be *greedy* with respect to the value function $V^\gamma_\star(s)$. For example, a stochastic greedy policy can be defined as

In discrete action spaces this can be implemented via enumeration in linear time. In continuous action spaces it is not quite so simple.

$$\pi_\star(a \mid s) \doteq \delta\left(a \in \mathcal{A}_\star(s)\right) = \begin{cases} \frac{1}{|\mathcal{A}_\star(s)|} & \text{if } a \in \mathcal{A}_\star(s), \\ 0 & \text{otherwise,} \end{cases} \tag{2.30}$$

where $\mathcal{A}_\star(s) \doteq \{a \in \mathcal{A}(s) : Q^\gamma_\star(s, a) = \max_{a' \in \mathcal{A}(s)} Q^\gamma_\star(s, a')\}$ is the subset of optimal actions for each state s, and $|\mathcal{A}_\star(s)|$ denotes the size of this set. The deterministic variant of (2.30) would simply choose between one of the actions in $\mathcal{A}(s)$ using some fixed decision rule, leaving probability density on a single value. More generally, any policy that places its probability mass only on elements of $\mathcal{A}_\star(s)$ is optimal.

Clearly if there is a state s for which $|\mathcal{A}_\star(s)| > 1$, there exist an infinite number of optimal policies.

2.2 FUNCTION APPROXIMATION

In MDPs with discrete state- and action-spaces, the value functions V^γ_π and Q^γ_π may be represented trivially as matrices; i.e. for each state (respectively, state-action pair), we simply assign a unique real scalar. While this setting comes with its own challenges — revolving mainly around inefficiency and scaling (the curse of dimensionality) — the use of a lookup table allows for exact recovery of the true value functions. For continuous domains, however, we must resort to approximate methods since there is no feasible way of representing the entire space of value functions without reconstruction error. In general, any approximation will likely lead to a phenomenon known as "perceptual aliasing" [43] in which indistinguishable states with respect to the approximation require different actions. This pathology arises often when there are discontinuities in the state-space such as walls or other imposed constraints.

In this thesis we restrict ourselves to the space of *linear (in-the-weights) functions*. These are expressed as the inner product between a finite set of weights and a collection of basis functions, such as $\phi_i : \mathcal{S} \to \mathbb{R}$ or $\phi_i : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$. In this setting, approximators for the state-value function are expressed as

$$\widehat{V}_{\boldsymbol{v}}(s) \doteq \langle \boldsymbol{v}, \boldsymbol{\phi}(s) \rangle = \sum_{i=1}^{m} v_i \phi_i(s), \tag{2.31}$$

and for the action-value function by

$$\widehat{Q}_{\boldsymbol{w}}(s, a) \doteq \langle \boldsymbol{w}, \boldsymbol{\phi}(s, a) \rangle = \sum_{i=1}^{m} w_i \phi_i(s, a), \tag{2.32}$$

where $\boldsymbol{\phi}(s)$ and $\boldsymbol{\phi}(s, a)$ map inputs into m-dimensional column vectors, and $\boldsymbol{v}, \boldsymbol{w} \in \mathbb{R}^m$ are m-dimensional row vectors; note the circumflex is used more generally to mark function approximators. This choice leads to simple learning algorithms and amendable error surfaces, despite the fact that the basis functions themselves may be arbitrarily complex. Exactly how one constructs these basis functions for a given problem, however, is a non-trivial question that depends on the properties of the problem domain.

Indeed, this is the main argument in favour of models that automatically extract features, such as deep neural networks.

In the following sub-sections, we present some of the standard techniques used throughout the literature, and those used in the research presented herein. We will focus only on functions of the form in Equation 2.31, and assume that the state space is defined on the n-dimensional set of real values; i.e. $\mathcal{S} = \mathbb{R}^n$. A state $\mathbf{s} \in \mathcal{S}$ will thus be denoted by the vector $\mathbf{s} = [s_1, \ldots, s_n]$. The last sub-section will then cover methods for representing action-value functions when $\mathcal{A}$ is discrete/ordinal.

2.2.1 *Local Representations*

PARTITIONING Perhaps the simplest way of representing functions in a continuous space is to use a piecewise-constant approximation. In this case, the domain is partitioned into a finite set of m disjoint boxes. The i^{th} feature in the basis is then associated with a single box, b_i, such that $\phi_i(s) = \mathbf{1}_{s \in b_i}$. The key advantage of this approach is that we can increase the resolution as much as required with no overhead in computational cost of evaluation — we just perform a table lookup.

Essentially we have constructed a finite, partially-observable MDP to serve proxy.

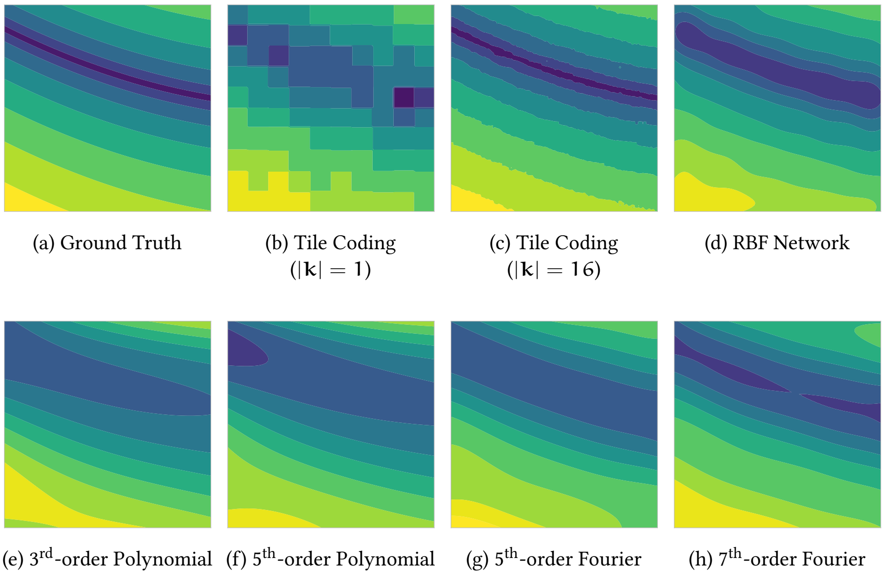


Figure 2.2: Collection of function basis representations approximating the 6^{th} Bukin function, $f(x, y) = 100\sqrt{|y - 0.01x^2|} + 0.01|x + 10|$, learnt using stochastic gradient descent.

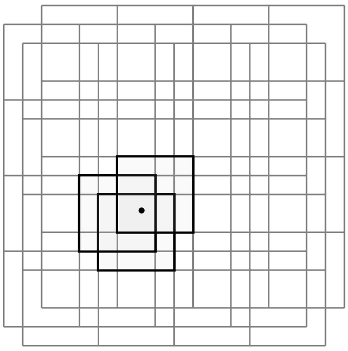


Figure 2.3: An example illustration of the tile coding representation. Three tilings are shown, for which the activated tiles are highlighted around the point state instance.

As $m \to \infty$, we will able to recover the function exactly; this is just integration. The memory required to uniformly cover the space, however, scales exponentially with $|\mathcal{S}|$, and the amount of experience required to estimate the value of each bin independently — i.e. without generalising — quickly becomes intractable.

TILE CODING A better approach for sparse representations, which facilitates generalisation with reduced memory requirements, is tile coding (a form of coarse coding). In this scheme we construct $|\mathbf{k}|$ "tilings," each of which is a unique partitioning over the state space with k_i boxes for a total of $m = \sum_{i=1}^{|\mathbf{k}|} k_i$ features and the ability to share information across receptive fields; see Figure 2.3. This results in a "smoothing" of the boundaries between tiles and, in principle, improved estimates at any one point in the space. As with simple partitioning, the computational cost is very low — linear in $|\mathbf{k}|$ — but now the memory requirements can be controlled by either refinement (exponential), or by the addition of more tilings (linear).

RBF NETWORKS Radial basis function networks are one extension of the basic partitioning representation to continuous features. In this case, the basis is constructed from a collection of prototype Normal distributions,

$$\phi_i(\mathbf{s}) = \frac{1}{\sqrt{(2\pi)^k \det \boldsymbol{\Sigma}}} \exp\left\{-\frac{1}{2}\left\langle (\mathbf{s}-\mathbf{c}_i), \boldsymbol{\Sigma}^{-1}(\mathbf{s}-\mathbf{c}_i)\right\rangle\right\}, \tag{2.33}$$

where π here should be taken as the mathematical constant. Here, the values $\mathbf{c}_i$ are the centres of each node in the state space, and $\boldsymbol{\Sigma}$ denotes the covariance matrix between dimensions of $\mathcal{S}$. In the majority of cases, $\boldsymbol{\Sigma}$ is taken to be a diagonal matrix which greatly simplifies computations and improves stability during gradient updates. As illustrated in Figure 2.2, the continuity implied by the distance metric above leads to a smooth approximation of the value function. This compares to the previous examples which are more discontinuous.

2.2.2 *Global Representations*

POLYNOMIAL BASIS An important and frequently used linear representation — first pioneered by Lagoudakis and Parr [96] — is the polynomial basis. In this scheme, an n-dimensional state $\mathbf{s}$ is projected onto an m-dimensional feature-space using basis functions of the form

$$\phi_i(\mathbf{s}) = \prod_{j=1}^{n} s_j^{e_{ij}}, \tag{2.34}$$

where e_{ij} denotes the exponent of the j^{th} state-variable along the i^{th} feature dimension; the zeroth term is usually taken to be a constant such that $e_{0j} = 0 \; \forall \, j \in \{1, \ldots, n\}$. The structure of the matrix e_{ij} will have a significant bearing on the relationships between state-variables that can be represented. There is once again a trade-off between representational capacity, the risk of over-fitting, and the curse of dimensionality. All of these can be addressed through careful construction of e_{ij}, such as reducing the number of cross-terms and using domain knowledge to limit complexity.

Implementations should ensure that e_{ij} is a triangular matrix.

FOURIER BASIS An important limitation of the polynomial basis is that higher order terms can be very unstable when the state-variables are not properly scaled. An alternative approach is to use the cosine terms of a Fourier series expansion; i.e. represent the value function as a sum of periodic functions over the interval $[0, 1]^n$. This choice is motivated in part by the fact that any periodic function (or arbitrary function defined on the period interval) can be represented this way. The i^{th} feature of the Fourier cosine basis takes the form

Fourier transformations arises frequently in e.g. the physical sciences and in signal processing.

$$\phi_i(\mathbf{s}) = \cos(\pi \langle \mathbf{c}_i, \mathbf{s} \rangle), \tag{2.35}$$

where $\mathbf{c}_i \doteq (c_{11}, \ldots, c_{1n})$ is the integer coefficient vector, with $c_{ij} \in \{0, \ldots, k\}$ and $0 \leqslant j \leqslant n$; note that π should be taken as the mathematical constant here. As shown by Konidaris, Osentoski, and Thomas [95], this basis often outperforms both the polynomial and RBF bases, and even performs competitively on problems with discontinuities. While it is known to struggle at representing flat landscapes (due to the Gibbs phenomenon [65]), experimental evidence suggests that this an effective method for use in RL.

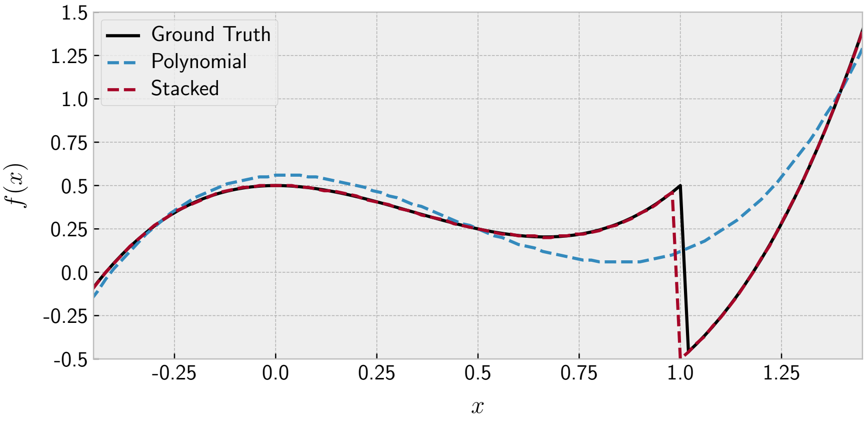


Figure 2.4: Illustration of two least squares approximations of a discontinuous polynomial function. The stacked basis combines the polynomial basis with a standard, uniform partitioning.

2.2.3 *Extensions*

2.2.3.1 *Stacking*

It is worth noting that there are no restrictions on how these representations are used or combined. As we have seen, each basis comes with it's own unique set of merits and limitations. In some cases, these may be mutually beneficial. For example, one could use a linear basis in tandem with tile coding to combine the benefits of generalisation that come from global estimation with the precision of localised features; see Figure 2.4 for an illustration of this point. While this all depends on the particular problem being solved, the construction of the feature set is tantamount to the construction of the solution space.

2.2.3.2 *Anchoring*

In the case of a finite horizon MDP, with horizon $0 < T < \infty$, we know that the value at any terminal state is necessarily zero. In many cases, this manifests in the value function as a discontinuity at the boundary between $t < T$ and $t = T$. This can lead to inaccuracy at the boundaries of the estimator. One can improve the stability of function approximation by explicitly constraining the terminal value to be zero. This means that the basis, $\boldsymbol{\phi}(s)$, need not handle the discontinuity itself which greatly improves performance, especially for global representations. As an example, one might define the approximator as

$$\hat{V}_{\boldsymbol{v}}(s) \doteq \begin{cases} \langle \boldsymbol{v}, \boldsymbol{\phi}(s) \rangle & \text{for } t < T, \\ 0 & \text{otherwise.} \end{cases}$$

We shall use this construction implicitly throughout the thesis, unless otherwise stated.

2.2.3.3 *Handling actions*

So far we have shown that there many ways to construct **LFA!**s (**LFA!**s) when the arguments to the basis functions are real vectors. However, it's not immediately clear how these translate to action-value representations when the action space is discrete or ordinal. For example, it would be highly unnatural — and ill-defined —

to map actions of the form $a \in \{\texttt{OpenDoor}, \texttt{MoveToTarget}, \dots\}$ directly through a polynomial basis. This is only valid, in any sense, when the actions themselves are real vectors. In general, this issue is handled by assigning a set of basis functions to each of the individual actions such that

$$\boldsymbol{\phi}(s, a) = \begin{bmatrix} | & & | \\ \mathbf{1}_{a_1} \odot \boldsymbol{\phi}(s, a_1) & \cdots & \mathbf{1}_{a_n} \odot \boldsymbol{\phi}(s, a_n) \\ | & & | \end{bmatrix}, \tag{2.36}$$

where $n = |\mathcal{A}|$ and $\odot$ denotes the Hadamard product. The value function would then be represented as the Hadamard product of an $(m \times n)$-dimensional feature matrix and an $(m \times n)$-dimensional weight matrix, and summing the values in each column. The resulting n-dimensional vector would contains the Q-values for each action. For improved learning efficiency, one can also include an action-independent baseline term. This facilitates sharing of mutual information between actions.

We can implement this efficiently by exploiting sparsity.

2.3 POLICY EVALUATION

See the excellent survey by Dann, Neumann, Peters, et al. [48] for a thorough exposition of policy evaluation methods.

Estimating value functions from (partial) interactions with an environment underpins many of the algorithms used in RL. Indeed, the ability to predict quantities associated with a task is helpful in many domains in and of itself. As we saw in Section 2.1.4, the two most common cases are that of the state-value function V_π^γ and action-value function Q_π^γ, but these are by no means exhaustive. Value functions can be used to estimate risk [150, 159], predict the time before an event occurs, or answer "what-if" questions about a domain [165].

Consider the problem of learning to estimate the state-value function V_π^γ (Equation 2.19) from interactions generated by a policy π. In this thesis, we assume the *model-free* setting in which the agent only has access to a black-box simulator of the environment; when the model is known to the agent, the setting is instead known as model-based RL. Our objective is to find an approximator, $\hat{V}_{\boldsymbol{v}}$, that minimises the distance from the true function, which we express in terms of the *mean squared error*,

This is known as the on-policy prediction problem.

$$\mathrm{MSE}(\boldsymbol{v}) \doteq \left\| \hat{V}_{\boldsymbol{v}} - V_\pi^\gamma \right\|_\mu^2 = \int_{\mathcal{S}} \mu(s) \left[\hat{V}(s) - V_\pi^\gamma(s) \right]^2. \tag{2.37}$$

The function $\mu : \mathcal{S} \to \mathbb{R}$ denotes any stationary distribution over states, which is usually taken to be d_π^γ. This is just an instance of a weighted least-squares regression problem and, if the approximator is continuously differentiable with respect to it's weights, $\boldsymbol{v}$, then the gradient of (2.37) is given by

$$\begin{aligned} 2\nabla \mathrm{MSE}(\boldsymbol{v}) &= \int_{\mathcal{S}} \mu(s) \left[\hat{V}(s) - V_\pi^\gamma(s) \right] \nabla_{\boldsymbol{v}} \hat{V}_{\boldsymbol{v}}(s), \\ &= \mathbb{E}_{s \sim \mu(\cdot)} \left[\left[\hat{V}(s) - V_\pi^\gamma(s) \right] \nabla_{\boldsymbol{v}} \hat{V}_{\boldsymbol{v}}(s) \right], \end{aligned} \tag{2.38}$$

which follows from Leibniz's integral rule. Evaluating this integral, however, is usually intractable since $\mathcal{S}$ may be very large. Instead, we exploit the fact this this expression takes the form of an expectation with respect to the distribution μ, and use stochastic gradient descent to update the weights incrementally in the direction of steepest descent in the mean-squared error. This leads to a stochastic sequence of weights

$$\boldsymbol{v}_{t+1} \leftarrow \boldsymbol{v}_t - \alpha_t \left[\hat{V}_{\boldsymbol{v}}(s_t) - V_\pi^\gamma(s_t) \right] \nabla_{\boldsymbol{v}} \hat{V}_{\boldsymbol{v}}(s_t), \tag{2.39}$$

where the factor $1/2$ has been subsumed into the *learning rates* α_t.

Iterative methods of this form belong to a class of *stochastic approximation algorithms* that can be solved using the Robbins-Monro procedure [138]. As a result, we know that the true value function in Equation 2.39 can actually be replaced by any target, $U_t \in \mathbb{R}$, that is an unbiased estimator of V_π^γ without affecting the fixed-point. These stochastic updates are guaranteed to converge to a local optimum under mild technical conditions, including that the learning rates $\alpha_t \geqslant 0$ satisfy

$$\sum_{t=0}^{\infty} \alpha_t = \infty, \quad \text{and} \quad \sum_{t=0}^{\infty} \alpha_t^2 < \infty; \tag{2.40}$$

these are known as the Robbins-Monro conditions. The proof of this result was shown by Borkar [25] using a beautiful technique, now known as the ODE method. From similar arguments, concentration with high probability in a neighbourhood of a fixed point can also be shown for fixed learning rates, but the this result is clearly much weaker [25]. Perhaps the most natural choice for the target is the discounted return sequence generated by interacting with the environment, since $U_t \doteq G_t^\gamma$ is an unbiased estimator of the true value function by definition (2.19). Algorithms using this target are known as *Monte-Carlo methods* and they suffer from two key limitations: first, they require full trajectory rollouts to generate estimates; and second, they have variance that scales with the length of the trajectories,

$$\mathbb{V}\left[G_t^\gamma\right] = \sum_{i=0}^{\infty} \gamma^{2i} \mathbb{V}\left[r_{t+i}\right] + \sum_{i \neq j} \gamma^{i+j} \operatorname{Cov}\left[r_{t+i}, r_{t+j}\right].$$

This presents a dichotomy: while many rollouts are needed to reduce uncertainty, collecting full return sequences takes an indefinite length of time. The remainder of this section is dedicated to other choices for U_t that are known to alleviate precisely these issues.

2.3.1 *Temporal-Difference Methods*

Monte-Carlo methods, as outlined above, are something of a brute-force approach to generating estimates of V_π^γ. Because they make no use of consolidated information, they require complete policy trajectories in order to construct a single estimate. While this leads to unbiased updates, the increased variance of the estimator often renders the approach impractical. On the other hand, if we were to use our estimate of the value function in place of the target — i.e. let $U_t \doteq \widehat{V}_w(s_t)$ — we would be in the opposite regime: one of total bias in exchange for zero variance.

The insight of temporal-difference (TD) methods is to note that there exists a spectrum of algorithms at the intersection of these two extremes that balance bias and variance. In general, this use of consolidated information in forming an target for the stochastic approximation scheme is known as *bootstrapping*. The idea derives from dynamic programming principles and the recursive definition of the value function (Equation 2.22) which may be unrolled to form a family of n-step bootstrap estimates, $U_t \doteq \sum_{i=1}^{n} \gamma^{i-1} r_{t+i} + \gamma^n \widehat{V}_v(s_{t+n})$. Clearly, as n decreases, so too does the variance at the expense of increasing the bias; note that the Monte-Carlo target is recovered in the limit as $n \to \infty$.

Technically speaking, the n-step bootstrap targets do not yield true gradient descent updates as we have retrospectively ignored the dependence on v in Equation 2.38. Instead, they fall under the category of *semi-gradient methods* which do not converge as robustly — save for a few special cases such as with linear architectures [175] — but have the benefit of fast learning and the ability to do online updates. One prototypical algorithm, known as TD(0), makes use of the 1-step bootstrap target, for which we define the canonical *Bellman error*,

$$\delta_t^\gamma \doteq r_{t+1} + \gamma \widehat{V}_v(s_{t+1}) - \widehat{V}_v(s_t); \tag{2.41}$$

Baird [13] was the first to note this, proposing an alternative "residual-gradient" algorithm, but this comes with it's own challenges.

Algorithm 1 Episodic semi-gradient TD(0)

```
procedure TrainEpisode($d_0$, $\pi$, $p$, $\widehat{V}_{\boldsymbol{v}}$, $\alpha$, $\gamma$)
    Initialise $s \sim d_0(\cdot)$
    while $s$ non-terminal do
        Sample $a \sim \pi(\cdot \mid s)$ and $(r, s') \sim p(\cdot, \cdot \mid s, a)$
        Compute $\delta^\gamma \leftarrow r + \gamma \widehat{V}_{\boldsymbol{v}}(s') - \widehat{V}(s)$
        Update $\boldsymbol{v} \leftarrow \boldsymbol{v} + \alpha \delta^\gamma \nabla_{\boldsymbol{v}} \widehat{V}_{\boldsymbol{v}}(s)$
        Innovate $s \leftarrow s'$
    end while
end procedure
```

Algorithm 2 Episodic semi-gradient SARSA(0)

```
Require: initial state distribution $d_0$, policy $\pi$, transition kernel $p$ and differentiable
    function $\widehat{Q}_{\boldsymbol{w}}$ with learning rate $\alpha$
Initialise $s \sim d_0(\cdot)$
loop
    Sample $a \sim \pi(\cdot \mid s)$ and $(r, s') \sim p(\cdot, \cdot \mid s, a)$
    if $s'$ non-terminal then
        Sample $a' \sim \pi(\cdot \mid s')$
        Compute $\delta^\gamma \leftarrow r + \gamma \widehat{Q}(s', a') - \widehat{Q}(s, a)$
        Update $\boldsymbol{w} \leftarrow \boldsymbol{w} + \alpha \delta^\gamma \nabla \widehat{Q}(s, a)$
        Innovate $(s, a) \leftarrow (s', a')$
    else
        Compute $\delta^\gamma \leftarrow r - \widehat{Q}(s, a)$
        Update $\boldsymbol{w} \leftarrow \boldsymbol{w} + \alpha \delta^\gamma \nabla \widehat{Q}(s, a)$
        Initialise $s \sim d_0(\cdot)$
    end if
end loop
```

the corresponding pseudocode is given in Algorithm 1. Of course, this expression generalises to any n, but the choice of what lookahead to use depends entirely on the problem setting. In Section 2.3.2 we show how this can be done in a principled way through the use of eligibility traces.

ACTION-VALUE ESTIMATION The extension of the policy evaluation methods described thus far to action-value functions is trivial. The only difference is that the prediction targets are now associated with a state-action pair, not just a state. As before, we can use n-step returns as a target for incremental learning, and, as before, we are faced with the same trade-off between bias and variance. It follows that we can define an analogous 1-step Bellman error,

$$\delta_t^\gamma \doteq r_{t+1} + \gamma \widehat{Q}_{\boldsymbol{w}}(s_{t+1}, a_{t+1}) - \widehat{Q}_{\boldsymbol{w}}(s_t, a_t), \tag{2.42}$$

for which the corresponding algorithm is known as SARSA(0); see Algorithm 2.

EXPECTED SARSA In certain special cases we can actually reduce the variance on the Bellman residual even further by replacing the second term in (2.42) to give

$$\begin{aligned} \delta_t^\gamma &\doteq r_{t+1} + \gamma \mathbb{E}_{a' \sim \pi_\theta(\cdot \mid s_t)}\left[\widehat{Q}_{\boldsymbol{w}}(s_{t+1}, a')\right] - \widehat{Q}_{\boldsymbol{w}}(s_t, a_t), \\ &= r_{t+1} + \gamma V_\pi^\gamma(s') - \widehat{Q}_{\boldsymbol{w}}(s_t, a_t). \end{aligned} \tag{2.43}$$

When $\mathcal{A}$ is discrete, and we are able to effectively enumerate the action-values, then this reduces to a summation over the values in $\mathcal{A}$, such that

$$\delta_t^\gamma = r_{t+1} + \gamma \sum_{a' \in \mathcal{A}} \pi_\theta(a' \mid s_t)\, \widehat{Q}_{\boldsymbol{w}}(s_{t+1}, a') - \widehat{Q}_{\boldsymbol{w}}(s_t, a_t).$$

As shown by Van Seijen, Van Hasselt, Whiteson, and Wiering [177], the resulting algorithm, known as Expected SARSA, is more stable than SARSA because it marginalises out the randomness due to the sampling of $a_{t+1} \sim \pi_\theta(\cdot \mid s_{t+1})$. The key point to take from this is that the target, U_t, can be engineered to have the properties we require for a particular problem. In the context of trading, variance reduction will turn out to be particularly important, as we show in Chapter 5.

2.3.2 *Eligibility Traces*

So far, only three special cases of the n-step bootstrap targets have been considered: $n \in \{0, 1, \infty\}$. Yet, there exists an infinite number of possibilities for finite values of $n > 1$. Indeed, any convex combination of these targets yields another viable choice; e.g. $U_t = \frac{1}{2} G_{t:t+2}^\gamma + \frac{1}{2} G_{t:t+5}^\gamma + \widehat{V}^\gamma(s_t)$. All of these trade-off bias against variance, but the “best” construction to use depends entirely on the problem at hand, and it is unclear how one establishes this a priori. An alternative approach is to mix between *all* of the targets simultaneously using what is known as the λ-*return*. It is well understood that this leads to much more efficient learning, and offers a principled way of interpolating between Monte-Carlo and temporal-difference methods.

The λ-*return target* is defined formally as the geometric series over all n-step returns and tail values,

$$U_{t:t+n}^{\gamma\lambda} \doteq (1-\lambda) \left[\sum_{i=1}^{n-1} \lambda^{i-1} G_{t:t+i}^\gamma + \gamma^i V_\pi^\gamma(s_{t+i}) \right] + \lambda^{n-1} \left[G_{t:t+n}^\gamma + \gamma^n V_\pi^\gamma(s_{t+n}) \right], \tag{2.44}$$

and $U_t^{\gamma\lambda} \doteq U_{t:\infty}^{\gamma\lambda}$, where λ is the decay rate. For $\lambda \in [0, 1]$, we can show that the expected value of (2.44) is again that of the true value function, and thus $U_t^{\gamma\lambda}$ is an unbiased target for stochastic approximation. Combining this with the insights of Section 2.3.1, we can construct a TD error based on this interpolated target, $\delta_t^{\gamma\lambda} \doteq U_t^{\gamma\lambda} - \widehat{V}_{\boldsymbol{v}}(s_t)$, where $V_\pi^\gamma(s)$ in (2.44) is replaced with the biased estimator $\widehat{V}_{\boldsymbol{v}}(s)$. This can now be used in lieu of (2.41) or (2.42) as the error term of our stochastic approximation. As before, this modified error is biased, but the trade-off between bias and variance can now be controlled through the choice of decay rate, λ. For $\lambda = 0$, the returns degenerate to the pure bootstrapping regime ($n = 1$), and for $\lambda = 1$, the Monte-Carlo target is recovered ($n = \infty$).

This formulation is known as the *forward view* because it defines the update in terms of future observations. This, of course, cannot be done in practice since there is no way of knowing in advance what states and rewards will follow, or how long an single episode will last. Fortunately, it is understood that there is an equivalent *backward view* in which the TD-error is projected back to past states, retrospectively. This is achieved through the use eligibility traces, which, as the name suggests, are a technique for assigning credit to states that may have influenced the latest observation in a temporal sequence. They have the effect of propagating information back up the Markov chain induced by the policy. When used alongside differentiable function approximation, the trace may be expressed as the difference relation

$$\begin{aligned} \boldsymbol{e}_0 &\doteq \nabla_{\boldsymbol{v}} \widehat{V}_{\boldsymbol{v}}(s_0), \\ \boldsymbol{e}_t &\doteq \gamma\lambda \boldsymbol{e}_{t-1} + \nabla_{\boldsymbol{v}} \widehat{V}_{\boldsymbol{v}}(s_t), \end{aligned} \tag{2.45}$$

Algorithm 3 Episodic semi-gradient $\text{TD}(\lambda)$

```
procedure TrainEpisode($d_0, \pi, p, \widehat{V}_{\boldsymbol{w}}, \alpha, \gamma, \lambda$)
    Initialise $s \sim d_0(\cdot)$ and $\boldsymbol{e} \leftarrow \mathbf{0}$        ▷ $|\boldsymbol{w}|$-dimensional vector
    while $s$ non-terminal do
        Sample $a \sim \pi(\cdot \mid s)$ and $(r, s') \sim p(\cdot, \cdot \mid s, a)$
        Update $\boldsymbol{e} \leftarrow \gamma\lambda\boldsymbol{e} + \nabla\widehat{V}(s)$
        Compute $\delta^{\gamma\lambda} \leftarrow r + \gamma\widehat{V}(s') - \widehat{V}(s)$
        Update $\boldsymbol{w} \leftarrow \boldsymbol{w} + \alpha\delta^{\gamma\lambda}\boldsymbol{e}$
        Innovate $s \leftarrow s'$
    end while
end procedure
```

where Equation 2.39 then reduces to $\boldsymbol{v}_{t+1} \leftarrow \boldsymbol{v}_t + \alpha_t\delta_t^{\gamma\lambda}\boldsymbol{e}_t$. All of these operations scale linearly with $|\boldsymbol{e}_t|$, meaning it can be integrated into the stochastic approximation framework in a computationally efficient manner. The canonical example of this, $\text{TD}(\lambda)$, is outlined in Algorithm 3 and is a direct extension of $\text{TD}(0)$. An analogous extension to action-value functions yields $\text{SARSA}(\lambda)$.

2.3.3 *Least-Squares Methods*

While temporal-difference (TD) methods have demonstrable benefits over Monte-Carlo policy evaluation, they can still suffer from instability due to the combination of function approximation, bootstrapping and reliance on stochastic gradient descent. This is especially true when complex function approximators are used, such as deep neural networks. An important class of algorithms, first identified by Bradtke and Barto [28], improves upon this by casting the prediction problem as a classical linear least-squares regression problem. These benefit from significantly greater sample efficiency and improved stability and can even be combined with eligibility traces.

These pathologies are known components of Sutton's "deadly triad" [162].

To achieve this, Bradtke and Barto observed that the sum of TD-updates sampled during learning can be factored into normal equations typically seen in regression analysis. Denoting $\zeta_T \doteq \sum_{t=1}^{T} U_t$ for a fixed approximator, $\widehat{V}_{\boldsymbol{v}}$, we can expand the summation to give:

$$\zeta_T = \underbrace{\sum_{t=1}^{T} \boldsymbol{\phi}_t r_{t+1}}_{\boldsymbol{b}_T} - \underbrace{\sum_{t=1}^{T} \boldsymbol{\phi}_t \left(\boldsymbol{\phi}_t - \gamma\boldsymbol{\phi}_{t+1}\right)^\top}_{\boldsymbol{A}_T} \boldsymbol{v},$$

where $\boldsymbol{\phi}_t = \boldsymbol{\phi}(s_{t+1})$ and we have used the standard 1-step bootstrap target (see Equation 2.41). It follows from standard arguments that the optimal choice of parameters, $\boldsymbol{v}$, at T according the mean-squared error is given by the solution to the system of simultaneous equations

$$\boldsymbol{v}_{T+1} = \boldsymbol{A}_T^{-1}\boldsymbol{b}_T.$$

This insight yields an immediate algorithm for policy evaluation in which the solution is recomputed at prescribed intervals. The challenge, however, is that computing $\boldsymbol{A}_T^{-1}$ requires $O(n^3)$ operations, where n is the number of features. A whole host of algorithms have since spawned from this key insight [26, 27], including recursive [185] (RLSTD) and incremental [63] (iLSTD) implementations designed to reduce the complexity of the matrix inversion operation using clever approximations. We leave further details of this topic to the reader, but note that the iLSTD algorithm is used in Chapter 6.

Algorithm 4 Episodic Q-learning

```
procedure TrainEpisode($d_0$, $\pi$, $p$, $\widehat{Q}_{\boldsymbol{w}}$, $\alpha$, $\gamma$)
    Initialise $s \sim d_0(\cdot)$
    while $s$ non-terminal do
        Sample $a \sim \pi(\cdot \mid s)$ and $(r, s') \sim p(\cdot, \cdot \mid s, a)$
        Compute $\delta^\gamma \leftarrow r + \gamma \max_{a'} \widehat{Q}_{\boldsymbol{w}}(s', a') - \widehat{Q}_{\boldsymbol{w}}(s, a)$
        Update $\boldsymbol{w} \leftarrow \boldsymbol{w} + \alpha \delta^\gamma \nabla_{\boldsymbol{w}} \widehat{Q}_{\boldsymbol{w}}(s, a)$
        Innovate $s \leftarrow s'$
    end while
end procedure
```

2.4 POLICY OPTIMISATION

The second key area of RL is concerned with learning a policy that maximises a chosen objective, $J(\pi)$, such as the discounted sum of future rewards (Equation 2.24). As in Section 2.3, we assume the model-free setting where the agent interacts with a black-box simulator and has no explicit knowledge of the underlying dynamics of the environment. Such problems are known as *learning problems*, and the focus is on algorithmic complexity rather than, say, online performance; i.e. the compute and memory required to find a solution.

One of the key concepts in policy optimisation — also known as *control* — is the exploration-exploitation dilemma. This refers to the problem of balancing the cost of trying something new (exploration) with the benefit of acting greedily with respect to the current knowledge of the problem (exploitation). The former is crucial in helping the agent break out of local optima and has been the subject of much research in and of itself [1, 41, 89]. In the following sections we will cover some of the important classes of methods that will be used in this thesis, each of which involves different approaches to exploration and exploitation.

2.4.1 *Value-Based Methods*

Greedy/value-based methods work directly in the space of value functions by solving for the optimal action-value function, $Q_\star^\gamma$. In the approximate setting that we study here, the algorithms can be seen as sample-based equivalents of value iteration in dynamic programming [17]. In general, the process revolves around generating a sequence of action-value estimates $\{\widehat{Q}_{\boldsymbol{w}_t}\}_{t \geqslant 0}$ which converge to $Q_\star^\gamma$ in the limit. This follows from the Bellman optimality equation in (2.29) and acting greedily with respect to this function guarantees that the implied policy is optimal (Section 2.1.5).

The most notable algorithm, which exploits the Bellman optimality equation to find the optimal policy, is known as Watkins' Q-learning [181]; see Algorithm 4. This method works similarly to SARSA, only the TD target (2.42) is replaced with

$$\delta_t^\gamma \doteq r_t + \gamma \max_{a' \in \mathcal{A}} \widehat{Q}_{\boldsymbol{w}}(s', a') - \widehat{Q}_{\boldsymbol{w}}(s, a). \tag{2.46}$$

The alternative is to use an inner optimisation routine to compute the maximum over actions as in [142].

Due to the maximisation step, this algorithm is almost exclusively used in MDPs where the action-space is discrete and enumerable; and where $|\mathcal{A}|$ is not too large. Despite this limitation, Q-learning is an exceptionally powerful technique that laid the foundations for much of RL today. It has enjoyed swathes of experimental success due to its simplicity and effective performance, and is even known to converge with probability 1 to a neighbourhood around the optimal solution even when combined with function approximation; or the exact solution in tabular settings. These theoretical results are, however, somewhat weak and require a number of strict assumptions on the problem itself and/or the function approximator used [106]; see e.g. [9, 111, 168]. Nevertheless, as we will show in Part II, `Q-learning`, for suitably chosen function approximators, can produce good results on the financial problems we study in this thesis.

STABILITY AND OFF-POLICY LEARNING The lack of strong theoretical guarantees for convergence with `Q-learning` with linear function approximation was later addressed by Sutton, Maei, Precup, Bhatnagar, Silver, Szepesvári, and Wiewiora [163] and Maei and Sutton [105]. These methods replace the traditional mean-squared Bellman error objective with one that takes into account the basis of the approximation space. The result is a family of "full-gradient" algorithms which have greatly improved stability in the off-policy setting. This, however, comes at the cost of slower learning compared with the classical semi-gradient methods such as TD, SARSA and `Q-learning`.

MAXIMISATION BIAS Another key issue with traditional `Q-learning` arises from the maximisation bias due to the term $\max_{a \in \mathcal{A}} \widehat{Q}_{\boldsymbol{w}}(s, a)$ in Equation 2.46. This pathology can lead to very poor performance in domains with highly stochastic transition dynamics and/or rewards. Specifically, overestimates in the action-value approximator lead to greedy policies with a positive bias towards actions whose rewards are highly dispersed. The canonical example used to illustrate this is Blackjack, where a lucky chance early on in training can lead to policies that repeatedly bet on highly improbable outcomes. This problem was studied by Hasselt [82] who proposed a solution in the form of a double estimator. The algorithm, known as `Double Q-learning` has been shown to solve this problem in exchange for a small amount of underestimation. This approach is very simple to implement and remains linear in update complexity, and will feature later in Chapter 5.

2.4.2 *Policy Gradient Methods*

At the other end of the spectrum, policy gradient methods work in the space of *explicit*, parameterised policies, $\Pi_{s,\theta}$. Rather than learn a value function and act greedily with respect to the latest estimate of $Q_\star^\gamma$, this family of approaches updates the parameters of a policy directly. The idea is to move θ in the direction of steepest ascent with respect to the chosen objective $J(\theta)$. This hinges on a key result in RL known as the policy gradient theorem which expresses this derivative in terms of the score of the policy and the policy's action-value function.

Theorem 1 (Policy gradient)**.** *Let $J(\theta)$ be the objective function defined in Equation 2.24. The gradient of this quantity with respect to the policy parameters, θ, is given by*

$$\nabla_\theta J(\theta) = \int_{\mathcal{S}} d_\pi^\gamma(s) \int_{\mathcal{A}(s)} Q_\pi^\gamma(s, a) \, \nabla_\theta \pi_\theta(a \mid s) \, \mathrm{d}a \, \mathrm{d}s, \tag{2.47}$$

where the stationary distribution, $d_{\pi_\theta}^\gamma(s)$ is as defined in Equation 2.14.

Proof. See Sutton, McAllester, Singh, and Mansour [164]. ■

Using the log-likelihood trick [183], the policy gradient above can be rewritten as an expectation,

$$\begin{aligned}\nabla_\theta J(\theta) &= \int_{\mathcal{S}} d_\pi^\gamma(s) \int_{\mathcal{A}(s)} Q_\pi^\gamma(s, a) \, \pi_\theta(a \mid s) \, \nabla_\theta \log \pi_\theta(a \mid s) \, \mathrm{d}a \, \mathrm{d}s, \\ &= \mathbb{E}_{d_\pi^\gamma, \pi}[Q_\pi^\gamma(s, a) \, \nabla_\theta \log \pi_\theta(a \mid s)],\end{aligned}$$

for which we can derive sample-based estimators. One of the most famous algorithms that does precisely this is known as REINFORCE [183] and it uses Monte-Carlo sampling to estimate $Q^{\gamma}_{\pi}(s, a)$ and compute the policy gradient. Specifically, REINFORCE exposes a gradient of the form:

$$\nabla_{\theta} J(\theta) = \mathbb{E}_{d^{\gamma}_{\pi}, \pi}\left[G^{\gamma}_{t} \, \nabla_{\theta} \log \pi_{\theta}(a_t \mid s_t)\right],$$

which is equivalent since $Q^{\gamma}_{\pi}(s, a) = \mathbb{E}_{d^{\gamma}_{\pi}, \pi}[G^{\gamma}_{t}]$; note that we assume that the limits in Equation 2.19 exist and are well defined. These methods do not use value function approximation and thus do not suffer from bias, but they are prone to high variance. The reason for this is that each episode may yield very different outcomes for the same policy. The resulting distribution of returns may thus have lots of dispersion which adversely effects the quality of our estimate of the expected value.

One solution for this problem is to subtract a "baseline" from the estimator, such that the gradient becomes

$$\nabla_{\theta} J(\theta) = \mathbb{E}_{d^{\gamma}_{\pi}, \pi}\left[\left(G^{\gamma}_{t} - b_{\pi}(s_t)\right) \, \nabla_{\theta} \log \pi_{\theta}(a_t \mid s_t)\right],$$

This term acts as a control variate, reducing the variance in the policy gradient estimate [72] while leaving the bias unchanged. In general, there is no single optimal baseline such that

$$b_{\pi}(s) = \min_{b(\cdot)} \mathbb{V}\left[\left(G^{\gamma}_{t} - b(s_t)\right) \, \nabla_{\theta} \log \pi_{\theta}(a_t \mid s_t)\right]$$

as the solution involves an instance of the policy's point Fisher information matrix which can be non-invertible [49, 88, 128]. In practice, one typically uses something much simpler, such as an estimate of the value function $V^{\gamma}_{\pi}(s)$. This reduces the inner term, $G^{\gamma}_{t} - V^{\gamma}_{\pi}(s)$, to something that is equivalent to the advantage function, $A^{\gamma}_{\pi}(s, a)$, defined in Equation 2.23. The inclusion of a baseline like this in REINFORCE greatly reduces the noise in the policy updates, leading to much more stable learning and allowing for much higher learning rates. However, we can improve upon this further still by accepting some bias in the gradient estimate and using policy evaluation to estimate $Q^{\gamma}_{\pi}(s, a)$ in parallel.

2.4.3 *The Actor-Critic Architecture*

Actor-critic methods are a form of generalised policy iteration [162] in which we alternate between learning an estimate of the value function, and improving the policy. These combine the principles of both value-based and policy gradient methods introduced in the previous two sections, forming an important class of algorithms for optimising continuously differentiable policies, $\pi \in \Pi_{s,\theta}$. As in Section 2.4.2, we make use of the policy gradient in Equation 2.47, but replace the function Q^{γ}_{π} with a learnt estimate. As shown by Sutton, McAllester, Singh, and Mansour [164], this can be done without introducing bias if the function approximator satisfies certain conditions.

Theorem 2 (Policy gradient with function approximation)**.** *If the action-value approximator minimises the mean-squared error* (2.37) *(replacing V_{π} with Q_{π}, and $\widehat{V}$ with $\widehat{Q}$), and is compatible with the policy parameterisation, in the sense that*

$$\nabla_{w} \widehat{Q}_{w}(s, a) = \frac{\nabla_{\theta} \pi_{\theta}(a \mid s)}{\pi_{\theta}(a \mid s)}, \tag{2.48}$$

then

$$\nabla_{\theta} J(\theta) = \int_{\mathcal{S}} d^{\gamma}_{\pi}(s) \int_{\mathcal{A}(s)} \widehat{Q}_{w}(s, a) \, \nabla_{\theta} \pi_{\theta}(a \mid s) \, \mathrm{d}a \, \mathrm{d}s, \tag{2.49}$$

where the stationary distribution, $d^{\gamma}_{\pi}(s)$ is as defined in Equation 2.14.

Proof. See Sutton, McAllester, Singh, and Mansour [164]. ■

Generalised policy iteration does not come with the same guarantees as perfect theoretical policy iteration and may lead to a worse policy after each iteration of the actor step. As stated by Szepesvári [167], one often observes improvements early on in learning, but oscillatory behaviour near a fixed point. However, the actor-critic architecture has shown great empirical performance in many problem domains. The key advantage is that the critic $\widehat{Q}_{\boldsymbol{w}}(s, a)$ has much lower variance than that of the estimates used in actor-only algorithms. Furthermore, a small gradient step in policy parameters leads to smoother changes in the policy. This is compared to greedy methods where a small change in the value function can lead to discontinuous changes in the policy. A fantastic survey of actor-critic methods was written by Grondman, Busoniu, Lopes, and Babuska [74] who highlight that there are a smorgasbord of possible implementations, each with their pros and cons.

2.4.4 *Natural Policy Gradients*

One problem with "vanilla" policy gradient methods is that they often get stuck in local optima. This pathology was first identified by Kakade [87] who, inspired by the work of Amari [7], provide theoretical and empirical evidence that conventional methods get stuck in plateaus in objective space. Natural gradients on the other hand, denoted by $\widetilde{\nabla}_\theta J(\theta)$, avoid this by following the steepest ascent direction with respect to the Fisher metric associated with the policy's likelihood distribution rather than the standard Euclidean metric. This accounts for the fact that the parameters themselves do not occupy a flat manifold in general, but instead have a Riemmanian geometry. From this, the natural gradient can be expressed as

$$\widetilde{\nabla}_\theta J(\theta) \doteq \mathbf{G}^{-1}(\theta)\nabla_\theta J(\theta), \tag{2.50}$$

where $\mathbf{G}(\theta)$ denotes the Fisher information matrix

$$\mathbf{G}(\theta) \doteq \mathbb{E}_{d_\pi^\gamma, \pi}\left[\frac{\partial \log \pi_\theta(a \,|\, s)}{\partial \theta} \frac{\partial \log \pi_\theta(a \,|\, s)}{\partial \theta}^\top\right]. \tag{2.51}$$

Natural gradients have a particularly elegant form when combined with the actor-critic architecture and compatible function approximation [129]. In this case, the gradient $\widetilde{\nabla}_\theta J(\theta)$ is exactly equal to the advantage weights of the critic, $\boldsymbol{w}$; i.e. those that are associated with the features defined in Equation 2.48. This means that the policy update steps take the form $\theta \leftarrow \theta + \alpha \boldsymbol{w}$. The fact that we need not explicitly estimate the stationary distribution $d_\pi^\gamma(s)$ or the Fisher information matrix $\mathbf{G}(\theta)$ directly, nor perform the matrix inversion to compute $\widetilde{\nabla}_\theta J(\theta)$ is incredibly powerful. There are a number of advantages of this approach that have been highlighted by Peters and Schaal [129].

(i) Natural gradients preserve the convergence guarantees to a local minimum as for "vanilla" policy gradients.

(ii) Empirical evidence suggests that natural gradients exhibit faster convergence and avoid premature convergence.

(iii) The natural policy gradient is covariant in the sense that it is independent of the coordinate frame used to specify the parameters.

(iv) It averages out the stochasticity of the policy and thus requires fewer data points to obtain a high quality gradient estimate.

In this thesis we will mostly use the NAC-S(λ) algorithm introduced by Thomas [173]. This combines the advantages of the natural actor-critic architecture of Peters and Schaal [129] with the incremental performance of the SARSA(λ) algorithm. While it is claimed that this unbiased variant can exhibit poor data-efficiency for small γ, we found it be highly effective in the (mostly) finite time horizon problems studied herein.

3 ALGORITHMIC TRADING

3.1 FINANCIAL MARKETS

The history of financial markets is one fraught with tales of riches, ruin and controversy. As early as the 1300s, moneylenders based in Venice would trade in debt between themselves, governments and individual investors. These were, in some sense, revolutionaries of their time, and pioneered the notion of brokering which underpins much of the trading activity seen today. Over the 700 years that followed, markets and exchanges evolved into something much more significant — often born out of necessity and the demand for globalisation, they became the centre of financial activity for many nations across the globe. Nowadays, the London Stock Exchange, New York Stock Exchange, and many other venues, host myriad different markets in which a dizzying array of goods and services are traded. In a world of incredible consumption, these financial markets represent one of few viable means of exchange on a global scale.

It's worth noting that exchanging goods through a medium like money is only one paradigm of trade. The notion of bartering, in which individuals exchange only in corporeal goods and services, dates back much further to ancient times.

Perhaps the most commonly understood type of asset that is traded on financial markets, besides currencies themselves, is *common stock* (or simply stock/shares). Stock represents a claim of partial ownership over a corporation and is typically leveraged by firms as a means of raising capital for growth. The "fundamental value" of shares derives from the contract they represent. Owning shares in a company entitles the holder to a portion of the profits and the right to vote in key decision-making. For example, holding 1000 units of Vodafone Group plc (costing ~£1300) would have earnt approximately £40 in dividends on the value of the stock in February 2020. Besides stocks, there are markets for trading commodities, currencies, debt, futures, derivates, and many other increasingly intangible assets. One can even buy and sell shared partnership in mutual funds, which are professionally managed portfolios of assets designed, typically, to track/reflect certain characteristics of the economy. While this may all seem somewhat bewildering, the cornucopia of instruments seen on today's exchanges exists for two key reasons: trading and risk management.

Not a recommendation. . .

Trading refers to the activity of market participants in the buying and/or selling of assets, the motivations of which vary greatly between individual traders. In general, the literature categorises the behaviour of traders into three classes:

FUNDAMENTAL TRADERS buy and sell based on principled economic factors that are exogenous to the exchange itself. These are often referred to as *noise* traders since their activity is predominantly uninformative on short time-scales.

INFORMED TRADERS leverage information that is inaccessible to the market and has predictive power over the appreciation/depreciation of an asset.

MARKET MAKERS are passive traders who facilitate transactions between participants without any particular preference over the direction, instead exploiting their ability to execute trades at favourable prices.

When there is sufficient activity on a market, the local interactions between these heterogeneous agents can lead to the emergence of highly complex phenomena and regularity. Indeed this is the subject of a great deal of research in economics and market microstructure.

3.1.1 *Electronic Markets*

Electronic markets emerged in the late 20th century with the advent of modern computing and the subsequent demand for increased automation and systematisation of trading strategies, as well as decentralisation. Nowadays, electronic markets are by the far the most active trading venues in the world. In Germany, for example, the Xetra exchange commands over 90% of all securities activity, and 68% of the volume of German blue-chip stocks on a European level [137]. Other venues such as the NASDAQ and Chicago Board Options Exchange boast similar dominance across a host of asset classes.

At their core, electronic markets do two things: they offer a platform for traders to signal their intentions to buy or sell; and provide a means by which participants are matched and trades resolved. On the majority of modern, electronic markets this takes the form of a continuous double auction implemented as a limit order book (LOB) — or some hybrid variation thereof — in which traders are able to execute with the level of immediacy they desire.

An important consequence of electronic markets and the rise of algorithmic trading (for better or worse) is the ability to trade on incredibly short time-scales. This practice, generally referred to as high-frequency trading, is characterised by the use of sophisticated algorithms and exceptionally fast, direct access to the exchange. Their technological advantage allows high-frequency traders to respond to new information and changes in the market faster than their competition. In many cases, these firms operate as market makers to an exchange, providing liquidity and facilitating transactions in the process of trading. This has the effect of stabilising a market, improving efficiency and leading to better prices for both fundamental and informed traders. The bulk of this thesis focusses on precisely this setting.

High-frequency trading is a double-edged sword — many believe it to be responsible for the 2010 Flash Crash [91].

3.2 A CALCULUS FOR TRADING

Quantitative/mathematical finance is the field of mathematics that studies problems in financial markets and the modelling challenges therein. One of the most common uses for this family of techniques is in the derivation of optimal trading strategies and in the pricing of assets (such as derivatives). Indeed, many of these problems can be expressed using a formalism (i.e. calculus) that operates in the space of *portfolios of assets*. The *trader* (or, equivalently, the agent) is assumed to operate in some universe in which there is a single riskless asset, the numéraire, whose value remains fixed, and $n > 0$ risky assets whose prices evolve stochastically, such as stocks, commodities, futures, etc. For this we define the following two temporal processes:

Note in the real world there is no such thing as a riskless asset.

Definition 1 (Cash process)**.** *Denote by* $X_t \in \mathbb{R}$ *the trader's cash holdings, the numéraire.*

Definition 2 (Inventory process)**.** *Let* $\boldsymbol{\Omega}_t^{\pm} \geqslant 0$ *be the ask* ($-$) *and bid* ($+$) *components of the cumulative volume flow arising from interactions between the trader and market. The trader's holdings in the* n *assets at time* t *are then defined as* $\boldsymbol{\Omega}_t \doteq \boldsymbol{\Omega}_t^+ - \boldsymbol{\Omega}_t^-$, *where*

$$\boldsymbol{\Omega}_t = \begin{bmatrix} \Omega_t^{(1)} & \Omega_t^{(2)} & \cdots & \Omega_t^{(n)} \end{bmatrix}^\top .$$

For $n = 1$*, we define the scalar notation* $\Omega_t \doteq \boldsymbol{\Omega}_t \in \mathbb{R}$.

These quantities, along with the corresponding prices for the n assets, defined below, form the foundation of mathematical finance in the sense that they define the space over which strategies are optimised. Trading itself is merely the act of converting between two or more of these “units of measure” at an exchange rate given by said prices relative to the numéraire.

Definition 3 (Asset prices). *Let $\mathbf{Z}_t \in \mathbb{R}_+^n$ define an n-dimensional column vector of non-negative asset prices,*

$$\mathbf{Z}_t = \begin{bmatrix} Z_t^{(1)} & Z_t^{(2)} & \cdots & Z_t^{(n)} \end{bmatrix}^\top.$$

For $n = 1$, we define the scalar $Z_t \doteq \mathbf{Z}_t \in \mathbb{R}_+$.

The goal of the trader is to interact with the market(s) by buying and selling these n instruments such that it’s objective is maximised. In the case of portfolio optimisation, for example, the agent must choose a vector $\mathbf{\Omega}_t$ that achieves the most appreciation in value; hence why predicting $\mathbf{Z}_t$ is such an important topic in algorithmic trading. But how do we define the overall value of this agent’s holdings? Typically, we define this quantity under the assumption of perfect liquidity which, while seldom true in practice, yields the simple expression given below in Equation 3.1.

Definition 4 (Portfolio value). *The mark-to-market portfolio value of the agent’s holdings is defined as*

$$\Upsilon_t \doteq X_t + \langle \mathbf{\Omega}_t, \mathbf{Z}_t \rangle . \tag{3.1}$$

EXAMPLE As an illustration of these core concepts, consider a scenario in which there are three risky assets whose prices evolves according to the following stochastic differential equation:

$$d\mathbf{Z}_t = \begin{bmatrix} -0.5 \\ 0.5\left(1.5 - Z_t^{(2)}\right) \\ 3 \end{bmatrix} dt + \begin{bmatrix} 5 \\ 15 \\ 25 \end{bmatrix} \circ d\mathbf{W}_t, \tag{3.2}$$

where $\mathbf{W}_t$ is a standard 3-dimensional Wiener process. The first and last entries are simple random walks, and the second is an Ornstein-Uhlenbeck process. Figure 3.1 shows one sample path of these random processes with the wealth trajectories (i.e. Υ_t) of four possible portfolios, $\mathbf{\Omega}_t$. Observe how the portfolios — each an element of a linear subspace over assets — can give rise to very different wealth sequences. Finding a good portfolio becomes increasingly difficult as n grows and the individual dynamics processes become more complex.

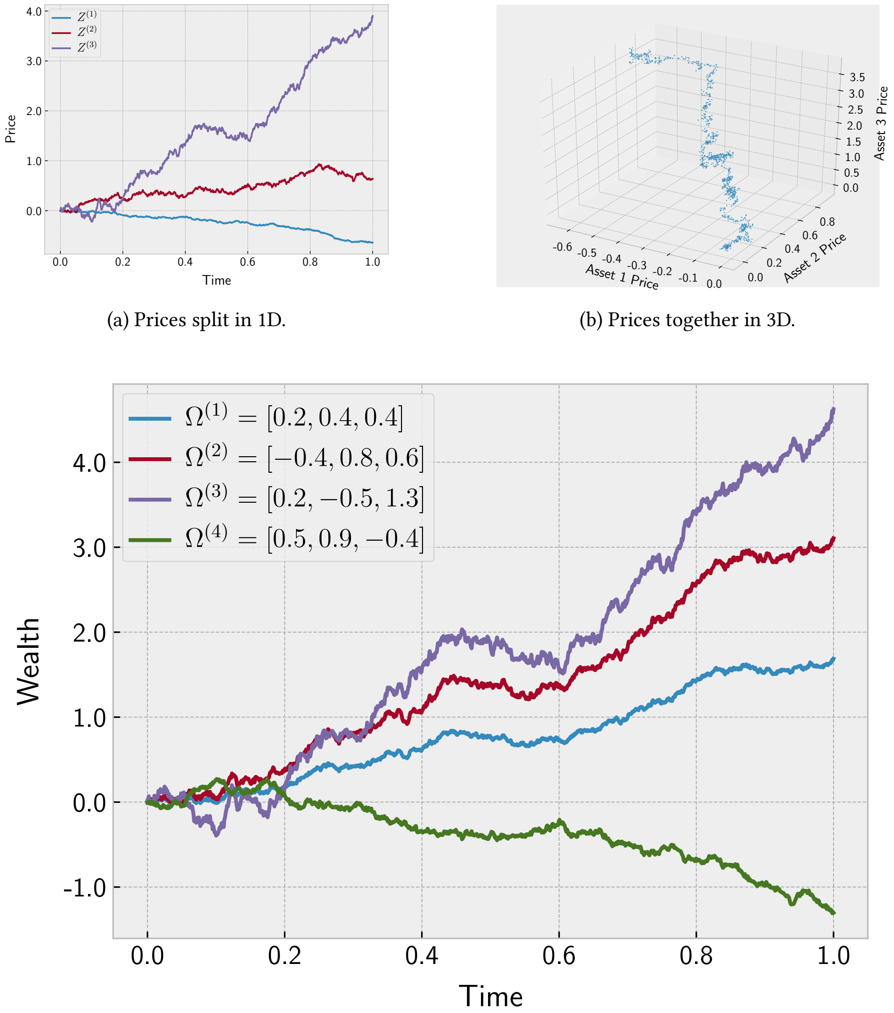
Even defining what is a “good” portfolio is fundamentally non-trivial.

3.2.1 *Interactions*

Up until this point we have not considered how the agent actually interacts with the market. For this, we first define the trading rate as the rate of change of the agent’s inventory process over time.

Definition 5 (Trading rate). *Let $\mathbf{\nu}_t^\pm \doteq \frac{d\mathbf{\Omega}_t^\pm}{dt}$ and define the trading rate as $\mathbf{\nu}_t \doteq \frac{d\mathbf{\Omega}_t}{dt} = \mathbf{\nu}_t^+ - \mathbf{\nu}_t^-$, with $\nu_t \doteq \mathbf{\nu}_t$ in the unidimensional case.*

Remark. *The trading rate $\mathbf{\nu}_t^\pm$ is rarely well behaved in practice. Not only is trading in continuous-time highly ineffective in the presence of trading costs (and also practically impossible), but we also know that the inventory process $\mathbf{\Omega}_t$ tends to exhibit jumps and non-linearities from batch transactions. It may therefore be more natural to talk about sub-derivatives when translating real algorithms into the nomenclature, but we leave this to treatises on pure mathematical finance.*



(a) Prices split in 1D.

(b) Prices together in 3D.

(c) Wealth curve for example portfolios.

Figure 3.1: Illustration of a price process drawn for the assets in Equation 3.2 with the associated wealth series of four sample inventories weights.

If $|\nu_t| > 0$, then, assuming the price is non-zero, there must be a corresponding change in cash. For example, if one intended to purchase k units of AAPL, then there would be an exchange between the two parties of shares and cash. The rate of change of cash over time may thus be defined in terms of the trading rates, $\boldsymbol{\nu}_t^{\pm}$.

Definition 6 (Cash flow)**.** *Define the rate of change of cash as* $\upsilon_t \doteq \frac{dX_t}{dt}$*, the breakdown by asset as* $\boldsymbol{\upsilon}_t$*, and the further refinement* $\boldsymbol{\upsilon}_t^{\pm} \geqslant 0$ *as the positive/negative cash flow arising from trading, such that* $dX_t \doteq \left\langle \mathbf{1}, \boldsymbol{\upsilon}_t^{+} - \boldsymbol{\upsilon}_t^{-} \right\rangle dt = \langle \mathbf{1}, \boldsymbol{\upsilon}_t \rangle \, dt$.

It follows from the definition above and the chain rule of differentiation that the cash flow may be expressed as

$$\upsilon_t = \left\langle \frac{dX_t}{d\boldsymbol{\Omega}_t}, \frac{d\boldsymbol{\Omega}_t}{dt} \right\rangle = \left\langle \boldsymbol{\nu}_t, \frac{dX_t}{d\boldsymbol{\Omega}_t} \right\rangle.$$

This leads us to define a further quantity: the effective transaction prices, $\frac{dX_t}{d\boldsymbol{\Omega}_t}$, express the sensitivity of the market's prices, $\mathbf{Z}_t$, to the agent's trading rate. In general, this value can be separated into the combination of both temporary and permanent price impact. The latter corresponds to irreversible, hysteretic changes in the prices $\mathbf{Z}_t$ and may be expressed as the $n \times n$ matrix of derivatives

$$\frac{d\mathbf{Z}_t}{d\boldsymbol{\nu}_t} = \begin{bmatrix} \frac{\partial Z_t^{(1)}}{\partial \nu_t^{(1)}} & \cdots & \frac{\partial Z_t^{(1)}}{\partial \nu_t^{(n)}} \\ \vdots & \ddots & \vdots \\ \frac{\partial Z_t^{(n)}}{\partial \nu_t^{(1)}} & \cdots & \frac{\partial Z_t^{(n)}}{\partial \nu_t^{(n)}} \end{bmatrix}. \tag{3.3}$$

The former — i.e. the premium paid which does not have any long-lasting impact on prices — is then given by the difference

$$\frac{dX_t}{d\boldsymbol{\Omega}_t} - \underbrace{\frac{d\mathbf{Z}_t}{d\boldsymbol{\nu}_t}\mathbf{1}}_{\mathbf{S}}, \tag{3.4}$$

where $\mathbf{1}$ is an $n \times 1$ sum vector. Here $\mathbf{S}$ is a vector containing the sum over each row in the permanent impact matrix; i.e. the total impact on each asset due to $\boldsymbol{\nu}_t$. One typically assumes that the permanent price impact matrix is diagonal such that trading on one asset has negligible impact on another. This would correspond to the special case where $\mathbf{S} = \operatorname{diag} \frac{d\mathbf{Z}_t}{d\boldsymbol{\nu}_t}$.

SELF-FINANCING STRATEGIES In a perfect, frictionless market the effective transaction prices are necessarily equal to the market prices: $\frac{dX_t}{d\boldsymbol{\Omega}_t} = \mathbf{Z}_t$. This implies that the permanent impact matrix contains only zeros, and that there is no temporary premium for trading. In other words, the cash, X_t, and stock, $\boldsymbol{\Omega}_t$, can be exchanged perfectly, such that $dX_t = -\langle \boldsymbol{\nu}_t, \mathbf{Z}_t \rangle \, dt$. A strategy $\boldsymbol{\nu}_t$ that satisfies this constraint is known as a *self-financing strategy*, and the tuple $(X_t, \boldsymbol{\Omega}_t)$ is called a *self-financing portfolio*. A direct consequence of this requirement is that the evolution of the mark-to-market portfolio value (Equation 3.1) must satisfy the stochastic differential equation

$$\begin{aligned} d\Upsilon_t &= dX_t + \langle \boldsymbol{\nu}_t, \mathbf{Z}_t \rangle \, dt + \langle \boldsymbol{\Omega}_t, d\mathbf{Z}_t \rangle, \\ &= \langle \boldsymbol{\Omega}_t, d\mathbf{Z}_t \rangle. \end{aligned} \tag{3.5}$$

Satisfying this relation means that there can be no exogenous infusion or withdrawal of money from the system. Changes in the value of the portfolio may only be derived from the change in the value of the underlying assets; i.e. changes in $\mathbf{Z}_t$. The implications of this are the same as that of the First Law of Thermodynamics, that energy can be neither created nor destroyed. This, of course, is only true in the frictionless case. Less idealised markets, such as LOB markets (Section 3.3), or those with transaction fees, have similar constraints but they are never quite so simple. Transactions between assets must be financed only by the sale or purchase of another within the portfolio.

Remark. *If one treats the market itself as an agent, then it is clear to see that there are two conserved quantities during interactions: (i) the total value and; (ii) the total outstanding of each asset. This follows from the fact that transactions are zero-sum with respect to the current market prices. This is equivalent to the notions of mass and energy in Physics, and suggests that the numinous insights of Noether [120] hold weight even in financial settings. Put simply, these two conserved quantities give rise to exactly two symmetries in the dynamics of the system and thus a frictionless market can always be formulated as a symmetric, zero-sum game. Indeed, any financial market can be symmetrised into a zero-sum game by the inclusion of an extra "environmental" player.*

3.2.2 *Time Discretisation*

While continuous time representations are certainly more elegant, the majority of problems discussed in this thesis are in the discrete-time setting. Indeed RL is, for the most part, constrained to operate in discrete intervals. To this end, we outline a framework for translating from the former to the latter which will prove especially important in Part III. First, we define the change in a time-dependent function f_t from $t \to t+1$ by the notation

$$\Delta f_t \doteq f_{t+1} - f_t.$$

Note that there is some abuse of notation here which is introduced for the sake of notational clarity. First, time points are now treated as discrete indices — i.e. $t \in \mathbb{N} \cup \{0\}$ — but have thus far been treated as a real values. Second, the term Δf can technically refer to *any change* in the function f, not just in time. The presence of a subscript t will be used to disambiguate when necessary.

The difference operator defined above satisfies a number of the same important properties as the conventional derivative. For example, the difference in time of a time-independent function or constant is clearly zero: $\Delta x = 0$. The difference in the product of two or more functions can be expressed as the sum over the scaled changes of each term independently and the one cross-term,

$$\Delta(f\,g) = g\,\Delta f + f\,\Delta g + \Delta f\,\Delta g.$$

The anti-derivative is also intuitively defined as the summation. The list goes on, and we refer the reader to the excellent work of Graham, Knuth, Patashnik, and Liu [71] for further details on discrete calculus. As an example, one can use the properties of the finite derivative to express the change in portfolio value in discrete time as

$$\begin{aligned}\Delta\Upsilon_t &= \Delta X_t + \Delta\left\langle \mathbf{Z}_t, \boldsymbol{\Omega}_t \right\rangle, \\ &= \Delta X_t + \left\langle \Delta\mathbf{Z}_t, \boldsymbol{\Omega}_t \right\rangle + \left\langle \mathbf{Z}_{t+1}, \Delta\boldsymbol{\Omega}_t \right\rangle,\end{aligned}$$

which is almost identical to the continuous-time variant. The key difference is that the final term includes the price at the next time step, Z_{t+1}.

3.3 LIMIT ORDER BOOKS

The vast majority of electronic markets operate as continuous double actions with a limit order book (LOB) as the matching mechanism. The purpose of this object is to

	Price	**Ask**
	101.00	12
	100.50	13
	100.25	
35	100.00	
3	99.75	
11	99.50	
Bid		

Figure 3.2: Snapshot of a limit order book with multiple price levels occupied by bid or ask orders and a total volume.

match the buyers and sellers of a given asset. An example is illustrated in Figure 3.2 which depicts six price levels of an LOB instantiation. The book has two sides (hence double auction): one for asks (sell requests); and one for bids (buy requests). Each of the levels in the two books are unique and have an associated volume corresponding to the cumulation of all passive requests to buy/sell at said price. Prices are, in effect, discrete because there is a minimum increment between each value called the *tick size*, ι, which is specified by the exchange; in Figure 3.2, $\iota \doteq 1/4$.

Prices are almost always defined as rational numbers in practice.

In the following we present a mathematical formulation of an LOB for the case of a single asset ($n = 1$); this will closely follow the excellent survey of Gould, Porter, Williams, McDonald, Fenn, and Howison [70]. The extension to multiple assets is trivial but the subscripts can get unwieldy which detracts from the purpose of this section. To begin, we define the fundamental building blocks of limit order book (LOB) markets.

Definition 7 (Limit order). *A limit order, o, is a tuple containing the time of submission, size and price: $o \doteq (t_o, \omega_o, z_o)$.*

Definition 8 (Limit order book). *An LOB, $\mathcal{B}_t$, is the set of all active orders in the market at time t.*

The evolution of the LOB is a strictly càdlàg process such that an order o placed at t_o is present in the book at t_o, $o \in \mathcal{B}_{t_o}$, but not before: $o \notin \lim_{t' \uparrow t_o} \mathcal{B}_{t'}$; where $\lim_{t' \uparrow t_o}$ denotes the left limit. This ensures that an order submitted at time t_o does not appear chronologically in the book until precisely t_o. The LOB collection can be partitioned into subsets of ask and bid orders, $\mathcal{B}_t^+ \doteq \{o \in \mathcal{B}, \omega_o > 0\}$ and $\mathcal{B}_t^- \doteq \{o \in \mathcal{B}, \omega_o < 0\}$, respectively. Given a non-empty instance $\mathcal{B}_t$, a market order (MO) is then defined as a special case of the LO tuple.

Definition 9 (Market order). *An ask (bid) market order, $o \doteq (t_o, \omega_o, z_o)$, is a special case of an LO with price $z_o \doteq \min_{o' \in \mathcal{B}_t^-} z_{o'}$ ($\max_{o' \in \mathcal{B}_t^+} z_{o'}$, respectively).*

As required, the definition above for market orders (MOs) is only valid when there are active LOs on the opposing side of the book. That is, an ask (respectively, bid) MO can only be executed — or is even well-defined — when there are bid (ask) LOs in $\mathcal{B}_t$.

PRICES The occupied prices in an LOB form a highly important set of values for many algorithmic trading strategies. Here, we define a price level as being occupied if there is at least one order with the given price and a non-zero volume. From this we may define two key quantities that describe the arrangement of active orders in the book. The most important of these definitions is that of the two *best prices*.

Definition 10 (Best prices)**.** *Let $\mathcal{B}_t$ be an LOB instance at time t. The best ask and bid prices are then defined, respectively, as*

$$Z_t^+ \doteq \min_{o \in \mathcal{B}_t^+} z_o, \quad \textit{and} \quad Z_t^- \doteq \max_{o \in \mathcal{B}_t^-} z_o. \tag{3.6}$$

The levels associated with the prices $Z_t^\pm$ are often referred to as the *top of the book* and they facilitate the definition of two fundamental properties from the market microstructure literature: the market mid-price and bid-ask spread. The former is most commonly used as an estimate of the latent price of an asset, Z_t; though there are many others one can construct. The latter is used as a measure of the liquidity contained in a book. In general, larger values imply that passive traders are seeking a greater premium against which to offset the risk. In highly liquid markets, this becomes very tight since trades rarely have a significant impact on price; we will see more of this in the next section.

Definition 11 (Mid-price)**.** *The market mid-price is defined as the average between the best ask and bid prices:*

$$\widetilde{Z}_t \doteq \frac{Z_t^+ - Z_t^-}{2}. \tag{3.7}$$

Definition 12 (Bid-ask spread)**.** *The bid-ask spread is defined as the difference between the best ask and bid prices:*

$$D_t \doteq Z_t^+ - Z_t^-. \tag{3.8}$$

VOLUMES The next most important property of the book that is of interest to algorithmic traders is the volume distribution. For any given price, we define the available liquidity, formally, as the sum of volumes of each order occupying the given level. For this, we define the *market depth.*

Definition 13 (Market depth)**.** *Let $\mathcal{B}_t(z) \doteq \{o \in \mathcal{B}_t : z_o = z\}$ for any price $z \in [0, \infty)$ and book (including subsets). Then, with some abuse of notation, define*

$$\omega_t^\pm(z) \doteq \sum_{o \in \mathcal{B}_t^\pm(z)} \omega_o \tag{3.9}$$

as the total ask/bid volume at a price z and time t.

3.3.1 *Matching*

When an MO is submitted to an LOB the engine performs a simple operation: walk the book and consume volume up to the amount specified. This is a well defined procedure that we will formalise shortly, but, in the case of an LO, this process is not quite as simple. Consider the book in Figure 3.2 and an incoming ask LO, o, with price $z_o \doteq 101$ and size $\omega_o \doteq 48$. In this case we trivially add the order to the book, yielding an increase in the volume $\omega^+(z_o)$ from 12 to 60. However, what happens if this order was instead submitted with a price, $z_o \doteq 100$, overlapping the top of the bid book? In this case the matching engine applies the procedure below.

(i) Treat the incoming LO as an MO with a limiting price level, k, given by the price z_o and walk the book. In this case, $k = 0$ as the LO only touches the top of the bid book.

(ii) If the LO is exhausted then we simply stop the process.

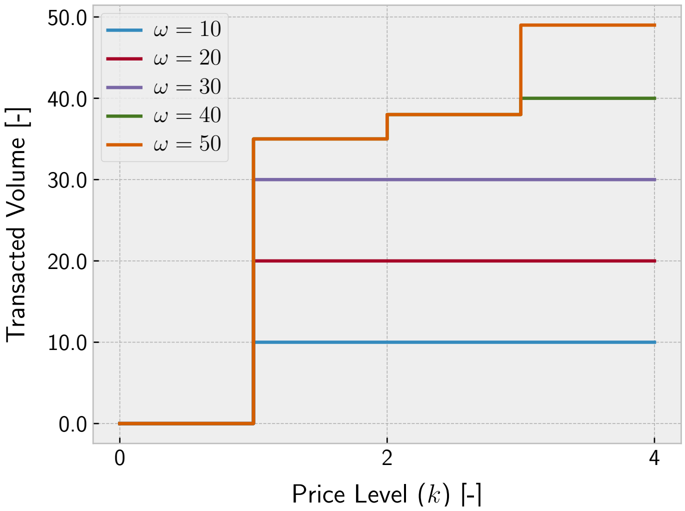


Figure 3.3: Illustration of book walking sequences, $\widetilde{\omega}_k$, for $k \in \{0, \ldots, 4\}$ and five initial sizes. The limiting values here define the total consumed volume by each MO.

(iii) If the remaining volume of the LO is non-zero, then the volume at all levels up to k must have been consumed and we add the order to the now empty queue at z_o. Note that the new price level will have changed type from bid to ask.

Walking the book starts with an arbitrary market order (MO) of size $|\omega| > 0$. The cash flow generated by executing this order is computed by incrementally walking the book and consuming liquidity at each level, starting from the best available price, until the request is satisfied or no liquidity remains. To define this formally, we first express the total volume consumed by the sequence:

$$\widetilde{\omega}_0 \doteq \min\left\{\omega,\, \omega^{\pm}(Z^{\pm})\right\}, \tag{3.10}$$

$$\widetilde{\omega}_1 \doteq \min\left\{\omega,\, \widetilde{\omega}_0^{\pm} + \omega^{\pm}(Z^{\pm} \pm \iota)\right\},$$

$$\vdots$$

$$\widetilde{\omega}_k \doteq \min\left\{\omega,\, \widetilde{\omega}_{k-1}^{\pm} + \omega^{\pm}(Z^{\pm} \pm k\iota)\right\}, \tag{3.11}$$

where the subscript k should not be confused with the time t which has been dropped here for brevity. This can be seen as a "partial integral" over the volume in the LOB with a saturation point at ω. Note that since Equation 3.11 is an increasing function of k, the supremum (i.e. the total volume consumed) is given by the limit $\widetilde{\omega} \doteq \sup_k \widetilde{\omega}_k = \lim_{k\to\infty} \widetilde{\omega}_k$. Intuitively, the total consumed volume $\widetilde{\omega}$ is the minimum of ω and the total volume in the book; see Figure 3.3 for instance.

An analogous process is defined for the cash flow generated by the transaction:

$$\widetilde{x}_0 \doteq Z^{\pm}\widetilde{\omega}_0^{\pm}, \tag{3.12}$$

$$\widetilde{x}_1 \doteq \widetilde{x}_0^{\pm} + \left(Z^{\pm} \pm \iota\right)\left(\widetilde{\omega}_1^{\pm} - \widetilde{\omega}_0^{\pm}\right),$$

$$\vdots$$

$$\widetilde{x}_k \doteq \widetilde{x}_{k-1}^{\pm} + \left(Z^{\pm} \pm k\iota\right)\left(\widetilde{\omega}_k^{\pm} - \widetilde{\omega}_{k-1}^{\pm}\right). \tag{3.13}$$

As with Equation 3.11, the process $\widetilde{x}_k$ takes the form of a (price-weighted) saturated integral over the LOB, and is again increasing as a function of k. In this case, the limit $\widetilde{x} \doteq \sup_k \widetilde{x}_k = \lim_{k\to\infty} \widetilde{x}_k$ defines the total cash generated (positive or negative) by sequentially consuming liquidity in the book.

3.3.2 *Revenue*

Unlike the idealised setting discussed in Section 3.2.1, LOB markets do not satisfy the traditional self-financing constraint in Equation 3.5. Here, a strategy consists of limit orders (LOs) and market orders (MOs), not just a continuous trading rate ν_t. These orders may be fully or partially executed at any given time, and the price one pays for transacting is almost surely not equal to the mid-price $\widetilde{Z}_t$, let alone the latent price Z_t. Altogether, these problems confound, making it very difficult to express an exact self-financing equation for the LOB setting. Nevertheless, if we assume one does indeed exists, then we can derive some useful formulae for quantifying the revenue of a strategy.

Generally, Equation 3.1 may be decomposed into two terms measuring the profit and loss due to: (a) transacting at prices away from $\mathbf{Z}_t$ (the *spread PnL*); and (b) changes in the value of the holdings $\mathbf{\Omega}_t$ (the *inventory PnL*). These are derived trivially by expanding the change in mark-to-market portfolio value and factoring terms as follows:

$$d\Upsilon_t = \underbrace{dX_t + \langle \boldsymbol{\nu}_t, \mathbf{Z}_t \rangle \, dt}_{\text{Execution}} + \underbrace{\langle \mathbf{\Omega}_t, \, d\mathbf{Z}_t \rangle}_{\text{Speculation}} . \tag{3.14}$$

In the LOB setting, the implied trading rate $\boldsymbol{\nu}_t$ above will not behave in a simple manner, and is driven by the processes discussed in the previous section. This formula will feature prominently in Part II and Part III for defining rewards.

Remark. *This problem was studied by Carmona and Webster [30] who performed an empirical analysis of NASDAQ order book data and arrived at the (uni-asset) condition:*

$$d\Upsilon_t \approx \Omega_t \, dZ_t \pm \frac{D_t \nu_t}{\sqrt{2\pi}} + d[\Omega, Z]_t,$$

where $\pm$ is a $+$ for limit orders and $-$ for market orders, and whenever trading with LOs the additional constraint that $d[\Omega, Z]_t < 0$ is imposed; here $[\Omega, Z]_t$ denotes the quadratic covariation between Ω_t and Z_t. We omit the details here and instead refer to the reader to the original text. The main takeaway is that describing the evolution of Υ_t in a general and consistent way is non-trivial in the high-frequency domain.

3.4 DESIDERATA

In the previous sections we developed a formalism for algorithmic trading in LOBs and derived expressions for the revenue generated therein. This is crucial for measuring the performance of a trading strategy and establishing an ordering over possible implementations; this is analogous to the ordering over policies introduced in Equation 2.25. Indeed, it is quite natural to assert that the best strategies make the most money; i.e. maximise Υ_t for some t. This greedy criterion is, however, a somewhat limited perspective and there exist many other — often conflicting — desiderata that one must account for when designing trading strategies.

Most quantities of interest are inherently stochastic in nature. The fundamental challenge for traders is precisely the fact that we don't know the future. Instead, we rely on distributional properties to describe how a random variable behaves in aggregate. These properties, known as the statistical moments, are used to inform decision-making in most financial contexts. They will be used liberally throughout the thesis using the standard notation of $\mathbb{E}[X]$ and $\mathbb{V}[X]$ for the expected value and variance on the random variable X, respectively.

PORTFOLIO VALUE The distribution of a trading strategy's aggregate portfolio value by the terminal time step, Υ_T, is the most fundamental variable used to

Table 3.1: Portfolio characteristics for the four instances illustrated in Figure 3.1. Each value is computed using 10000 Monte-Carlo samples. In order, we have mean and variance on the portfolio value, the first lower partial moment, and the Sharpe ratio, respectively.

	$\mathbb{E}[\Upsilon_T]$	$\mathbb{V}[\Upsilon_T]$	$\mathbb{M}_-[\Upsilon_T]$	$\mathbb{E}[\Upsilon_T]/\sqrt{\mathbb{V}[\Upsilon_T]}$
$\Omega^{(1)}$	1.33	0.12	1.19	3.84
$\Omega^{(2)}$	2.46	0.32	2.24	4.35
$\Omega^{(3)}$	3.51	1.08	3.10	3.38
$\Omega^{(4)}$	-0.92	0.22	-1.11	-1.96

gauge performance. In particular, we consider the first and second moments of this distribution — i.e. $\mathbb{E}[\Upsilon_T]$ and $\mathbb{V}[\Upsilon_T]$ — where an agent would generally aim to maximise the former while minimising the latter. For example, $\Omega_t^{(1)}$ in Figure 3.1 yields an mean profit and loss of $\mathbb{E}\left[\Upsilon_T^{(1)}\right] = 1.33$ with a variance of $\mathbb{V}\left[\Upsilon_T^{(1)}\right] = 0.12$.

SHARPE RATIO An important metric used in financial literature and in industry is the Sharpe ratio. This is defined using the first and second moments of Υ_T, as shown below:

$$\frac{\mathbb{E}[\Upsilon_T] - \mu_{RF}}{\mathbb{V}[\Upsilon_T]}, \tag{3.15}$$

where μ_{RF} denotes a “risk-free rate” which we typically take to be zero. In essence, this quantifies the expected amount of wealth that can be earnt per unit of risk. While larger values are better, it is important to note that the Sharpe ratio is not a sufficient statistic on its own .

Lo [101] provides a thorough examination of the subtleties of using the Sharpe ratio that is worth reading.

INVENTORY The distribution of terminal inventory, Ω_T, tells us about the robustness of the strategy to adverse price movements. As always, we can describe this using various moments, such as the expected value, $\mathbb{E}[\Omega_T]$, and the variance, $\mathbb{V}[\Omega_T]$. In practice, most trading strategies aim to finish the trading day with small absolute values for Ω_T; though this is not always a strict requirement.

PRICING AND EFFICIENCY The “competitiveness” of a market maker — a trader who is characterised simultaneously buying and selling — is often discussed in terms of the average quoted spread. This is defined as the expected value of the difference in price between the market maker’s best bid and ask LOs. Tighter spreads imply more efficient markets since it incurs a lower cost to the counterparty compared with the (unattainable) market mid-price. Exchanges often provide compensation in the form of rebates for smaller spreads, and tout this value as a means to attract traders to their platform.

Example

For example, consider the four wealth curves illustrated in Figure 3.1. While we can safely say that the fourth portfolio performs the worst out of them all, it is not as clear cut which one you should pick between the first three; see Table 3.1. Notice that while the final value is higher for $\Omega^{(3)}$, it suffers from greater variability for the duration of the simulation. So much so, that the relative portfolio value drops below zero for almost 20% of the episode. This is reflected in a lower Sharpe ratio and higher downside risk (measured by the lower partial moment). Portfolios one and two, on the other hand, are more consistent while still generating significant profit. In expectation, these strategies may behave very differently — as we shall see — but this example highlights the notion that humans are naturally risk-averse (though to different degrees) and that the risk (and other secondary characteristics) associated with a strategy is an important consideration; see Tversky and Kahneman [176].

Part II

DATA-DRIVEN TRADING

4 LIMIT ORDER BOOK SIMULATION

4.1 OUTLINE

Reconstruction of limit order books (LOBs) is an effective strategy for simulating financial markets with minimal assumptions on the underlying mechanics governing the behaviour of the participant agents. Rather than specify the dynamics of the system in a probabilistic framework, we simply replay the events as they occurred in the past. Now, while there are some caveats, this type of simulation has been used extensively in the literature and in practice with great success. The key challenge is implementation.

In this chapter, we survey some of the computational aspects of limit order book (LOB) reconstruction. To begin, we cover the three key classes of data that are available to practitioners and researchers, their pros, cons and applicability. We will then discuss implementation-level details that are crucial for efficient experimentation, and define a suite of technical indicators that can be derived from such a framework. In Section 4.5, we conclude the chapter by discussing some of the limitations of this approach in terms of factual versus counterfactual simulation, and how we can remedy this.

4.2 DATA

In algorithmic trading — especially in the context of quantitative finance — data is at the core of the development of any strategy. Whether that's through market reconstruction (as we discuss in this chapter), or through the estimation of model parameters (as in Part III). Given our mechanistic assumptions, one key objective is to ensure that strategies are calibrated to the true market dynamics. However, not all data is created equal, and there are various "levels" of information available which come at increasing cost to the consumer. We discuss these below.

LEVEL 1 The most primitive type of LOB data contains only the price and volume available at the top of the book: $Z_t^{\pm}$ and $(\omega_t^+(Z_t^+), \omega_t^-(Z_t^-))$. This tells us very little about the overall shape of the book, but allows us to compute statistics such as the mid-price and spread which are important. In Figure 3.2, for example, L1 data would constitute only the volume at prices 100 and 100.5. This type of data can be very noisy due to the overwhelming number of cancellations that occur at the top of the book [70]. As a result, L1 data provides very few "features" that can be used to predict changes in price in the mid- to long-term, nor sufficient information to define an effective observation space for RL. It would be ill-advised, if not impossible, to base a trading strategy such as market making on this data alone.

LEVEL 2 At the next level of fidelity, L2 data — also known as market depth data — provides the full scope of ask and bid prices in an aggregated format: i.e. full knowledge of $\omega_t^{\pm}(\cdot)$. That is, the total volume at each occupied level in the book. Whenever an update occurs (transaction, cancellation, placement, etc...), the data will reflect the overall change at the affected levels, but not which orders were mutated. This amounts to having access to snapshots of the book that look like Figure 3.2, but not $\mathcal{B}_t$ in its entirety. This allows trading algorithms to leverage information about the shape and distribution of volume in the book. For example, one can measure the dispersion of volume in the book, or even the skew between ask and bid. These measures all have use in predicting short-term price changes and have been used extensively in the literature for designing more effective trading algorithms [32, 69].

LEVEL 3 The last, and most fine-grained type of data — known as L3 or market-by-order data — provides information about each and every order in the book. This could be in the form of raw FIX messages, or some proprietary format. The key point is that L3 data can be used to recall every placement, cancellation and transaction that transpired in the market. Knowing the content of the queue at every level is an exceptionally powerful tool when running simulations of LOBs. In general, however, it is uncommon, even for major institutions, to have access to data at this level of detail and it is also *very* expensive. In most cases, we rely on L2 data with some additional, partial L3 data such as transactions to perform our reconstructions [44]. Unfortunately, this is a lossy process when combined with shadowing (Section 4.5.2), so one must balance the trade-off between cost and quality.

4.3 RECONSTRUCTION

It goes without saying that any kind of research on LOBs requires a simulator. Limit order book reconstruction — often called market replay in the literature [178] — refers to the family of techniques whereby historical events occurring in an order book are reconstructed directly from data. This provides a platform for *factual reasoning* about the evolution of LOBs in the real world at the time the data was recorded. This type of simulation, however, comes with a number of computational challenges which increase as we progress from L1 data to L3 data.

In the simplest case — with L1 data — an implementation is trivial since we need only monitor two pairs: $(Z_t^+, \omega_t(Z_t^+))$ and $(Z_t^-, \omega_t(Z_t^-))$. These values can be stored on the stack and thus the rate of mutation is limited only by the clock rate of the available hardware. The most likely constraint is the speed at which the data itself can be read into memory. In the L2 setting, writing a fast order book simulator in most languages can also be done very efficiently since there are a fixed number of observable price levels. Using a stack-allocated array type one can ensure incredibly fast read and write to the level buffers. This, however, does not extend to the L3 case.

Level 3 Data Example

One key feature of LOBs is that most price levels are unoccupied at any given instant in time. As a result, it would be *highly* inefficient, if not entirely intractable, to represent the volume/order queue at every level using a contiguous array as most of the entries will be empty. While this would be a computationally effective approach, due to amortised constant lookup complexity, the memory requirements would grow unbounded. Even with truncation, data structures of this type are ill-suited to the problem and represent an extremely naïve approach for reconstruction and simulation. One alternative might be to represent the order book using a map/dictionary which encodes the sparsity of the LOB object. For example, in Rust, if prices are defined in ticks such that each value lies in $\mathbb{N} \cup \{0\}$, then one could use collections such as the `BTreeMap` or `HashMap` to store the order queues. While this would certainly yield slower lookup times, it bounds the memory requirements such that they only scale with the number of occupied levels in the book. But we can do better.

When designing LOB implementations, one must always keep in mind the operations one will be performing on said data. Up until this point we have only been considering lookup with random access, but processes like walking the book require sequential access on two levels: price-by-price in the book and order-by-order in each queue; this follows from standard price-time priority. These kinds of problems have been studied in computer science for many years and there is, arguably, no *unique* optimal solution. We propose the following structure for a single side of the book — illustrated in Rust — that balances memory and computational complexity across all operations:

```rust
pub struct Book<I, P, V> {
    pub orders: HashMap<I, Order<P, V>>,
    pub queues: BTreeMap<P, PtrQueue<Order<P, V>>,
}
```

Here, `I` denotes an identifier type, `P` the price type, and `V` the volume type; note that both `I` and `P` must implement `std::hash::Hash`. Each order is represented by the generic type `Order<P, V>` and is stored in a dictionary with it's unique identifier as the key. This allows for rapid lookup based on some hashable identifier type. The `PtrQueue<_>` at each occupied price level then maintains the time-ordering over orders in the queue, where each entry is simply a pointer to the concrete order instance. A complete LOB model would consist of two instances of the `Book<_, _, _>` type; one for asks and one for bids. Of course, this is by no means the only possible choice, and indeed each language will have different functionality that lends itself to specialised implementations.

4.4 INDICATORS

Equipped with a dataset, and a simulator to reconstruct the book, we now also have the ability to retrospectively compute market indicators as they occurred in the past. These form the backbone of many algorithmic trading strategies, especially those that operate on the intra-day level. Historical replay of these values, therefore, is crucial in using simulators to assess performance or train machine learning models that use indicators as features. Below, we introduce a set of important indicators that are used throughout this work. Note, this exposition barely scratches the surface on the vast space of choices and we refer the interested reader to the work of, e.g. Laruelle and Lehalle [97].

4.4.1 *Price*

Many market indicators are derived directly from observations of the market price. The value of an asset, as computed through the “hive-mind” that is an LOB, tells us a great deal of information that may aid in decision-making.

MARKET MID-PRICE The current market mid-price, $\widetilde{Z}_t$, defined in Equation 3.7, can be used as an estimate of the latent price of the asset, Z_t, which cannot be observed directly. This tells us about the valuation of the asset based on the aggregation of all participants in the market. Assuming this is relatively efficient, then this price should be formed based on all the information available up to time t; i.e. $\mathbb{E}\left[\widetilde{Z}_t \,\middle|\, \mathcal{F}_t\right] \approx \mathbb{E}\left[Z_t \mid \mathcal{F}_t\right]$, where $\mathcal{F}_t$ is the natural filtration. If this doesn't match the true value, then one can trivially derive a strategy that buys when the asset is undervalued by the market, and sells when it is overpriced.

MID-PRICE MOVE The change in the market mid-price since the last period is defined, in discrete-time, as $\Delta\widetilde{Z}_t = \widetilde{Z}_{t+1} - \widetilde{Z}_t$. Such a price movement is associated with a change in the distribution of orders in the book due to cancellations and/or transactions. The mid-price move may thus be associated with further price movements in the short-term, or trading momentum. One may choose to normalise this value using the tick size for a more market-independent measurement, or consider higher-order lags such as $\widetilde{Z}_{t+10} - \widetilde{Z}_t$.

MARKET SPREAD The market (bid-ask) spread, D_t, is defined as the difference between the best bid and ask prices; see Equation 3.8. It is often used as a measure of liquidity, indicating 'how highly the market values the immediacy and certainty associated with market orders versus the waiting and uncertainty associated with limit orders' [70]. It also provides an estimate of the profit available to a market maker quoting at the top of the books or, equivalently, the immediate cost of consuming liquidity.

VOLATILITY Volatility is a measure of the dispersion of changes in the price of a stock, often measured naïvely using the standard deviation of historical price changes. It is an important consideration for trading strategies since it helps determine the execution probability of limit orders; higher volatilities have higher associated execution probability and vice-versa [86]. Further, 'since volume and volatility are highly correlated and display strong time series persistence, any variable correlated with volatility will, inevitably, possess non-trivial forecast power for future volatility. This is true for bid-ask spreads, the quote intensity, the transaction count, the (normalized) trading volume. . . ' [8]. This ability to predict future volatility and thus the likelihood of execution at some point in the future is an effective tool for market makers.

RELATIVE STRENGTH INDEX The relative strength index is a commonly used counter-trend (predicts over-extensions in a price series) technical indicator. It is computed as the ratio between the average upward price movements and the average downward movements, scaled to the range $[-1, 1]$; note any averaging technique can be used in lieu of the arithmetic mean. Though there is mixed evidence on its predictive power when used alone [184], it has been suggested that it may be predictive when combined with other trading signals.

4.4.2 *Volume*

Another large class of metrics used to measure the state of the market — i.e. state features used to render the domain Markovian — can also be computed from the volume distribution in the LOB. This is a highly intuitive concept since most people would assume asymmetry in a market indicates some level of asymmetry in the intents of it's participants.

ORDER BOOK IMBALANCE The order book imbalance — also known as *volume imbalance* — is the normalised ratio of the difference in volume between the ask and bid books,

$$I_t \doteq \frac{\int_0^\infty \omega_t^+(z) - \omega_t^-(z)\,dz}{\int_0^\infty \omega_t^+(z) + \omega_t^-(z)\,dz} \in [-1, 1]. \tag{4.1}$$

A significant amount of research has been done into the predictive power of the book imbalance [36, 99]. For example, Gould and Bonart [69] found that it has a strong statistical relationship with price movement and is an especially good predictor in large-tick stocks. It is surely one of the most ubiquitous — and arguably well-justified — technical indicators for predicting short-term directionality; i.e. it is well correlated with the state of bull versus bear market.

ORDER FLOW IMBALANCE The order flow imbalance — known colloquially as the *signed volume* — is defined as the normalised difference between the number of arriving buy and sell market orders at a given instant in time. Concretely, the signed volume at time $t+1$ is defined as the fraction

$$\frac{\sum_{o \in \mathcal{M}_t} \omega_o}{\sum_{o \in \mathcal{M}_t} |\omega_o|} \in [-1, 1], \tag{4.2}$$

where $\mathcal{M}_t$ is defined as the set of market orders (both ask and bid) that were submit at time t; see Definition 9. This quantity has been used successfully in existing literature [119] to improve the performance of execution algorithms. Research into market microstructure [104, 123] has also shown that it is associated with the behaviour of uninformed traders, and may thus hold predictive power over trade direction in the short-term.

4.4.3 *Hybrid*

Another important set of indicators lie in-between the two discussed in the previous sections by including information derived both from prices and volumes. These can be very effective at reducing the dimensionality of the full LOB state.

MARKET MICRO-PRICE The market *micro*-price is an extension of the *mid*-price that weights the best prices according to the total volume on either side of the LOB. This quantity, one can show, may be expressed as a translation of the mid-price:

$$\widetilde{Z}_t^{\dagger} \doteq \widetilde{Z}_t + \frac{I_t \, D_t}{2}, \tag{4.3}$$

where I_t is defined in Equation 4.1, and D_t in Equation 3.8. Observe that $\widetilde{Z}_t^{\dagger}$ is bounded between $\left[Z_t^-, Z_t^+\right]$ since $I_t \in [-1, 1]$. At these two extremes, the price is skewed in favour of the sell and buy sides, respectively. Of course, this definition does not add any new information. If we already know $\widetilde{Z}_t$, I_t and D_t. However, it can be very convenient to work directly with $\widetilde{Z}_t^{\dagger}$ when defining a (relative) market making strategy that adapts to the state of the LOB.

MICRO-PRICE MOVE Finally, much as in Section 4.4.1, we can define a micro-price *move* as the change in $\widetilde{Z}_t^{\dagger}$ over some finite time horizon. In this case, a change in value from t to t' may indicate a change in the mid-price itself, or a change in the LOB volume distribution. It follows that a non-zero value of $\Delta\widetilde{Z}_t^{\dagger}$ can predict a change in intention of the participants in aggregate and thus the short-term momentum of the market.

4.5 COUNTERFACTUALS

Reconstruction of a limit order book (LOB) as described in Section 4.3 is mostly a computational problem. But, as of yet, we have not addressed all the challenges that arise when data-driven simulations of this type are used for training strategies using RL. For this, we require a number of key ingredients: the state and observation spaces, the action space, and the transition dynamics. The ability to retrospectively compute market microstructure indicators, as discussed in the previous section, answers the first part of the puzzle. The action space, too, can be defined easily. The real challenge lies in specifying the transition dynamics under counterfactual scenarios:

(i) How do we account for the market impact due to walking the book?

(ii) How do we simulate the execution of artificial LOs?

Put simply, how would the market have behaved if some agent (our agent) had placed a given order in the market? How do we infer the transition dynamics from this proposed behavioural approximation? And, finally, how do ensure that the discrepancy between the proposed and true responses are minimised? We argue that the quality of the solutions that are possible depend strongly on the data you have available.

4.5.1 *Market Impact*

As covered in Chapter 3, market impact comes in two flavours: temporary (Equation 3.4) and permanent (Equation 3.3). The latter, in particular, is a non-trivial reconstruction challenge because the *simulated market will diverge* from the historical data following an aggressive action. In this case, the historical transitions no longer reflect the same market behaviour; the measures have changed. We are presented with two choices: (i) attempt to interpolate the factual and counterfactual scenarios; or (ii) assume that the agent's artificial order is negligible compared with the liquidity in the book.

The majority of academic work takes the second approach which, for blue chip stocks and other actively traded assets, is not unreasonable. In effect, we assume that $\frac{dZ_t}{d\nu_t} = 0$, and treat the temporary impact as implicit. The total impact, $\frac{dX_t}{d\Omega_t}$, is thus computed by "imagining" the cost of walking the book without actually mutating the book. See, for example, the simulated LOB illustrated in Figure 3.2 for which the cost of executing an MO (i.e. temporary market impact) is depicted in Figure 3.3.

This, of course, is a significant limitation of data-driven approaches. Indeed, this problem has been cited explicitly in work by Vyetrenko and Xu [179] who proposed an annealing-based method for rejoining the bifurcated processes. In short, their approach worked by replacing the simulated prices with the true prices (post market order) whenever a subsequent aggressive order by the main agent or other participants moved the market back in the opposite direction. This method, while certainly not perfect, provides a more realistic simulation against which to train RL-based agents. However, as of yet, very little work has been done to address this issue due to the following challenge: how does one assess the quality of a simulation when there is no ground truth available? This issue has also been acknowledged in work by Christensen, Turner, Hill, and Godsill [44]. Any future work in this area — and the techniques developed for analysis — would therefore be of great value to both researchers and practitioners.

4.5.2 *Queues*

The second key challenge arises when we only have *aggregate information* about the limit orders (LOs) in a book. In this case, we do not know how orders are distributed in each queue, nor how changes manifest as transactions and cancellations. Consider Figure 3.2, for example. If the volume at price level 100 increased to 50, we can be certain that some new orders were placed, but we cannot know if there were also cancellations or transactions. Indeed, even if no change in volume was observed, one cannot know with complete confidence whether this was due to an equal balance of flow from placements and cancellations/transactions, or whether nothing occurred at all. This problem — as commonly experienced in regression analysis — amounts to a lack of uniqueness in the space of solutions; i.e. the problem is ill-conditioned.

This becomes a serious issue when simulating the progression of an artificial order since the simulated execution process is ambiguous. With access to L3 data,

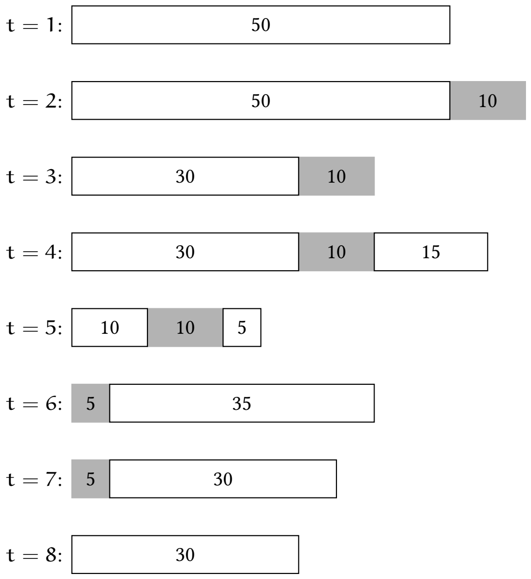


Figure 4.1: Illustration of queue approximation for a simulated order of size 10 under a volume-weighted cancellation scheme. At each stage, some combination of placement/cancellation requests are parsed and the subsequent queue estimate shown.

this would of course not be an issue as one can trivially reconstruct, with complete certainty, the changes in each queue. Less granular datasets such as L2 and below, however, do not contain sufficient information to distinguish between events. Instead, the only means by which to reduce uncertainty is with access to transaction data; this is not uncommon. With this information, one can disambiguate decreases in volume that occurred due to transactions from those that derived from cancellations. Of these two, transactions are especially simple to simulate as this volume change can only come from the front of the queue. Handling the cancellation of limit orders, on the other hand, is still not perfectly reconstructible as we do not know *where* in the queue these occurred.

To tackle this, we introduce a process we refer to as *shadowing* in which an RL agent's order is played out alongside a simulation in a post hoc fashion. In this case, we assume that cancellations are distributed uniformly throughout the queue, which we argue is reasonable. This means that cancellations are computed in a volume-weighted fashion before and after the location of the simulated order. Consider, for example, Figure 4.1, which depicts one possible sequence of events with an artificial order of 10 units located at the back of the queue (at time $t = 2$). A transaction of 20 units occurs, leaving only 30 units ahead of the agent's order, followed by an addition of 15 units from the historical data. At $t = 5$ we simulate the cancellation of 30 units of volume, which is split into reductions of 20 and 10. The remaining changes are all intuitive, ending with the artificial order being fully executed.

It is important to note that while this approach appears fair, we have no way of validating it's accuracy. As noted by Christensen, Turner, Hill, and Godsill [44], without access to complete L3 data, there is no way of assessing the bias of our estimators. What's more, we have no means of accounting for any change in market behaviour that may have occurred if the agent's order had actually been present at the time. As with market impact, we must simply ensure that the artificial order sizes are negligible compared with the distribution of volume in the book.

5 RL ∩ DATA-DRIVEN TRADING

5.1 OUTLINE

Traditional approaches to market making in mathematical finance are based on stochastic optimal control. A researcher proposes a model of market dynamics — such as those pioneered by Ho and Stoll [84] or Glosten and Milgrom [66] — and derives a set of controls that maximise/minimise a chosen objective/utility function. This is typically achieved using some variant of the Hamilton-Jacobi-Bellman equation which provides the necessary and sufficient condition for optimality. The problem is that very few realistic models of the market are soluble, and those that are require highly advanced mathematics, such as the notion of viscosity solutions. Moreover, the Hamilton-Jacobi-Bellmanon is a non-linear, partial differential equation. While these types of results offer deep insight into the structure of a problem, and therefore hold academic value in and of themselves, they are of limited practical value if the assumptions do not reflect reality, or if they are used inappropriately.

We need only look at the 2007-2008 Financial Crisis to confirm this [102].

One of biggest advantages of electronic markets today is the incredible availability of high-quality historical data; albeit very expensive. As seen in Chapter 4, this can be used to reconstruct limit order book markets without introducing structural bias, and with minimal assumptions on the underlying transition dynamics. These can then be used to evaluate strategies, explore counterfactual scenarios and even train RL-based trading agents directly from simulation. They remove the need to specify and calibrate a model, and are much more flexible in their capacity for handling exotic control/action-spaces. While these simulation-based approaches do, of course, have their own limitations (as discussed in Chapter 4), they represent a powerful and complementary tool that can be used alongside prevailing methods in stochastic optimal control.

In the literature, the term "replay" is often used in place of "reconstruction".

Contributions

The main contribution of this chapter is to evaluate value-based control methods for deriving market making strategies directly from an LOB reconstruction, with a focus on the discrete-action setting; more complex approaches are reserved for Part III. We identify eligibility traces as a solution to the unresolved issues previously associated with reward attribution and noisy environment dynamics [40]. We then design a novel reward function and state representation that are shown to be key factors in the success of the combined approach. An outline of the steps taken to develop this framework is given below:

(i) We address concerns raised in past work about the efficacy of one-step temporal-difference (TD) learning, corroborating their results but demonstrating that **eligibility traces** are a simple and effective solution.

(ii) We evaluate a wide range of new and old TD learning algorithms, highlighting the discrepancies in performance and providing qualitative justification for the observed results.

(iii) We show that a simple risk-neutral reward function does not lead to the best performance and regularly induces instability during learning. We propose a solution in the form of an **asymmetrically damped** reward function which improves learning stability, and produces higher and more consistent returns.

(iv) We explore three different value function representations and propose a **factored representation**, which is shown to yield competitive performance and more stable learning.

(v) We present a **consolidation** of the best results from the above, showing that it produces the best risk-adjusted out-of-sample performance compared to a set of simple benchmarks, a basic RL agent, and a recent online learning approach [3]. Moreover, it is argued that the performance of the strategies derived using our proposed method are competitive enough to represent a viable approach for use in practice.

5.2 RELATED WORK

Market making has been studied across a number of disciplines, including economics, finance, artificial intelligence (AI), and machine learning. A classic approach in the finance literature is to treat market making as a problem of *stochastic optimal control*. Here, a model for order arrivals and executions is developed and then control algorithms for the resulting dynamics are designed [11, 37, 38, 75, 80, 84]. Recent results in this line of research have studied price impact, adverse selection and predictability [2], and augmented the problem characteristics with risk measures and inventory constraints [34, 78].

Another prominent approach to studying market making and limit order book markets has been that of *zero-intelligence (ZI) agents*. The study of ZI agents has spanned economics, finance and AI. These agents do not "observe, remember, or learn", but can, for example, adhere to inventory constraints [67]. Newer, more intelligent variants, now even incorporate learning mechanisms [45, 180]. Here, agents are typically evaluated in simulated markets without using real market data.

A significant body of literature, in particular in AI, has studied the market making problem for *prediction markets* [29, 121, 122]. In this setting, the agent's main goal is to elicit information from informed participants in the market. While later studies have addressed profitability, the problem setup remains quite distinct from the financial one considered here.

Reinforcement learning has also been applied to other financial trading problems [114, 145, 152], including optimal execution [119] (a topic we shall cover in more detail in Section 6.3) and foreign exchange trading [52]. The first case of applying RL to market making [40] focused on the impact of noise (due to uninformed traders) on the agent's quoting behaviour and showed that RL successfully converges on the expected strategies for a number of controlled environments. They did not, however, capture the challenges associated with explicitly handling order placement and cancellation, nor the complexities of using continuous state variables. Moreover, [40] found that temporal-difference RL struggled in their setting, a finding echoed in [151]. [40] attributed this to partial observability and excessive noise in the problem domain, despite the relative simplicity of their market simulation. In follow-up work, [148] used importance sampling as a solution to the problems observed with off-policy learning. In contrast, we find temporal-difference RL to be effective for the market making problem, provided that we use eligibility traces and carefully design our function approximator and reward function.

One of the most recent related works is [3], which uses an *online learning* approach to develop a market making agent. They prove nice theoretical results for a stylized model, and empirically evaluate their agents under strong assumptions on executions. For example, they assume that the market has sufficient liquidity to execute market orders entirely at the posted price with no slippage. We use this approach as one of the benchmarks for our empirical evaluation and address the impact of trading in a more realistic environment.

5.3 PROBLEM SPECIFICATION

Consider a canonical market making problem in which an MM agent trades directly over an LOB. There are two assets: (i) a riskless asset, cash, whose value does not change over time (our numéraire); and (ii) a risky asset whose price, Z_t, evolves stochastically and exogenously to the market. This price defines the fundamental value of the risky asset and is treated as a latent variable of the model; i.e. it is not known exactly by any participant. Noisy estimates of this value, however, can be computed using various estimators based on the aggregation of orders in the LOB; see Section 3.3.

At each time $t \in \mathbb{N} \cup \{0\}$, the MM sends requests to the market to place new LOs, possibly replacing those outstanding. It may alternatively choose to place a MO in order to manage it's inventory by consuming liquidity from the book. Simultaneously, the state of the LOB innovates due to new information entering the market and the corresponding behaviour of all the participants. This gives rise to a combination of profits for the MM, derived from transactions and from speculation on accumulated inventory. Note that we make the relatively strong assumption that transactions fees are negligible. These would typically take the form of a principal cost on any trade based on volume, as enforced by the exchange. However, it is often the case that large institutional market makers also receive rebates on these fees under contractual agreement that they never leave the market. It is precisely this setting that we study here and thus motivates our choice to ignore fees.

This problem specification can be formally expressed in the language of RL as a partially-observable MDP. For this, we first define the state at time t as the tuple $(\Omega_t, \mathcal{B}_t, Z_t, \ldots)$, where the ellipsis represents an arbitrary number of unknown properties governing the market dynamics. Given a state s_t, the MM observes the values $(\Omega_t, \mathcal{B}_t)$ with probability 1, and chooses an action to take from the set $\mathcal{A} \doteq \{0, \ldots, n\}$ according to it's policy. The transition probabilities are then dictated by the data, historical simulation and action a_t.

5.3.1 *Desiderata*

As outlined in Section 3.4, the primary quantities used to assess the performance of a trading strategy are the first and second moments of Υ_T. However, in this chapter, we experiment with a basket of securities which presents a scaling challenge when trying to compare results. To deal with this, we introduce a *normalised daily PnL* that measures a strategy's ability to capture the market spread. This metric is defined on a daily basis as the total portfolio value divided by the average market spread which normalises the profit across different markets: $\mathbb{E}\left[\Upsilon_T / \mathbb{E}_t\left[D_t\right]\right]$. This equates to the number of market spreads that would need to be captured in order to obtain a given profit margin.

As a secondary objective, market makers should attempt to maintain near-zero net inventories. This is to avoid exposure to risk associated with unanticipated, adverse price movements. To measure how well our agents achieve this, the *mean absolute position* (MAP) held by the agent is quoted as well. High values for this metric may indicate that the agent has taken a speculative approach to trading. On the other hand, small values could suggest that the agent relies less on future changes in market value to derive it's earnings. A risk-sensitive agent would do the latter and that is what we intend to derive through careful reward construction.

Table 5.1: The security tickers comprising the full dataset with their associated company name and sector.

Ticker	Company name	Sector
CRDI.MI	UniCredit SpA	Finance
GASI.MI	Assicurazioni Generali SpA	Insurance
GSK.L	GlaxoSmithKline PLC	Pharmaceuticals
HSBA.L	HSBC Holdings PLC	Finance
ING.AS	ING Group NV	Finance
LGEN.L	Legal & General Group PLC	Finance
NOK1V.HE	Nokia Corp	Technology
SAN.MC	Banco Santander SA	Finance
VOD.L	Vodafone Group PLC	Technology

Table 5.2: Reference of stock exchanges indexed by the ticker suffix.

Suffix	Venue
AS	Amsterdam SE
HE	Helsinki SE
L	London SE
MC	Madrid SE
MI	Milan SE

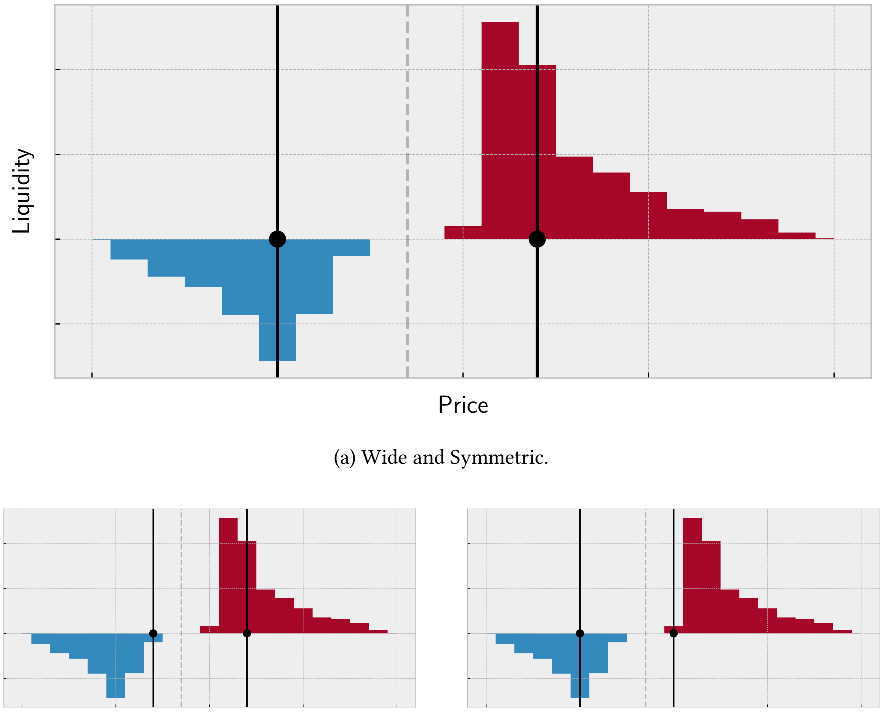
5.3.2 *Simulation*

The market environment itself was implemented as an event-by-event reconstruction using the methods outlined in Chapter 4. That is, we assumed negligible permanent market impact, and use shadowing to simulate the execution of artificial orders generated by the agent; see Section 4.5. The code was written in C++ and is publicly accessible at https://github.com/tspooner/rl_markets under the BSD 3-Clause License. This includes efficient implementations of LOBs, reinforcement learning (RL) algorithms and synchronised data streamers.

The dataset used to run the simulation comprised 10 assets traded over 5 venues and 4 different sectors; see Table 5.1. The venue for each ticker is given by the suffix, such as AS, each of which is translated in Table 5.2. While all 10 securities were traded on major exchanges, the liquidity of each varied greatly during the 8 months (January — August 2010) of recorded data. In each experiment, the market depth and transaction-level data was split into disjoint training, evaluation and testing sets, where all of the testing data occurs chronologically later than the evaluation data, and the same for the evaluation and training data. As per convention, the latter was used to actually train each agent, the evaluation set was used to measure the performance of the strategy during any hyper-parameter tuning, and the final set for comparing algorithms.

Table 5.3: Default parameters as used by the learning algorithm and the underlying trading strategy.

	Value
Training episodes	1000 days
Training sample size	~ 120 days
Testing sample size	40 days
Memory size	10^7
Number of tilings (M)	32
Weights for linear combination of tile codings [agent, market, full] (c_i)	$(0.6, 0.1, 0.3)$
Learning rate (α)	0.001
Step-size [R-learning] (β)	0.005
Discount factor (γ)	0.97
Trace parameter (λ)	0.96
Exploration rate (ε)	0.7
$\varepsilon_{\text{Floor}}$	0.0001
ε_T	1000
Order size (ω)	1000
Min inventory ($\min \text{Inv}$)	-10000
Max inventory ($\max \text{Inv}$)	10000



(a) Wide and Symmetric.

(b) Tight and Asymmetric (pro-bid).

(c) Tight and Asymmetric (pro-ask).

Figure 5.1: Illustration of a spread-skew market making strategy in an LOB.

HYPER-PARAMETERS The hyper-parameters used in the experiments to follow we held fixed across all variants. These values are summarised in Table 5.3 and should be assumed to hold throughout this chapter, unless otherwise stated.

5.4 THE STRATEGY

As we know from Chapter 3, market makers are primarily concerned with two things: *pricing* and *sizing*. That is, where should the MM place its orders, and what quantity and distribution of its desired volume should be placed amongst these orders. All other considerations, such as inventory management and liquidity provision are simply derivatives of these core concepts. While simple to state, doing this effectively is non-trivial and is the subject of a great deal of research.

In this chapter, we consider a standard class of ladder-based market making strategies that use the principle of *spreading* and *skewing* [38]. To keep things simple, we only consider strategies that place limit orders (LOs) at one ask price and one bid price, and with the same fixed quantity on each side. This collapses the problem into one of pricing alone, since we have frozen one dimension of the problem. Now, for pricing, the idea behind this type of behaviour is that the MM has some notion of what the fundamental value, Z_t, is at any given time. Orders are placed about this value with some width and skew in accordance with the agent's need for immediacy and/or bias in execution, respectively. A high-level illustration of this strategy is given in Figure 5.1.

To define this more formally, let $\widehat{Z}_t$ denote the point estimate of the fundamental value Z_t as held by the MM. This can be thought of as the *reference price* used by the MM, with all of it's orders are placed relative to this value. The estimate itself is typically given by the market mid-price (Equation 3.7), the micro-price (Equation 4.3) or a smoothed variant thereof; note this list is by no means exhaustive. The MM then selects two controls, $\delta_t^\pm \in \mathbb{R}^2$, which define the price offsets away from the reference $\widehat{Z}_t$. The difference between these two values, $\delta_t^+ - \delta_t^-$, is defined as the *quoted spread*, a quantity analogous to the market spread, D_t. This is not constrained by definition, but it is sensible to assume that $\delta_t^+ - \delta_t^- > 0$ in most cases.

At each time step, the MM generates prices, $\delta_t^\pm$, which are translated into limit orders (LOs) that are sent to the LOB:

$$o_t^+ = \begin{cases} \left(t, \omega, \widehat{Z}_t + \delta_t^+\right) & \text{if } \Omega_t - \omega \geqslant \underline{\Omega} \wedge \left|\Delta\delta_{t-1}^+ - \Delta\widehat{Z}_{t-1}\right| > 0, \\ (\infty, 0, t) & \text{otherwise,} \end{cases} \tag{5.1}$$

and

$$o_t^- = \begin{cases} \left(t, -\omega, \widehat{Z}_t - \delta_t^-\right) & \text{if } \Omega_t + \omega \leqslant \overline{\Omega} \wedge \left|\Delta\delta_{t-1}^- + \Delta\widehat{Z}_{t-1}\right| > 0, \\ (0, 0, t) & \text{otherwise.} \end{cases} \tag{5.2}$$

These define the ask and bid orders, respectively, and the two conditions under which they are non-trivial. The first requires that an execution not move the inventory of the MM beyond the upper/lower limit. The latter requires that the new order be at a different price compared to the previous time step. Otherwise, the order is empty and will be ignored. In the case that a new order is created, all of the outstanding orders created by the MM are cancelled and removed from the book. These constraints and cancellation mechanism together ensure that the MM has only a single active order in each side of the book at any one time; and that we do not replace orders at an already occupied price.

The MM is also allowed to submit an MO in order to aggressively consume liquidity. This of course incurs a cost from walking the book (see Section 3.3.1), but allows the agent to quickly reduce inventory exposure when necessary. For this, we introduce a third control variable ω_t^m which denotes the volume of the market order at any given time. As with the LOs, this control is translated into a tuple

$$o_t^m = \begin{cases} (t, \omega_t^m, 0) & \text{for } \omega_t^m \geqslant 0, \\ (t, \omega_t^m, \infty) & \text{for } \omega_t^m < 0. \end{cases} \tag{5.3}$$

When $\omega_t^m = 0$ the order has no effect, but for any value $|\omega_t^m| > 0$ the agent will consume volume from the book.

In summary, the strategy of the MM is specified at each time step by:

δ_t^+ The price offset of the ask LO.

δ_t^- The price offset of the bid LO.

ω_t^m The volume of the MO.

5.4.1 *Discrete Encoding*

Now, in order to optimise this strategy using value-based RL, we must engineer a discrete encoding of the three control variables, $\delta_t^\pm$ and ω_t^m. This poses a challenge since there is a trade-off between representational capacity and computational efficiency. If the agent has only a small number of actions (i.e. $|\mathcal{A}|$ is small), then learning will be faster, but the space of solutions may be too restrictive to be effective. On the other hand, having a larger number of actions incurs a computational cost due to exploration and the curse of dimensionality, but a better solution is more likely to exist within the space of representable policies.

Table 5.4: Discrete encoding of the LO/MO MM strategy.

Action ID	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Ask (δ_t^-/β_t)	1	2	3	4	5	1	3	2	5
Bid (δ_t^+/β_t)	1	2	3	4	5	3	1	5	2
Action 9	MO with $\omega_t^m = \Omega_t$								

Table 5.4 outlines our construction for $\mathcal{A}$. It includes the ability to quote *wide or tight* with a *symmetric or asymmetric skew*, and the option place an MO to clear inventory. Note that the controls $\delta_t^\pm$ have been normalised by a time-dependent scale factor, β_t, which is given by a simple n-period moving average of the market spread:

$$\beta_t \doteq \frac{1}{n} \sum_{i=0}^{n-1} D_{t-i}, \quad \forall t \geqslant n-1, \tag{5.4}$$

where D_t is defined in Equation 3.8. This allows us to use a single generic action-space for all assets in the dataset without the need to explicitly recalibrate prices in each market. To understand why this normalisation is important, consider Figure 5.1. While this market is liquid and has a tight spread, other, more illiquid markets, may have a book where the volume is fragmented and thus has a spread of multiple ticks. Since we do not know a priori what the market liquidity will be, we must either: define a large enough action-space to cover $\delta^\pm$ at many scales; or use an adaptive encoding as above.

5.4.2 *Trading Clocks*

In addition to the core strategy constructions, it is important to address time in the model: what it is and how it is defined. The means by which one aggregates interactions in an order book, and thus what constitutes an actionable event to a trader — i.e. what $t \mapsto t+1$ actually means in the real world — is a research question in and of itself. Indeed, the root of this question can be traced back to the 1960s with the groundbreaking work of Mandelbrot and Taylor [107]. Traditionally, the standard "wall clock" is used as an appropriate metric: changes in the book are grouped into fixed length intervals (days, minutes, seconds, . . .). Updating of beliefs and strategic decision-making is done chronologically in accordance with the sum differences across these intervening periods. But this is not the only choice available.

If, say, a quartz clock, does this mean that "rational" behaviour is defined by the vibrations of crystals?

Recent work by Easley, Prado, and O'Hara [55] suggests that, nowadays, many high-frequency traders instead operate on "intrinsic time," such as the volume clock. In this setting, a trading interval is partitioned into bins of equal volume throughput. It can be shown that that this representation has convenient statistical properties and is more faithful to the fact that LOBs are, by definition, discretely evolving systems. Naturally, this principle could be extended to define a clock such that the required stylised facts are recovered. Events in the book can, and should, be grouped based on whatever a trader believes is most appropriate for their strategy.

Remark. *In some sense, this notion of clocks is equivalent to simple termination conditions in Semi-MDPs [134, 166]. We have implicitly assumed the existence of an underlying continuous-time process that drives the latent state; i.e. Z_t and the exogenous properties of the universe. The actual actions taken by the MM — now options — are then distributed heterogeneously in "real time". We have substituted the "true" problem* *with one that is more amenable to learning. How similar the optimal solutions are in this new setting to the original problem, however, is unclear.*

We take inspiration from existing approaches on event-based clocks and define time based on innovations in latent space. This is a natural reflection of the core modelling choice to represent the problem as a partially-observable MDP. It means that each actionable time point — *from the perspective of the MM* — occurs only when a change in the underlying state of the market yields a measurable change in some subset of elements of the observation. This could be a change in price, volume or in the arrangement of orders in the book. Here, *we require that the market mid-price (Equation 3.7) changes (in either direction and by any non-zero amount) before the agent may act again.* This helps reduce the number of instances where the MM reacts to random fluctuations in the market that do not correspond to meaningful transitions in latent space. While this is a simple choice, we found it to be effective. It remains an open question as to what the "most effective" clock is, if indeed there even exists a single, unilaterally optimal model.

5.5 BENCHMARKS

When it comes to measuring performance in trading there are many schools of thought, but very few absolutes; see Section 3.4. For example, claiming that a strategy earns £1000 a day, or achieves a Sharpe ratio of 1.2 is meaningless on its own. Without significant domain knowledge, the only way to make sense of these statistics is in relative terms. For this, we require benchmarks, and in the following sections we introduce three variants of the trading strategy with increasing levels of sophistication that will be used to ground the results presented later on. The last of these benchmarks is based on a state-of-the-art algorithm from the online learning literature.

Domain experts have their own internalised benchmark.

5.5.1 *Randomised Pricing with Clearing*

The first benchmark is based on a simple randomised policy which uniformly mixes between all 10 actions in Table 5.4 at each time step, forcibly choosing action 9 when the inventory reaches the upper/lower limit. Clearing inventory like this helps to reduce exposure to adverse selection in periods of strongly biased volume flow, but it is clear that randomly selecting prices at which to quote is not an effective strategy. This is reflected in Table 5.5. Note that the low maximum absolute position observed is merely an artefact of randomisation and is not a sign of strategic inventory management. Because the orders at each time step are replaced with high probability, it is very likely to cancel and replace its LOs and thus occupy the back of queues for the majority of time steps, thereby decreasing the likelihood of execution.

Remark. *Agents could manipulate a simulator and avoid trading altogether by randomising their actions and thus avoiding execution. It is important to identify these pathologies and engineer the simulator or strategy to avoid these local, meta-optimal solutions. In this work, we took the approach of defining the agent's intrinsic time based on non-zero changes in the market mid-price. This helped to prevent the agent from changing its orders too often.*

5.5.2 *Fixed-Symmetric Pricing with Clearing*

The second set of benchmarks quote at fixed, symmetric distances from the reference price at all times: i.e. $\delta_t^+ = \delta_t^-$ at all times t. In effect, each variant is associated with a

Table 5.5: Performance attributes for a set of fixed and random benchmark instances of the strategy with boundary-based inventory clearing, evaluated on HSBA.L; ND-PnL refers to normalised daily profit and loss, and MAP refers to mean absolute position.

	ND-PnL $[10^4]$	**MAP [units]**
Randomised (Table 5.4)	-10.82 ± 5.63	135 ± 234
Fixed ($\delta^{\pm}/\beta = 1$)	-20.95 ± 17.52	3646 ± 2195
Fixed ($\delta^{\pm}/\beta = 2$)	2.97 ± 13.12	3373 ± 2181
Fixed ($\delta^{\pm}/\beta = 3$)	0.42 ± 9.62	2674 ± 1862
Fixed ($\delta^{\pm}/\beta = 4$)	1.85 ± 10.80	2580 ± 1820
Fixed ($\delta^{\pm}/\beta = 5$)	2.80 ± 10.30	2678 ± 1981

single action in the range $[0, 4]$ from Table 5.4. Market making strategies of this form are a special case/simplification of the ladder strategies studied by Chakraborty and Kearns [38]. While trivially simple to implement, these naturally account for liquidity in the market by adapting prices to changes in the bid-ask spread, depending on the choice of the scaling factor β_t. In other words, there is some level of intelligence to the strategy.

A sample of performance for 5 instances is given in Table 5.5 for HSBA.L; the full set of results are omitted for brevity since agent performance was consistent across all securities. Fixed strategies with $\delta^{\pm}/\beta > 1$ were found to be *just* profitable on average, with decreasing MAP as $\delta^{\pm}$ was increased. This already represents an improvement over the randomised policy. The variance on both ND-PnL and MAP, however, is significant in terms of scale compared to their mean values — i.e. risk-adjusted performance is very poor. This may be caused by a lack of inventory management, as indicated by the consistently high mean average positions; though this is offered only as a possible, qualitative explanation.

5.5.3 *Online Pricing*

The final pair of benchmarks are based on an adaptation of the spread-based strategies introduced by Abernethy and Kale [3]. These leverage online learning meta-algorithms to construct behaviours from sets of simple strategies that are not only adapted to the market, but also enjoy provably low regret. In this work, we focus on the market-making multiplicative weights (MMMW) variant which uses the multiplicative weights method [10] to pick, in each period, from a class of simple strategies parametrised by a minimum quoted spread. This set of simple strategies is defined as the nine unique LO actions in Table 5.4 — it thus includes both symmetric and asymmetric quotes. Rather than include the MO action as an independent "strategy", we opt for automatic clearing as in previous two benchmarks.

The performance of the MMMW benchmark is shown in Table 5.6 alongside the follow-the-leader variant that was also proposed by Abernethy and Kale [3]. The latter was included only for reference as the performance was consistently worse than the MMMW algorithm. These results are less favourable than those presented in the original paper which found the strategy to be profitable over all dates and securities considered. This could be attributed to the use of a less realistic market simulation that did not, for example, track the limit order book to the same level of precision considered here. Indeed, this may indicate that their results do not generalise to more realistic markets.

Table 5.6: Out-of-sample normalised daily PnL (ND-PnL) and mean absolute positions (MAP) of the follow-the-leader (FTL) benchmark strategy derived from [3].

	MMMW		**FTL**	
	ND-PnL [10^4]	MAP [units]	ND-PnL [10^4]	MAP [units]
CRDI.MI	-1.44 ± 22.78	7814 ± 1012	-14.93 ± 25.52	5705 ± 1565
GASI.MI	-1.86 ± 9.22	5743 ± 1333	-8.50 ± 24.00	6779 ± 1499
GSK.L	-3.36 ± 13.75	8181 ± 1041	-29.33 ± 95.39	8183 ± 1272
HSBA.L	1.66 ± 22.48	7330 ± 1059	-4.00 ± 35.84	8875 ± 683
ING.AS	-6.53 ± 41.85	7997 ± 1265	-27.53 ± 114.97	9206 ± 981
LGEN.L	-0.03 ± 11.42	5386 ± 1297	0.29 ± 12.45	5824 ± 1512
LSE.L	-2.54 ± 4.50	4684 ± 1507	-2.60 ± 4.49	4776 ± 1615
NOK1V.HE	-0.97 ± 8.20	5991 ± 1304	-3.47 ± 8.81	5662 ± 1533
SAN.MC	-2.53 ± 26.51	8865 ± 671	-8.80 ± 50.00	9273 ± 470
VOD.L	1.80 ± 22.83	7283 ± 1579	-1.72 ± 25.11	8031 ± 1610

Looking in more detail, this is not entirely surprising given the assumptions of their model. For example, their analysis relied in the assumption that market orders are executed perfectly at the mid-price rather than walk the book. While this does not detract from the validity of the approach, it does suggest that the derived regret/performance bounds are optimistic. Secondly, they enforce that unexecuted LOs be cancelled and re-placed at each period. In a simple model this can be done for mathematical convenience without affecting the results, but in a more realistic model such as ours, this has a significant impact. When simulating the full queue at each price, repeated cancellation orders will yield a very low chance of execution. This relates to our past remark on using intrinsic time to prevent the agent acting too often and never observing any transactions. Nevertheless, the results are included here as a benchmark since they still represent an important contribution to the field and thus also represent a valid comparator.

5.6 RISK-NEUTRAL BEHAVIOUR

With this section we begin our evaluation of reinforcement learning (RL) for optimising The Strategy, focussing only, for time being, on the risk-neutral domain. In this setting the agent's objective is to maximimise it's expected profit and loss — however that may be defined — with no additional constraints on behaviour, such as risk penalties. To do this, we first define a reward function which captures the change in the value of the agent's portfolio from t to $t+1$. There are two possible definitions one can consider.

The first approach is to define the reward directly as the change in cash, $r_t \doteq \Delta X_t$. If the agent removes it's LOs at $T-1$ and liquidates it's entire portfolio in the period $T-1 \mapsto T$, then this will exactly recover the mark-to-market portfolio value of the agent, Υ_T. Indeed, even without these two assumptions, if the liquidation occurs entirely at Z_{T-1} (i.e. if $\bar{Z}^m_{T-1} = Z_{T-1}$), then the terminal cash value again corresponds to Υ_T. To define this formally, observe that the expressions for the cash generated due to asks and bids, respectively, are given by

$$\Delta\upsilon_t^+ \doteq \underbrace{\left(\nu_t^+ - \max\{\widetilde{\omega}_t^m, 0\}\right)\left(Z_t + \delta_t^+\right)}_{\text{Ask LO}} + \underbrace{\max\{\widetilde{\omega}_t^m, 0\}\bar{Z}_t^m}_{\text{Ask MO}},$$

and

$$\Delta\upsilon_t^- \doteq \underbrace{\left(\nu_t^- + \min\{\widetilde{\omega}_t^m, 0\}\right)\left(Z_t - \delta_t^-\right)}_{\text{Bid LO}} + \underbrace{\min\{\widetilde{\omega}_t^m, 0\}\bar{Z}_t^m}_{\text{Bid MO}}.$$

This allows us to define a "cash flow" reward function by the value

$$\Delta X_t = \left(\nu_t^+ - \max\{\widetilde{\omega}_t^m, 0\}\right)\delta_t^+ + \left(\nu_t^- + \min\{\widetilde{\omega}_t^m, 0\}\right)\delta_t^- + \widetilde{\omega}_t^m\bar{Z}_t^m - \nu_t Z_t.$$

While this has a natural interpretation, it is important to note that the values of this expression scale with the prices, Z_t. This means that the rewards may be very large and may also suffer from high variability when there are many transactions. This is especially true when those transactions are skewed in favour of bid/ask across a single time step. This is an undesirable property since market makers, in general, rely on high volume throughput to generate revenue and are often caught on one side of the market for multi-step periods.

The alternative approach is to track the change in the mark-to-market portfolio value directly, such that $r_t \doteq \Delta\Upsilon_t$. Substituting $\Delta\upsilon_t^\pm$ into (3.1), it can be shown that

$\bar{\delta}_t^m$ equals $Z_t - \bar{Z}_t^m$ if $\widetilde{\omega}_t^m > 0$ and $\bar{Z}_t^m - Z_t$, otherwise.

$$\Delta\Upsilon_t = \underbrace{\left(\nu_t^+ - \max\{\widetilde{\omega}_t^m, 0\}\right)\delta_t^+ + \left(\nu_t^- + \min\{\widetilde{\omega}_t^m, 0\}\right)\delta_t^- + \widetilde{\omega}_t^m\bar{\delta}_t^m}_{\text{Spread PnL}} + \underbrace{\Omega_{t+1}\Delta Z_t}_{\text{Inventory PnL}},$$

where $\bar{\delta}_t^m$ is the volume-weighted average spread paid by the MO. In this case, we note that the reward, while exactly equivalent to the previous definition by time T, scales only with changes in price. This quantity is lower bounded by the tick size, but typically registers on the order of 1 to 2 ι. This is a significant reduction (at least for realistic scenarios) and enjoys a significantly reduced variance as a result. In other words, using $\Delta\Upsilon_t$ as a reward function defines a path of "least variation" to the terminal objective. We define this incremental mark-to-market reward function below.

Definition 14 (Incremental mark-to-market reward)**.** *The (risk-neutral) mark-to-market reward is expressed by the four terms:*

$$r_t \doteq r_t^+ + r_t^- + r_t^m + r_t^Z, \tag{5.5}$$

where

$$\begin{aligned} r_t^+ &\doteq \left(\nu_t^+ - \max\{\widetilde{\omega}_t^m, 0\}\right)\delta_t^+, \\ r_t^- &\doteq \left(\nu_t^- + \min\{\widetilde{\omega}_t^m, 0\}\right)\delta_t^-, \\ r_t^m &\doteq \widetilde{\omega}_t^m\bar{\delta}_t^m, \\ r_t^Z &\doteq \Omega_t\Delta Z_t. \end{aligned}$$

VARIANCE INEQUALITIES The difference in variance between the two approaches can be derived rigorously by identifying the conditions under which $\mathbb{V}[\Delta\Upsilon_t] \leqslant \mathbb{V}[\Delta X_t]$. First, note that these terms may be expressed as

$$\mathbb{V}[\Delta\Upsilon_t] = \mathbb{V}\left[r_t^+ + r_t^- + r_t^m\right] + \mathbb{V}[\nu_t Z_t] + \mathrm{Cov}\left[r_t^+ + r_t^- + r_t^m, \nu_t Z_t\right],$$

and

$$\mathbb{V}[\Delta X_t] = \mathbb{V}\left[r_t^+ + r_t^- + r_t^m\right] + \mathbb{V}[\Omega_t\Delta Z_t] + \mathrm{Cov}\left[r_t^+ + r_t^- + r_t^m, \Omega_t\Delta Z_t\right].$$

Subtracting these two equations, one arrives at the equivalence relation

$$\begin{aligned}\mathbb{V}[\Delta X_t] - \mathbb{V}[\Delta \Upsilon_t] = \mathbb{V}[\Omega_t \Delta Z_t] + \mathrm{Cov}\left[r_t^+ + r_t^- + r_t^m, \Omega_t \Delta Z_t\right] \\ - \mathbb{V}[\nu_t Z_t] - \mathrm{Cov}\left[r_t^+ + r_t^- + r_t^m, \nu_t Z_t\right].\end{aligned}$$

If we assume that the covariance terms are negligible, then it follows that $\mathbb{V}[\Delta \Upsilon_t] \leqslant \mathbb{V}[\Delta X_t] \iff \mathbb{V}[\nu_t Z_t] \geqslant \mathbb{V}[\Omega_t \Delta Z_t]$. This means that the difference in variance between $\Delta \Upsilon_t$ and ΔX_t depends, approximately, on the difference in variance between the mark-to-market cash flow term and inventory PnL, respectively. To see when this is satisfied we can analyse the scale of each term. In general, Ω_t and ν_t are on a similar scale, but Z_t may be many orders of magnitude larger than ι and thus ΔZ_t. It seems apparent that this inequality holds in all realistic scenarios since the variance on the cash flow term will almost always dominate. One must simply ensure that Ω_t does not grow too large.

APPROXIMATIONS In all the definitions above, the reward is computed with respect to the fundamental value Z_t, which is unknown to the simulator. It is latent not only by construction of the partially-observable MDP itself, but by the fact we are using data. To handle this, all instances of Z_t are replaced with an estimate $\widehat{Z}_t$. If $\Delta Z_t = \Delta \widehat{Z}_t$, then this transformation has no effect on the value $\Delta \Upsilon_t$ and the derived estimator, $\widehat{\Delta \Upsilon}_t$, is unbiased. Another subtle challenge with this configuration is that the objective function, $J(\pi) \doteq \mathbb{E}_{d_0,\pi}[\Upsilon_T]$, is itself non-stationary due to the explicit dependence on the time t and finite length of an episode. To address this, one can either include time in the observations, or use discounted returns as a proxy for the true objective. Here we opt for the latter and define the discounted value function

$$Q_\pi^\gamma(s, a) = \mathbb{E}_\pi[\Delta \Upsilon_t + \gamma Q_\pi^\gamma(s_{t+1}, a_{t+1}) \mid s_t = s, a_t = a] \,.$$

Since $\lim_{\gamma \to 1} Q_\pi^\gamma = Q_\pi$, an arbitrarily close approximation to the original objective can be recovered by increasing γ. The problem now reduces to one of policy evaluation, since acting greedily with respect to this quantity will yield and optimal policy.

Remark. *While one must be careful not to re-introduce non-stationarity for large γ, it is worth noting that our data-driven LOB simulation is unlikely to be perfectly stationary itself anyway. The challenge in tuning the discount factor is thus three-fold: reducing variance in the estimator; reducing bias between Q_π and Q_π^γ; and minimising the impact of non-stationary from the dependence on time and that inherent to the environment.*

5.6.1 *Credit Assignment*

Define the action-value estimator, $\widehat{Q}_{\boldsymbol{w}}(s, a)$, as in Equation 2.32 with basis functions given by a hashed tile coding over the space of values $(\Omega,\ \delta^+/\beta,\ \delta^-/\beta)$ and action a; the term β is as defined in Equation 5.4. Clearly the arguments to this function can be computed from the observations available to the market maker, $(\Omega, \mathcal{B})$; the dimensionality reduction avoids issues of scaling due to the (indefinite) size of the raw LOB object. More expressive representations are reserved for Section 5.8, and for now we focus only on the information directly pertaining to the internal state of the MM: it's inventory and active orders.

The construction $\widehat{Q}(s, a)$ was first learnt using the traditional one-step Q-Learning and SARSA but, consistent with the findings of past work [40], we were unable to obtain useful results in either case. Learning is simply too inefficient without making more effective use of the samples given the limited amount of data. The

Table 5.7: Mean and standard deviation on the normalised daily PnL (given in units of 10^4) for Q(λ) and SARSA(λ) using the incremental mark-to-market reward function and the algorithm parameters specified in Table 5.3.

	Q(λ)	**SARSA(λ)**
CRDI.MI	**8.14 ± 21.75**	4.25 ± 42.76
GASI.MI	-4.06 ± 48.36	**9.05 ± 37.81**
GSK.L	4.00 ± 89.44	**13.45 ± 29.91**
HSBA.L	-12.65 ± 124.26	-12.45 ± 155.31
ING.AS	-67.40 ± 261.91	**-11.01 ± 343.28**
LGEN.L	**5.13 ± 36.38**	2.53 ± 37.24
LSE.L	4.40 ± 16.39	**5.94 ± 18.55**
NOK1V.HE	**-7.65 ± 34.70**	-10.08 ± 52.10
SAN.MC	-4.98 ± 144.47	**39.59 ± 255.68**
VOD.L	**15.70 ± 43.55**	6.65 ± 37.26

addition of eligibility traces — producing the Q(λ) and SARSA(λ) algorithms — on the other hand, improved the agents' performance and lead to policies that occasionally generated profits out-of-sample; see Table 5.7.

Broadly speaking, the results in Table 5.7 suggest that there are two key conclusions: (i) that Q(λ) *fails* to significantly outperform MMMW (the best benchmark) on average, *across all stocks*, when we exclude LGEN.L; (ii) and that SARSA(λ) performs better than both MMMW and Q(λ). Indeed, we found that SARSA(λ) achieved an average improvement of 139% over Q(λ) in terms of the mean ND-PnL (a proxy for the risk-neutral objective), and 1043% over MMMW. In the case of SAN.MC, the excess over Q(λ) was nearly as much as 900%. The analysis also showed that SARSA(λ) was also more stable during learning and tended to be more sample efficient, especially early in training. It is plausible that this discrepancy derives from the differences between on- and off-policy learning algorithms as it is well known that off-policy learning is likely to cause divergence when combined with function approximation and bootstrapping [13, 19].

These claims are in agreement with conclusions first drawn by Chan and Shelton [40] back in 2001. In this case, however, the pathologies are exacerbated by the addition of eligibility traces as they are known to be less effective at assigning credit with Q(λ) compared with on-policy methods. This is due to the distinction between the sampling distributions of the behaviour and target policies [162]; credit can only be propagated a short way back up the history because exploratory actions are not drawn from the target distribution and the eligibility trace is reset. While solutions have been proposed, such as importance sampling [131, 132, 148], there are still very few simple approaches that can be used without introducing additional instability issues. As such, it may be that traditional Q(λ) is ill-suited to this domain.

IS uses likelihood ratios which are unstable when the probability of a sample under the proposal distribution approaches zero.

Partial observability is also a major challenge. It is well known that Q-Learning may not converge on the optimal value function when the observations are not sufficiently informative [153].

5.6.2 *Average Reward vs. Discounted Reward*

Market making can clearly be formulated as a finite horizon problem. A trading day has the natural interpretation as an episode, and market makers often behave

Table 5.8: Mean and standard deviation on the normalised daily PnL (given in units of 10^4) for the off- and on-policy variants of the R(λ) algorithm, both evaluated over the whole basket of securities.

	Off-Policy R(λ)	**On-Policy R(λ)**
CRDI.MI	**5.48 ± 25.73**	0.00
GASI.MI	-3.57 ± 54.79	**4.59 ± 17.27**
GSK.L	12.45 ± 33.95	**14.18 ± 32.30**
HSBA.L	-22.97 ± 211.88	**9.56 ± 30.40**
ING.AS	-244.20 ± 306.05	**18.91 ± 84.43**
LGEN.L	-3.59 ± 137.44	**-1.14 ± 40.68**
LSE.L	**8.31 ± 23.50**	5.46 ± 12.54
NOK1V.HE	-0.51 ± 3.22	**0.18 ± 5.52**
SAN.MC	8.31 ± 273.47	**25.14 ± 143.25**
VOD.L	32.94 ± 109.84	**16.30 ± 32.69**

with this sequential structure in mind — it is common for MMs to end each trading day flat (i.e. with $\Omega_T = 0$) to avoid overnight changes in price. While this is the approach taken so far, it is unclear whether this "natural" formulation is necessarily the best approach from a learning perspective. Here we evaluate the performance of R-Learning, a variant of the Q-Learning algorithm which optimises the average long-run return rather than the discounted return [162]. As with Q-Learning, the R-Learning algorithm can be extended to leverage eligibility traces for improved credit assignment. The resulting algorithm, R(λ), can then also be adapted to work on-policy such that it takes the form of an average-reward equivalent of SARSA(λ). That is, rather than take the maximum over actions in the next state, we simply sample the current policy.

The performance of the on- and off-policy variants of R(λ) on each of the 10 securities is quoted in Table 5.8. From these empirical results we can make the following comparisons:

(i) The on-policy variant of R(λ) achieves an out-of-sample ND-PnL with an average excess of 71% over the off-policy variant.

(ii) The pessimistic behaviour of the on-policy variant leads to much lower variance on ND-PnL.

(iii) Off-policy R(λ) outperforms Q(λ) by an average of 24%.

(iv) On-policy R(λ) outperforms SARSA(λ) by an average of 36%.

This suggests two things: first, that the on-policy/off-policy discrepancy observed previously translates to the average-reward setting; and second, that both variants of R(λ) outperform their discounted return counterparts on average. This does not necessarily mean that a finite-horizon perspective is incorrect. It is hard to draw strict conclusions as to the cause, but it is plausible that the long-run average term in the R(λ) update may aid learning. For example, there could be a regularising and/or variance reducing effect — much like the baseline in actor-critic methods — due to the fact that it will converge faster to it's value $\mathbb{E}_{d_0,\pi}[G_0]$. This is because the long-run average is independent of states/actions and thus uses all observed samples rather than a subset based on the state visitation distribution $d(s)$.

5.6.3 *Bias-Variance Reduction*

As we have seen thus far, limit order books (LOBs) in high-frequency regimes are subject to a great deal of microstructure noise which increases the variance in our estimate of $Q_\pi(s, a)$. This frequently leads to unstable learning, especially when using off-policy evaluation, and more generally for sampling based algorithms in which the TD-target is computed from a single transition. While eligibility traces clearly improve stability, Table 5.7 and Table 5.8 suggest that there are still major issues. Assuming the proposition on the regularising effect of R(λ) is true, this would imply that variance reducing techniques will hold value.

In addition to variance, it is also well understood that algorithms featuring a max operation in the TD-target computation are susceptible to maximisation/overestimation bias. This pathology was first studied by Hasselt [82] who showed that Q-Learning can perform very poorly in stochastic environments due to the poor estimates of the maximum expected action-value; see Section 2.4 for more details. This is highly relevant to the MM problem and contributes yet another source of error, this time in the form of a bias.

To this end, we introduce three alternative algorithms that address each of the issues mentioned above. Double Q(λ) and Double R(λ) are both double-estimator variants of Q(λ) and R(λ), respectively. Expected-SARSA(λ) is then included as it is known to reduce variance in the estimate of the TD-target by evaluating the expectation of the value at the next state across all $a \in \mathcal{A}$; see Section 2.3 for more details. A summary of the performance of these algorithms is given in Table 5.9, for which we can state the following:

(i) Double Q(λ) outperforms Q(λ) by 90% on average, has more positive instances and appears to exhibit fewer cases of extreme values (ING.AS).

(ii) Double R(λ) similarly outperforms R(λ) by an average of 146%, Q(λ) by 159%, and again has fewer extremal values.

(iii) Expected-SARSA(λ) only performs slightly better than SARSA(λ) on average (33%). As with Double Q(λ)/R(λ), this extension has much more consistent out-of-sample performance across the basket of securities.

Overall, we find no hard evidence that a single algorithm performs best *across all assets*. Indeed Table 5.9 suggests that all the proposed extensions perform well, with Expected-SARSA(λ) being the most consistent. However, it is important to note that the scale of variance on the distribution of ND-PnL makes it hard to draw strong conclusions when comparing to, say, regular SARSA(λ). By far the most congruous observation, as previously stated, is that on-policy methods are more stable in this domain; though eligibility traces and double-estimation do clearly improve performance.

5.7 CONSTRAINED BEHAVIOUR

In the preceding section we saw how better credit assignment, regularisation and algorithmic bias/variance reduction can significantly improve performance. The limitation is that this only considers the *expectation* of ND-PnL and not higher order characteristics such as variance. For this we must rethink the risk-neutral objective and incorporate penalties on behaviour that fails to satisfy our constraints. Historically speaking, risk has been integrated into RL through the use of exponential utility functions, mean-variance criteria or direct penalties on non-zero inventories, but these techniques often don't translate well, or require unintuitive tuning pro-

Table 5.9: Mean and standard deviation on the normalised daily PnL (given in units of 10^4) for two double-estimator techniques and a variance reduced estimator, each evaluated over the whole basket of securities.

	Double Q(λ)	(Off-Policy) Double R(λ)	Expected-SARSA(λ)
CRDI.MI	-5.04 ± 83.90	19.79 ± 85.46	0.09 ± 0.58
GASI.MI	5.46 ± 59.03	-1.17 ± 29.49	3.79 ± 35.64
GSK.L	6.22 ± 59.17	21.07 ± 112.17	-9.96 ± 102.85
HSBA.L	5.59 ± 159.38	-14.80 ± 108.74	25.20 ± 209.33
ING.AS	58.75 ± 394.15	5.33 ± 209.34	6.07 ± 432.89
LGEN.L	2.26 ± 66.53	-1.40 ± 55.59	2.92 ± 37.01
LSE.L	16.49 ± 43.10	6.06 ± 25.19	6.79 ± 27.46
NOK1V.HE	-2.68 ± 19.35	2.70 ± 15.40	-3.26 ± 25.60
SAN.MC	5.65 ± 259.06	32.21 ± 238.29	32.28 ± 272.88
VOD.L	7.50 ± 42.50	25.28 ± 92.46	15.18 ± 84.86

cesses. Below we introduce two alternative definitions of reward that use *damping* to penalise risky policies without losing interpretability.

Definition 15 (Symmetrically damped reward)**.** *Let* $\eta \in [0, 1]$ *be a constant and define the symmetrically damped reward as*

$$r_t \doteq \Delta\Upsilon_t - \eta r_t^Z = r_t^+ + r_t^- + r_t^m + (1-\eta)\Omega_t \Delta Z_t. \tag{5.6}$$

Definition 16 (Asymmetrically damped reward)**.** *Let* $\eta \in [0, 1]$ *be a constant and define the asymmetrically damped reward as*

$$r_t \doteq \Delta\Upsilon_t - \eta \max\left[0,\, r_t^Z\right]. \tag{5.7}$$

The proposed reward damping is applied only to the inventory PnL term, $r_t^Z = \Omega_t \Delta Z_t$, in order to reduce the reward that the MM can earn from speculation on the future value of $\Omega_{t'>t}$. The symmetric version targets both profits and losses from speculation, while asymmetric damping reduces only the profit derived from speculative positions and keeps losses intact. In both cases, the amount of reward that can be gained from capturing the spread increases relative to the amount of reward that can be gained through speculation. This has the effect of encouraging "good" market making behaviour in the policy. A nice property of this formulation is that the penalty is in the same units as the reward itself, making η simple to tune and interpret; e.g. a value $\eta = 0.5$ in Equation 5.7 would halve the amount of reward generated by appreciations in the value of the agent's inventory Ω_t.

Both of the proposed extensions to the risk-neutral reward function were evaluated across the full basket of securities; the performance is summarised in Table 5.10. While symmetric damping can be seen to exacerbate the flaws in the basic agent, asymmetric damping of the speculative reward term, with sufficiently high penalty, produced significantly better risk-adjusted performance in most cases. This is exemplified by Figure 5.2 which shows how the distribution of out-of-sample PnL and MAP varies with η; note that the PnL converges to a positive, non-zero value. For example, at $\eta \sim 0.1$ there is an apparent regime shift at which the agent starts converging on significantly different policies than those found in the risk-neutral case. This shift in solutions is manifested in a change from holding large, biased

Table 5.10: Mean and standard deviation on the normalised daily PnL (given in units of 10^4) for Q-learning and SARSA using non-damped PnL reward function and agent-state.

	Symmetric ($\eta = 0.6$)	**Asymmetric** ($\eta = 0.6$)
CRDI.MI	12.41 ± 143.46	0.08 ± 2.21
GASI.MI	9.07 ± 68.39	-0.10 ± 1.04
GSK.L	30.04 ± 135.89	9.59 ± 10.72
HSBA.L	-11.80 ± 214.15	13.88 ± 10.60
ING.AS	90.05 ± 446.09	-6.74 ± 68.80
LGEN.L	5.54 ± 119.86	4.08 ± 7.73
LSE.L	8.62 ± 27.23	1.23 ± 1.80
NOK1V.HE	-4.40 ± 84.93	0.52 ± 3.29
SAN.MC	27.38 ± 155.93	5.79 ± 13.24
VOD.L	8.87 ± 93.14	9.63 ± 6.94

inventories towards small, neutral positions, and a reduction in the dispersion of the distribution of PnL. Though the point at which this occurs does vary between securities, the impact of asymmetric damping is clear and provides strong evidence that the inventory term, r_t^Z, is the leading driver of risky behaviour. This result corroborates the use of inventory penalising terms in value functions in stochastic optimal control [34, 35] and is a key contribution of this chapter.

In addition to better asymptotic performance, the asymmetrically damped reward function also exhibited improved stability during learning. Figure 5.3 shows how the mean and standard deviation of episodic reward varied with increasing values of the damping factor, η. Observe how the standard deviation for $\eta = 0$ (i.e. risk-neutral) diverges as the mean reward grows. This implies that a small increase in the mean PnL comes at the cost of a large increase in the risk. Correspondingly, we find that while the mean reward is reduced for $\eta \in \{0.05, 0.1\}$ compared with $\eta = 0$, they come with much greater stability during learning and out-of-sample results. This, however, does not appear to translate to $\eta = 0.7$ which outperforms the other instances both in terms of the mean *and* variance on profit and loss. This is likely due to a reduction in variance of the reward signal, leading to more stable learning.

These results initially seemed to be unintuitive, but we discovered empirically that this was caused by sub-optimality in the solutions found for $\eta < 0.7$. Specifically, the inventory component r_t^Z in Equation 5.5 not only drives risky behaviour, but is also the main source of instability during learning. Increasing the value of η was found to yield better and more consistent performance. In other words, the penalty acts not only as a constraint on behaviour, but also as a regulariser much like the average-reward term in $R(\lambda)$; see also the work of Spooner, Vadori, and Ganesh [160] who exploit this ex ante knowledge of improved performance in policy gradient methods. Since the majority of *uninformative* randomness derives from ΔZ_t, it follows that the damping has the effect of reducing variance in the estimator $\widehat{Q}(s, a)$; this effect should scale quadratically with η. The benefits of the asymmetrically damped reward function are clearly two-fold.

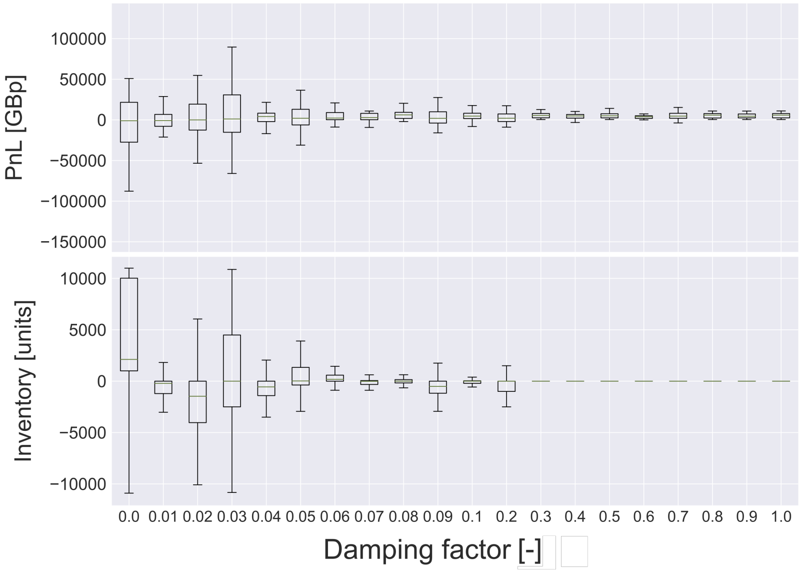


Figure 5.2: Distributions of daily out-of-sample PnL and mean inventory for increasing values of the damping factor, η, evaluated on HSBA.L using the asymmetric reward variant.

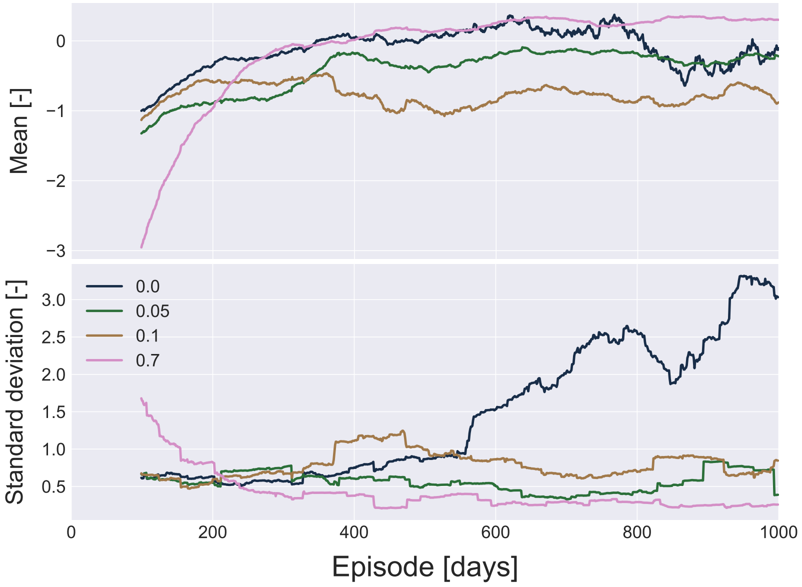


Figure 5.3: Rolling mean and standard deviation of the average episodic reward during training for increasing values of the damping factor, η, evaluated on HSBA.L.

Table 5.11: Mean and standard deviation on the normalised daily PnL (given in units of 10^4) for SARSA(λ) using the risk-neutral reward and either the joint- or factored-state representation.

	Joint-State	**Factored-State**
CRDI.MI	-31.29 ± 27.97	-5.32 ± 52.34
GASI.MI	-35.83 ± 13.96	5.92 ± 40.65
GSK.L	-31.29 ± 27.97	5.45 ± 40.79
HSBA.L	-84.78 ± 31.71	-0.79 ± 68.59
ING.AS	-189.81 ± 68.31	9.00 ± 159.91
LGEN.L	-14.39 ± 9.38	6.73 ± 22.88
LSE.L	-6.76 ± 11.52	3.04 ± 5.83
NOK1V.HE	-9.30 ± 23.17	-2.72 ± 19.23
SAN.MC	-144.70 ± 104.64	52.55 ± 81.70
VOD.L	-21.76 ± 17.71	7.02 ± 48.80

5.8 STATE AUGMENTATION

Thus far we have defined the observations of the MM by tuples of the form $(\Omega, \delta^+/\beta, \delta^-/\beta)$, but this ignores many of the features of the order book, $\mathcal{B}$, that can help decision making. For example, it is well known that the imbalance of volume between the bid and ask is a good short-term predictor of price [32, 99]. One would conjecture that incorporating other market information into the state representation would yield a better estimator, $\widehat{Q}(s, a)$, of the true value $Q_\pi(s, a)$. The key challenge is balancing expressivity with informational value and avoiding Bellman's curse of dimensionality. Drawing on the extensive literature on market microstructure, we propose to extend the agent-state with the following indicators (for more details see Section 4.4):

(i) Market spread.

(ii) Mid-price move.

(iii) Volume imbalance.

(iv) Signed volume.

(v) Volatility.

(vi) Relative strength index.

While these are by no means the only choice, we argue that it is a natural selection of indicators for the equities setting considered here.

The mean out-of-sample ND-PnL of this extended observation construction — which we label the *joint-state* representation — with the risk-neutral objective is quoted in Table 5.11. Notice that *all securities lost money* on average, suggesting that SARSA(λ) failed to find a performant instance of the strategy at all. Since the MM still has access to the agent-state, it would seem that the extended representation either introduces too much noise, or is too large to effectively explore given the limited amount of data. Contrary to intuition, we did not even observe any considerable improvement in performance with increased training. Instead, the agent was regularly seen to degrade and even diverge (both in episodic reward and TD-error); this may correspond to the induced non-stationary in the conventional Bellman operator as identified by Bellemare, Ostrovski, Guez, Thomas, and Munos [16].

At this point, one might perform an ablation study and/or sensitivity analysis to gauge the contribution of each of the 6 predictors. This is very time consuming and requires extensive meta-optimisation; which again requires more data. Instead, we propose a *factored* representation.

5.8.1 *Factored Representation*

Assume that there are N_Q independent tile coding bases, each using some set of quantities derived from the raw observation $(\Omega, \mathcal{B})$. Now, let $\widehat{Q}_i(s, a) \doteq \langle \boldsymbol{\phi}_t(s, a), \boldsymbol{w}_i \rangle$ denote the value estimate of the state, s, given by the i^{th} set of basis functions, $\boldsymbol{\phi}_i$. The total value of a state is then defined as the sum:

$$\widehat{Q}(s, a) = \sum_{i=1}^{N_Q} c_i \widehat{Q}_i(s, a) = \sum_{i=1}^{N_Q} c_i \left\langle \boldsymbol{\phi}_i(s, a), \boldsymbol{w}_i \right\rangle, \tag{5.8}$$

where $\boldsymbol{w}_i$ are the weight vectors of the i^{th} approximator and the c_i factors are constrained such that $\sum_{i=1}^{N_Q} c_i = 1$. This approach amounts to learning an ensemble of value functions, each of which is updated using the same TD-error but a different basis. Related approaches involve using multiple sets of basis functions with varying granularity [50], but using the same set of state variables, and specific applications to dimension mapping for dynamics modelling [54, 76]. It has been argued that a coarse representation improves learning efficiency of a high resolution representation by directing the agent towards more optimal regions of policy space. The justification here is the same, the crucial difference is that we exploit the (approximate) *independence between variables*.

Here we consider an instance of the factored value function construction in Equation 5.8 with $N_Q = 3$, with basis functions for: (i) the agent-state; (ii) the market-state; and (iii) the joint-state. A comparison between the performance of the original joint-state approach and this proposed factorisation is provided in Table 5.11; note again these were evaluated only on the risk-neutral objective. Evidently, the factored approach performs better, and indeed it enjoys an 18% improvement in the ND-PnL on average over the agent-state SARSA(λ) algorithm (see Table 5.7). Interestingly, we also find that the variance on the out-of-sample ND-PnL is significantly reduced in a number cases, including HSBA.L and GSK.L.

This approach is particularly relevant for problems in which a lot of domain-specific knowledge is available a priori. For example, consider a trading agent holding an inventory of, say, 100 units of a stock. Though the expected future value of the holdings are surely conditional on the state of the market, the most important factor for the agent to consider is the risk associated with being exposed to unpredictable changes in price. Learning the value for the agent-, market- and joint-state representations independently and in parallel enables the agent to learn this concept much faster as it doesn't rely on observing every permutation of the joint-state to evaluate the value of it's inventory. We claim that this helps the agent converge to better solutions by guiding the agent away from local optima in policy space.

5.9 CONSOLIDATION

Up until now we have treated each contribution in isolation. These have addressed bias/variance in the TD updates, risk sensitivity, and how to incorporate additional

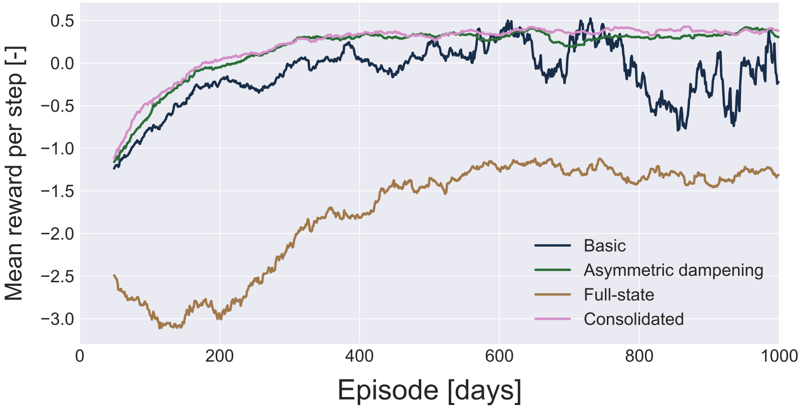


Figure 5.4: Rolling mean (period 50) of the average episodic reward for basic, damped (asymmetric), joint-state and consolidated agents training (HSBA.L).

Table 5.12: Mean and standard deviation of ND-PnL (given in units of 10^4) and MAP for SARSA(λ) using the consolidated agent.

	ND-PnL [10^4]	MAP [units]
CRDI.MI	0.15 ± 0.59	1 ± 2
GASI.MI	0.00 ± 1.01	33 ± 65
GSK.L	7.32 ± 7.23	57 ± 105
HSBA.L	15.43 ± 13.01	104 ± 179
ING.AS	-3.21 ± 29.05	10 ± 20
LGEN.L	4.52 ± 8.29	229 ± 361
LSE.L	1.83 ± 3.32	72 ± 139
NOK1V.HE	-5.28 ± 33.42	31 ± 62
SAN.MC	5.67 ± 13.41	4 ± 9
VOD.L	5.02 ± 6.35	46 ± 87

information without succumbing to the curse of dimensionality. For completeness, we now evaluate the performance of an algorithm combining:

(i) SARSA(λ);

(ii) the asymmetrically damped reward function; and

(iii) a factored $\widehat{Q}(s, a)$ function.

The learning curves for this algorithm and it's precursors are illustrated in Figure 5.4 which demonstrates the stability advantage of our consolidated approach. In the majority of cases, this agent was found to generate slightly lower returns than the best individual variants seen thus far — see Table 5.12 — but achieves significantly improved out-of-sample consistency. We also observe that the instances of this strategy tended to hold smaller inventories, which may have been a contributing factor towards the reduced variance on ND-PnL. Though the results vary slightly across the basket of securities, this consolidated agent was found to produce superior risk-adjusted performance over the basic agent and extended variants overall.

For example, Figure 5.5 compares the equity and inventory processes for the basic and consolidated agents' out-of-sample tests. Both time series illustrate that there

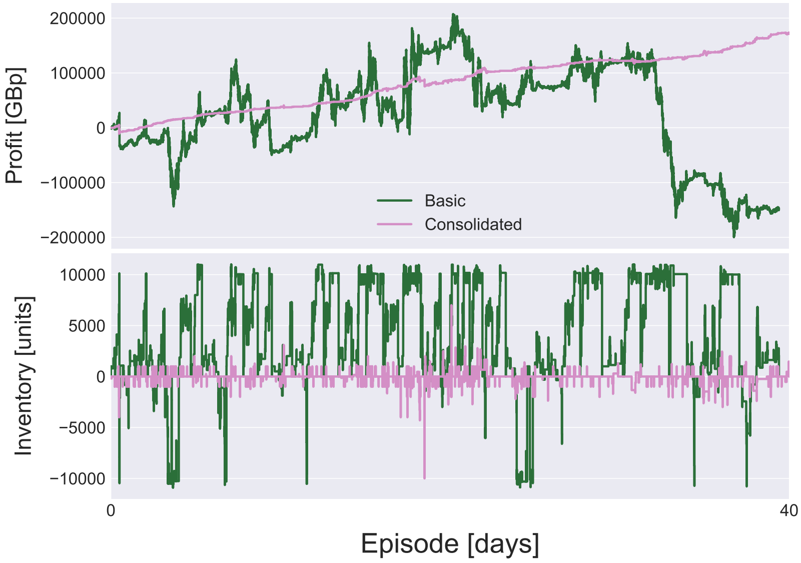


Figure 5.5: Out-of-sample equity curve and inventory process for the basic (naïve) and consolidated agents, evaluated on HSBA.L.

is a profound difference in behaviour between the two instances of the strategy. Where the former is highly volatile, the latter is stable. The naïve agent regularly holds a non-zero inventory, exposing itself to changes in the security's value for extended periods of time, leading to the noise observed in the equity curve. For the consolidated agent, it appears that the learnt policy targets a near-zero inventory, relying less on speculative trading and thus yielding the consistency one expects from a good market making strategy.

The process Ω_t is effectively described by an integer-valued autoregressive process.

5.10 CONCLUSIONS

In this chapter we have developed a suite of techniques to improve upon past RL-based methods in data-driven simulations of market making. The result is an algorithm that produces competitive out-of-sample performance across a basket of securities. We first developed a highly realistic simulation of the problem domain and then showed how eligibility traces solve the problems raised in past work around credit assignment and reward stochasticity. A range of different learning algorithms, reward functions and state representations were evaluated. Insight was provided through empirical analysis and variance analyses. We conclude by showing that a consolidation of the best techniques into a single agent yielded superior risk-adjusted performance across the whole dataset. To summarise: a combination of factored value function representations, well calibrated reward functions and on-policy learning algorithms help address the significant challenges highlighted in past research.

Part III

MODEL-DRIVEN TRADING

6 RL ∩ MODEL-DRIVEN TRADING

6.1 OUTLINE

In Part II we saw how RL can be used to derive market making strategies directly from historical reconstructions of an LOB market. This approach offers a powerful means by which to reduce the discrepancy between train-time and test-time dynamics. And the motivation is clear: a policy derived from *real* data should, in principle, translate more effectively to the actual market with comparable performance. To do this, however, relies on access to (very) high-quality and reliable data which is not only expensive, but time consuming and complex to process; as discussed in Chapter 4. Anything less than total access to all levels in the book and every event will lead to reconstruction errors and thus an epistemically biased policy. It is also very challenging to simulate market responses to artificial orders [179], and harder still to validate our approximations and assumptions — if possible at all — without incurring significant financial risk.

In this part of the thesis we explore a different direction. Rather than treat the environment as an actual black-box, we revert to using the models traditionally seen in the mathematical finance literature. These are typically expressed as stochastic differential equations as introduced in Chapter 3. In this context, RL has the interpretation as an approximate replacement for analytical dynamic programming. There are a number of advantages to this approach, such as:

(i) reduced computational cost;

(ii) interpretability; and

(iii) transparency.

Of course, there are also limitations of this type of approach. If the model is incorrectly specified, or the parameters therein are poorly calibrated, then the solution will also be wrong with respect to the true market dynamics. Indeed, one of the most important areas of financial modelling is precisely in the estimation of model parameters from data so as to reduce the gap between simulations and the real world. However, this is non-trivial and many would argue that one cannot fully capture the behaviour of the financial markets with parametric assumptions. Nevertheless, model-driven trading is an incredibly important paradigm, both for research and practical applications. What's more, we will show in Chapter 7 that this problem of invariance to epistemic risk can be addressed using adversarial RL, and that robustness to downside aleatoric risk can be tackled naturally via extensions to standard methods in Chapter 8.

The purpose of this chapter is thus to set the scene for the contributions of Chapter 7 and Chapter 8. We begin by introducing a set of policy classes that will be used throughout to represent different trading behaviours. These will be crucial to effectively tackle problems in which the action space is both continuous and restricted, and highlight the flexibility of reinforcement learning (RL). We then define the proposed models for four important problems in algorithmic trading: optimal liquidation, market making, portfolio optimisation and optimal control. Demonstrations of applying RL to these problems will be provided, as well as some non-standard extensions, and discussions about the challenges presented by each settings.

6.2 POLICY CLASSES

Stochastic policies fall under two categories — discrete and continuous — for which there are canonical probability distributions used to define the likelihood of an action. In discrete action-spaces, it is common to use a variant of the greedy (deterministic) policy, such as epsilon-greedy, with value-based methods, or a Gibbs distribution with policy gradient approaches. In continuous action-spaces, one typically uses a Normal distribution. For the most part, these choices are sensible and often perform very well, but they have limitations. In the following sections we introduce a taxonomy of policy classes that can be used to model a rich set of problems by taking into account the geometry of the underlying action space. While this list will not be exhaustive — and indeed we only present the univariate cases (though all generalise to multi-dimensional action-spaces) — we do cover key classes that give good empirical performance.

Specifying a probability distribution to use as the likelihood of an action requires a few key quantities as outlined in Section 2.4. To derive the vanilla policy gradient, we need be able to compute the score of a distribution and compute it's derivative. For continuous policies, this is given by the logarithm of the probability *density* function. For discrete policies, this is given by the logarithm of the probability *mass* function. We also, clearly, require that the score be continuously differentiable with respect to the parameters; i.e. the score function is of class C^1.

For understanding the advantage of natural gradients, it is also interesting to study second derivatives and the Fisher information of a policy distribution. The Cramér-Rao bound [46, 136] tells us that there is a minimum value for the variance of any unbiased estimator $\widehat{\theta}$ of θ. That is, $\mathbb{V}[\widehat{\theta}] \geqslant 1/\mathcal{I}(\theta)$, where $\mathcal{I}(\theta)$ is the Fisher information. This means that the variance on finite-sample MLE estimators scales inversely with $\mathcal{I}$, and thus the updates used in stochastic gradient descent should account for this by adjusting the per-parameter learning rates. A policy that exhibits extrema in the values of $\mathcal{I}$ can otherwise lead to gradient saturation or even numerical instability and divergence. For example, as we shall see next, the gradient of the score function for the Normal distribution with respect to the mean explodes as the variance tends to zero. The value of natural gradients cannot be understated and building intuition into why likelihoods are susceptible/stable is key.

6.2.1 *Supported on* $\mathbb{R}$

The Normal distribution is by far the most prominent likelihood to use in problem domains where $\mathcal{A}$ is continuous. Arguably, the main reason for this is its ubiquity, amenable properties and relative simplicity. A policy of this type has the probability density function

$$\pi_\theta(a \,|\, s) = \frac{1}{\widehat{\sigma}\sqrt{2\pi}} e^{\frac{-(a-\widehat{\mu})^2}{2\widehat{\sigma}^2}}, \tag{6.1}$$

where $\widehat{\mu} = \widehat{\mu}_{\theta_\mu}(s)$ and $\widehat{\sigma} = \widehat{\sigma}_{\theta_\sigma}(s)$ are parameterised functions of state yielding the mean (location parameter) and standard deviation (scale parameter) of the distribution. The score function of a Normal policy with respect to its parameters, θ_μ and θ_σ, is then given by

$$\frac{\partial}{\partial\theta_\mu} \ln \pi_\theta(a \,|\, s) = \frac{a-\widehat{\mu}}{\widehat{\sigma}^2} \frac{\partial\widehat{\mu}}{\partial\theta_\mu}, \tag{6.2}$$

and

$$\frac{\partial}{\partial\theta_\sigma} \ln \pi_\theta(a \,|\, s) = \left[\frac{(a-\widehat{\mu})^2}{\widehat{\sigma}^3} - \frac{1}{\widehat{\sigma}}\right] \frac{\partial\widehat{\sigma}}{\partial\theta_\sigma}. \tag{6.3}$$

An illustration of the score surface — of which the policy gradient is proportional — for different instances and action samples is given in Figure 6.1.

It is important to observe that while the shape of the surfaces for μ and σ are very similar, the *scale* for σ is much larger as a consequence of the higher exponents in the derivative. As a result of this, the learning rate (and indeed other hyperparameters) used in policy gradient methods must be tuned to take into account the potential instability arising from actions sampled in the tails of the distribution. This can be seen from the diagonal terms of the Fisher information matrix

$$\mathcal{I}(\mu, \sigma^2) = \begin{bmatrix} \frac{1}{\sigma^2} & 0 \\ 0 & \frac{1}{2\sigma^4} \end{bmatrix}.$$

In other words, the lower bound on the variance of $\widehat{\sigma}$ (assuming it is unbiased) scales with σ^4, two orders of magnitude greater than for μ. This also highlights a source of instability that can be introduced if the variance estimate is allowed to decrease to zero. In this regime, the score functions grow very large very quickly and, in a computational setting, one often gets divergence and instability without careful tuning of the learning rate. A better solution is to use methods that take into account the shape of this manifold, such as NAC [129] or PPO [144].

6.2.2 *Supported on Half-Bounded Intervals of* $\mathbb{R}$

In many problem settings there is some lower/upper bound on the value an action can take; e.g. for $\mathcal{A} = \mathbb{R}_+$. This is usually dealt with by "clipping" the sampled action and using either the original value or the transformed value to compute the policy gradient update. As noted by Fujita and Maeda [60], however, both of these approaches introduce bias that is detrimental to learning. In some sense, the former treats the environment as a black-box in which we only know that the action-space is some subset of the reals. The issue is that any actions falling outside the bounds are indistinguishable with respect to the dynamics, but the corresponding updates may be very different. In the latter case, a sampling bias is introduced since the limit inherits all the lost probability density. The approach of Fujita and Maeda [60] is to use the CDF of the policy distribution at the boundary of the action-space to perform a correction; this is independent of the policy distribution itself. Alternatively, one can simply choose a policy class that is supported, by construction, on semi-infinite intervals.

Given a Normal random variable $X \sim \mathcal{N}(\mu, \sigma^2)$ with mean μ and variance σ^2, the random variable $Y \sim |X|$ is said to have a folded Normal distribution. This new variable Y has support on the non-negative reals, $\mathbb{R}_+$. A policy of this form has a probability density function given by

$$\pi_\theta(a \mid s) = \sqrt{\frac{2}{\pi\widehat{\sigma}^2}}\, e^{\frac{-(a^2+\widehat{\mu}^2)}{2\widehat{\sigma}^2}} \cosh\left(\frac{\widehat{\mu}a}{\widehat{\sigma}^2}\right), \tag{6.4}$$

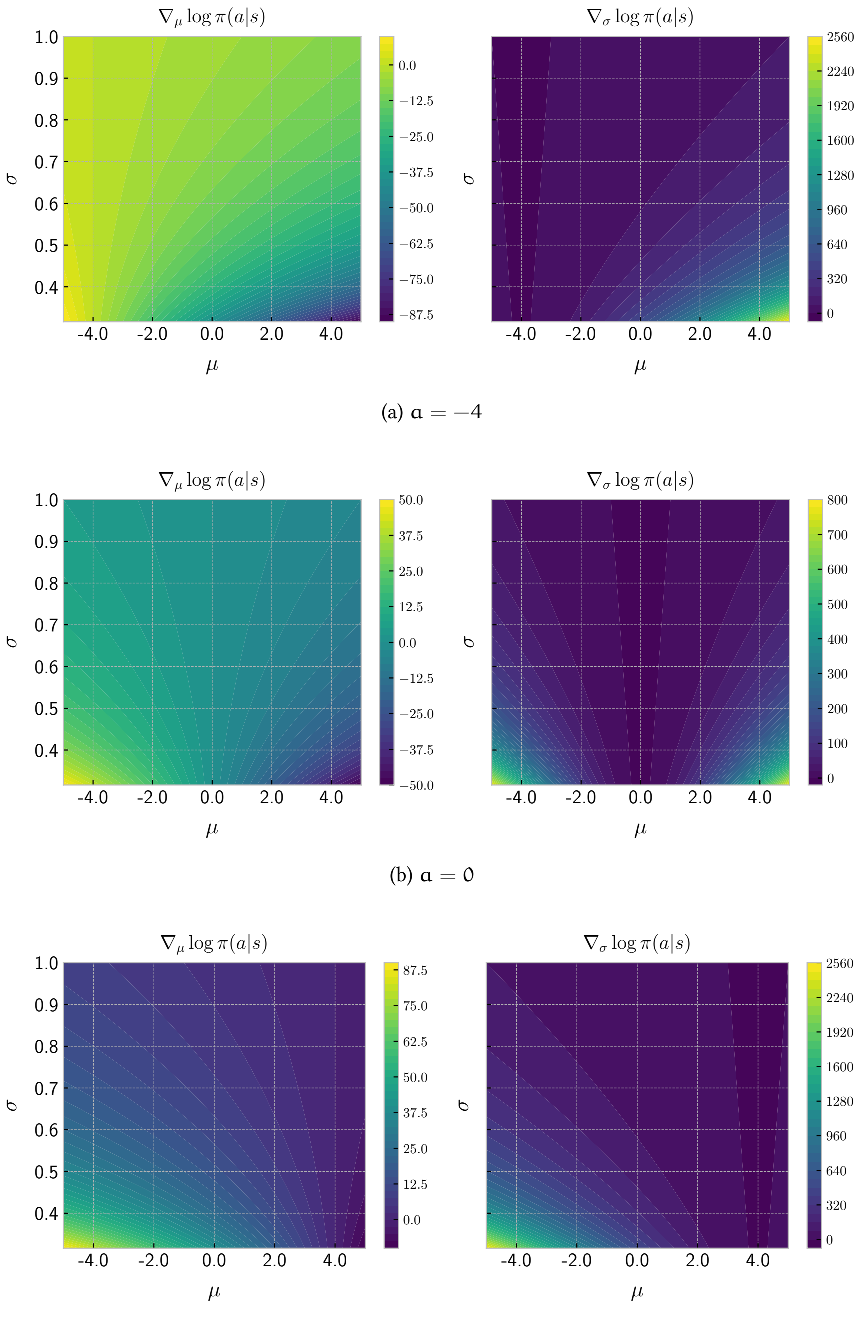
where, as before, $\widehat{\mu} = \widehat{\mu}_{\theta_\mu}(s)$ and $\widehat{\sigma} = \widehat{\sigma}_{\theta_\sigma}(s)$. The score of the policy then takes values

$$\frac{\partial}{\partial\theta_\mu} \ln \pi_\theta(a \mid s) = \frac{1}{\widehat{\sigma}^2}\left[a \tanh\left(\frac{\widehat{\mu}a}{\widehat{\sigma}^2}\right) - \widehat{\mu}\right] \frac{\partial\widehat{\mu}}{\partial\theta_\mu}, \tag{6.5}$$

and

$$\frac{\partial}{\partial\theta_\sigma} \ln \pi_\theta(a \mid s) = \frac{1}{\widehat{\sigma}^3}\left[\widehat{\mu}^2 + \widehat{\sigma}^2 + a^2 - 2\widehat{\mu}a \tanh\left(\frac{\widehat{\mu}a}{\widehat{\sigma}^2}\right)\right] \frac{\partial\widehat{\sigma}}{\partial\theta_\sigma} \tag{6.6}$$

for which an illustration is provided in Figure 6.2. Note that these derivatives are almost identical to those of the Normal distribution. The key difference is the addition

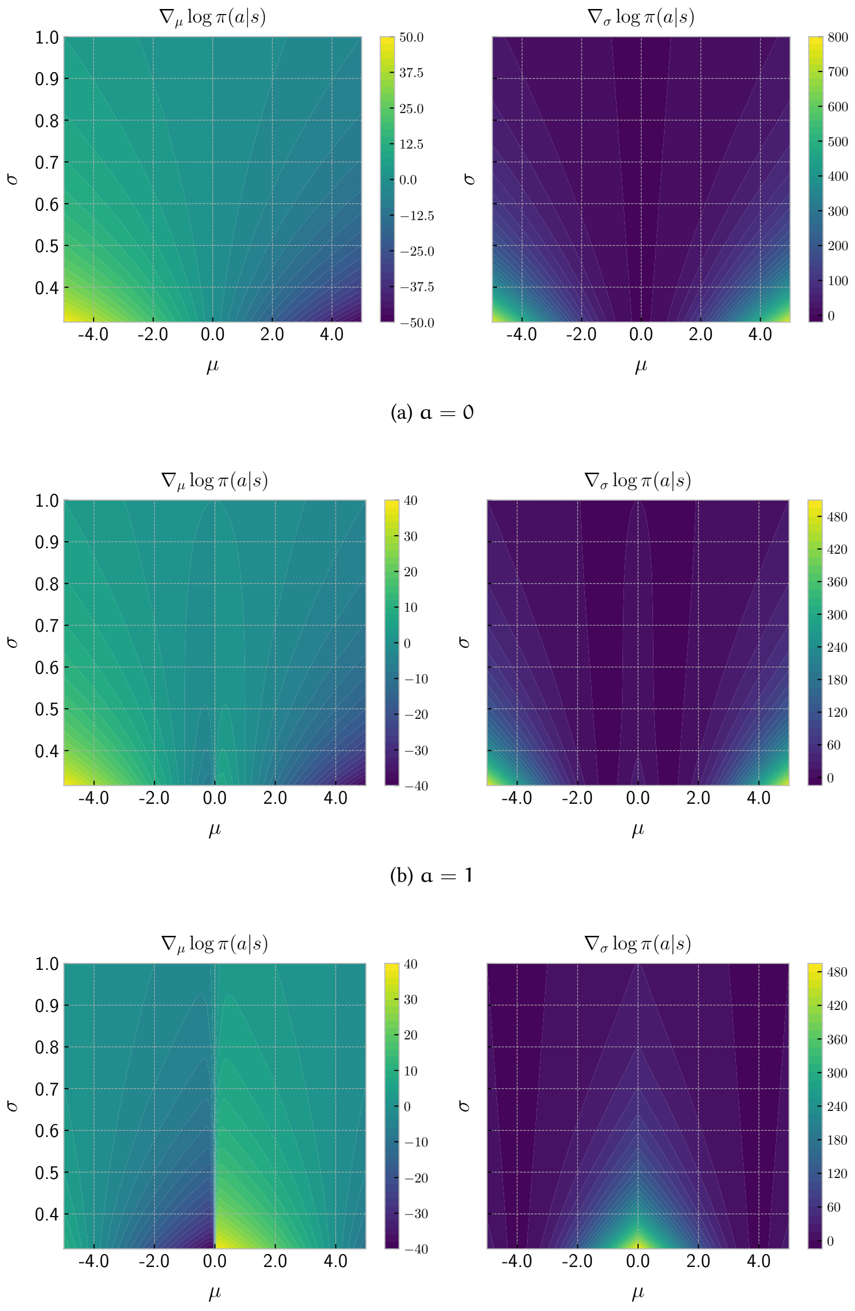


(a) $a = -4$

(b) $a = 0$

(c) $a = 4$

Figure 6.1: Illustration of the level sets of the score function with respect to μ and σ for a policy parameterised with a Normal distribution.



(a) $a = 0$

(b) $a = 1$

(c) $a = 4$

Figure 6.2: Illustration of the level sets of the score function with respect to μ and σ for a policy parameterised with a folded Normal distribution.

of a multiplicative correction: $\tanh(\widehat{\mu}a/\widehat{\sigma}^2)$. This has the effect of saturating the derivative near the origin. As with the Normal distribution, we find that the score with respect to θ_σ is much greater than for θ_μ. This makes sense given that the folded Normal distribution is just a transformed variant of the Normal distribution. What's interesting here, is that the score now exhibits asymmetric (respectively, symmetric) behaviour about the origin for θ_μ (and θ_σ). This derives from the symmetry about zero of the absolute function, and might suggest that $\widehat{\mu}(s)$ should be kept strictly non-negative to prevent ringing about the origin.

Remark. *There is a wide range of distributions with support on semi-infinite intervals: the Gamma, Fréchet and Weibull distributions to name a few. In some sense these seem more "natural" as they are defined on $[0, \infty)$, or $(0, \infty)$, by construction. However, these more "natural" distributions often suffer from very wide tails and significant asymmetry (skew) which — in the vast majority of applications — are inappropriate and lead to extreme exploration. An advantage of the folded Normal is that it inherits most of the properties of the Normal distribution, and is essentially identical when $\mu > 3\sigma$.*

Extreme exploratory actions may not always be a bad thing. Indeed, a Laplace distribution may be an interesting choice over a Normal in some problem settings.

6.2.3 *Supported on Bounded-Intervals of* $\mathbb{R}$

Another common setting for reinforcement learning (RL) problems are those with a bounded action space; i.e. $\mathcal{A} = [b, c]$ with b and c finite. For example, actions could represent probabilities, torque values of a robot arm or limit orders under inventory constraints. For this class of policy we follow the work of Chou, Maturana, and Scherer [42] and make use of the Beta distribution, a well understood distribution which has found great value in probabilistic modelling. This yields a policy with probability density function

$$\pi_\theta(a \mid s) = \frac{1}{B(\widehat{\alpha}, \widehat{\beta})} a^{\widehat{\alpha}-1} (1-a)^{\widehat{\beta}-1} \tag{6.7}$$

where $\widehat{\alpha} = \widehat{\alpha}_{\theta_\alpha}(s)$ and $\widehat{\beta} = \widehat{\beta}_{\theta_\beta}(s)$ are parameterised function approximators. The normalisation term, $B(\alpha, \beta) = \Gamma(\alpha)\Gamma(\beta)/\Gamma(\alpha+\beta)$, is defined in terms of the gamma function $\Gamma(u) = \int_0^\infty x^{u-1} e^{-x}\,dx$ for $u > 0$. The corresponding score function is then given by the column vector composed of

$$\frac{\partial}{\partial\theta_\alpha} \ln \pi_\theta(a \mid s) = \left[\ln(a) + \widetilde{\psi}(\alpha, \alpha+\beta)\right] \frac{\partial\widehat{\alpha}}{\partial\theta_\alpha},$$

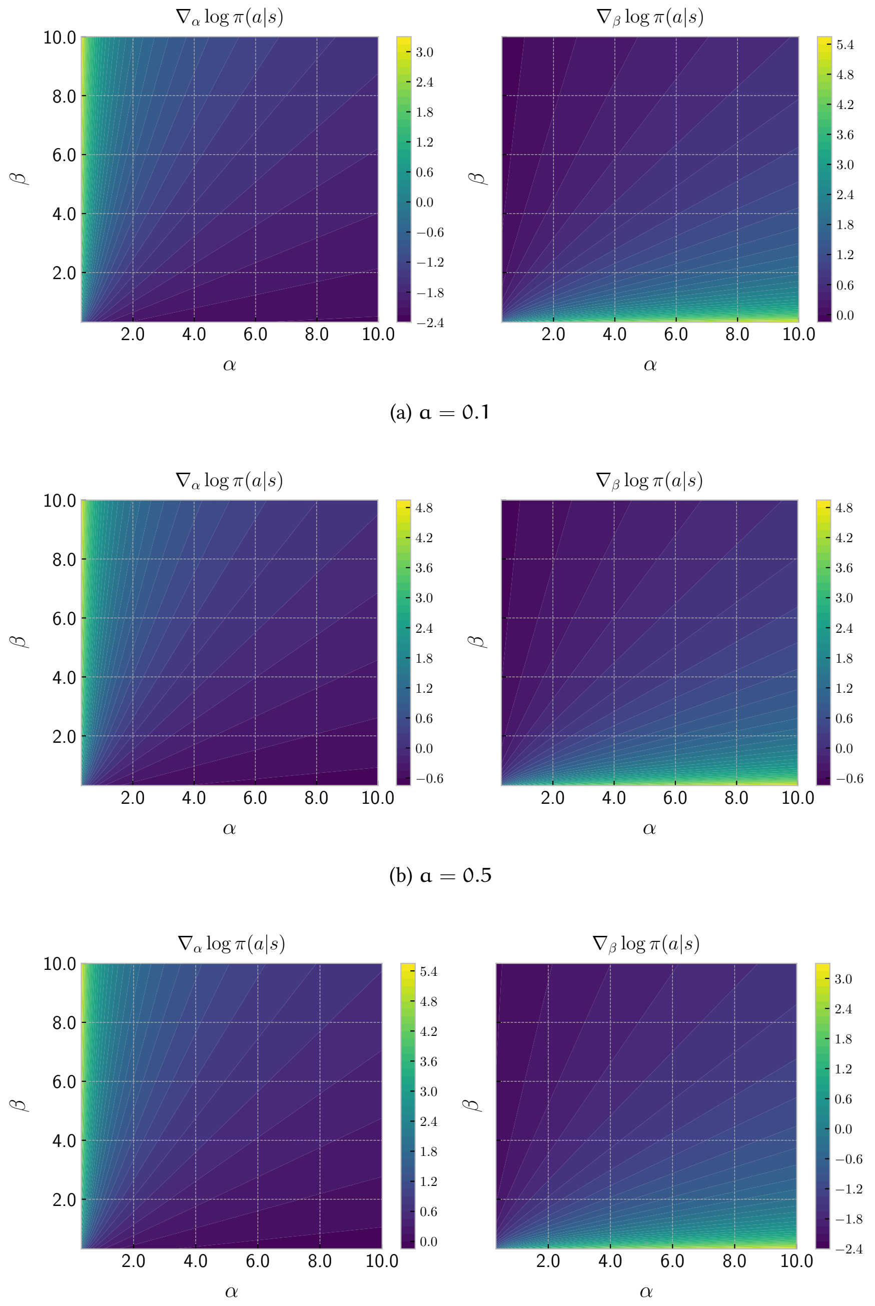
and

$$\frac{\partial}{\partial\theta_\beta} \ln \pi_\theta(a \mid s) = \left[\ln(1-a) + \widetilde{\psi}(\beta, \alpha+\beta)\right] \frac{\partial\widehat{\beta}}{\partial\theta_\beta},$$

where $\widetilde{\psi}(x, y) \doteq \psi(y) - \psi(x)$, and $\psi(u) = \frac{d}{du} \ln(\Gamma(u))$ is the digamma function. Unlike for Normal and folded Normal policies, the Beta distribution has a clear symmetry between score with respect to θ_α and θ_β. This is exemplified in Figure 6.3.

As pointed out by Chou, Maturana, and Scherer [42], the Beta distribution suffers from issues deriving from the curvature of the likelihood function as with the Normal distribution. In this case, however, we observe the reverse problem. While the Normal distribution is prone to overshoot when the variance tends to zero, the gradient of the Beta likelihood reduces to zero with increasing determinism. Specifically, the Fisher information matrix of a Beta distributed random variable, $X \sim B(\alpha, \beta)$, is given by:

$$\mathcal{I}(\alpha, \beta) = \begin{bmatrix} \mathbb{V}[\ln X] & \mathrm{Cov}[\ln X, \ln(1-X)] \\ \mathrm{Cov}[\ln X, \ln(1-X)] & \mathbb{V}[\ln(1-X)] \end{bmatrix}.$$



(a) $a = 0.1$

(b) $a = 0.5$

(c) $a = 0.9$

Figure 6.3: Illustration of the level sets of the score function with respect to α and β for a policy parameterised with a Beta distribution.

It thus follows directly from the Cramér-Rao bound that the policy gradient will essentially vanish later in training as the agent becomes more confident. Again, a standard way to deal with this in the literature is to use trust-region methods such as natural gradients.

6.2.4 *Supported on* $\mathbb{N}$

I'm sure this makes computer scientists uncomfortable, but rest assured $\mathcal{A} = \{0, \dots, n-1\}$ *is allowed too.*

The final class of problems with importance in RL are those settings with a discrete action space where $\mathcal{A} = \{1, \dots, n\} \subset \mathbb{N}$ and $n < \infty$ is the number of actions. For this we adopt the conventional choice of using a softmax selection rule corresponding to a Gibbs (or Boltzmann) distribution over the actions, such that

$$\pi_\theta(a \,|\, s) = \frac{e^{\widehat{f}(s,a)}}{\sum_{a' \in \mathcal{A}} e^{\widehat{f}(s,a')}}.$$

Here the function $\widehat{f}(s, a) \doteq \widehat{f}_\theta(s, a)$ is a continuously differentiable function of state and action, the interpretation of which is that of a potential (or energy) associated with the action a. The denominator is thus a partition function over the action space $\mathcal{A}$. As before, we can derive the score function of a policy with a Gibbs distribution by taking the logarithmic derivative:

$$\begin{aligned} \frac{\partial}{\partial \theta} \ln \pi_\theta(a \,|\, s) &= \frac{\partial \widehat{f}(s,a)}{\partial \theta} - \sum_{a' \in \mathcal{A}} \pi_\theta(a' \,|\, s) \frac{\partial \widehat{f}(s,a')}{\partial \theta}, \\ &= \frac{\partial \widehat{f}(s,a)}{\partial \theta} - \mathbb{E}_\pi\left[\frac{\partial \widehat{f}(s,\cdot)}{\partial \theta}\right]. \end{aligned} \tag{6.8}$$

The latter expression shows how the score function is given by the excess compared to the probability-weighted average for each action.

6.2.5 *Supported on Product Spaces*

In some cases, an action space has different structure in each dimension; e.g. $\mathcal{A} = [b, c] \times [d, \infty)$. For this it is not as simple as extending one of the previous distributions to the multivariate case as this will introduce boundary bias in one of the dimensions. A simple solution that we make use of later in the thesis is to define an independent product of distributions. Take $\mathcal{A}$ defined above, for this particular space one could combine a Beta distribution with a folded Normal. In general, the probability density function for an n-dimensional action space is defined as $\pi_\theta(\mathbf{a} \,|\, s) = \prod_{i=1}^n \pi_{\theta_i}^{(i)}(a|s)$ (under the assumption of independence), where θ is a column vector of weights. The score function then reduces to a simple summation:

$$\frac{\partial}{\partial \theta} \ln \pi_\theta(a \,|\, s) = \sum_{i=1}^{n} \frac{\partial}{\partial \theta} \ln \pi_{\theta_i}^{(i)}(a|s).$$

With this we can derive the policy gradient for any combination of distributions and model potentially very complex action spaces in a natural way. That is, in a way that adheres to the basic geometry of the space.

6.3 OPTIMAL EXECUTION

In this thesis, the optimal execution problem is modelled using the framework originally proposed in the seminal work of Almgren and Chriss [5]; see also [36]. In this setup, the agent is challenged with specifying a trading strategy, ν_t, that minimises the cost of execution (equivalently, maximises profit). We consider the special case of linear price impact, both temporary and permanent, and define the price process by the following difference equation:

$$Z_{t+1} = Z_t + b_t \nu_t \Delta t + \sigma_t W_t. \tag{6.9}$$

In this expression, the value b_t — i.e. the permanent impact factor — is the rate of change of price as a function of the agent's trading rate at a given time; the value σ_t^2 describes the volatility of the process; and W_t is a sequence of independent Normal random variates with mean zero and variance Δt. The change in the agent's cash as a result of trading at the rate ν_t is then given by

$$X_{t+1} = X_t - \underbrace{(Z_t + k_t \nu_t)}_{\text{Execution Price}} \nu_t \Delta t, \tag{6.10}$$

where k_t is a temporary impact factor on price. This captures the notion that market orders incur a cost from having to walk the book, where k_t represents the premium paid for immediacy; note that in continuous-time, trading at ν_t is not strictly equivalent to trading with market orders in an LOB. Importantly, in this model, we assume that executions occur deterministically and that the market has sufficient liquidity to fulfil any order, though the price paid may be at a significant premium.

In the most general sense, one can think of the optimal execution problem as finding a "path of least resistance" between a starting inventory, Ω_0, and some target terminal inventory, Ω_T. In most cases the target inventory is taken to be zero, with $\Omega_T \doteq 0$. Optimal execution using this setup has two variants: liquidation and acquisition. The former refers to the case where $\Omega_0 > \Omega_T$, and the latter to $\Omega_0 < \Omega_T$. The strategy, ν_t, can be seen as the gradient of this path. But what is meant precisely by a path that has minimal resistance?

Consider the optimal liquidation setting and define the *implementation shortfall* of a strategy as $\Omega_0 Z_0 - X_T$. This quantity measures the difference between the cash generated through trading versus the mark-to-market value of the starting inventory. When this quantity is positive it implies that trading has incurred a cost, and when it is negative, that we have executed favourably. This idea translates to optimal acquisition by simply reversing the signs. Much of the literature is concerned with minimising this quantity, but there are other considerations. For example, a risk sensitive trader may desire a path that minimises exposure to the market, or heavily penalises any strategy that falls short of the target, even if it comes at a shortfall premium. To formalise these ideas, we define the following objective function as the condition for optimality in liquidation: $X_T - \Omega_T (Z_T - \eta_1 \Omega_T) - \eta_2 \sum_{t=0}^{T} \Omega_t^2$; where $\eta_{1,2}$ are constants. The first term here is the terminal cash generated through trading; the second term is a quadratic penalty on any existing inventory at the terminal timestep; and the final term is a running penalty on holding a non-zero inventory.

Translating the liquidation problem into a formulation suitable for RL follows trivially. First, define the action space as $\mathcal{A} \doteq \mathbb{R}_+$ with values representing the trading rate: $a_t \doteq -\nu_t$. Next, define the state space as $\mathcal{S} \doteq [0, 1]^2$, where states are given by the be tuples $s_t \doteq (t/T, \Omega_t/\Omega_0)$. The objective for a policy, π, can now be expressed in the form:

$$J(\pi) \doteq \mathbb{E}_{d_0,\pi}\left[X_T + \Omega_T (Z_T - \eta_1 \Omega_T) - \eta_2 \sum_{t=0}^{T} \Omega_t^2\right]. \tag{6.11}$$

We note that the term $\Omega_0 Z_0$ from the implementation shortfall does not appear in the objective. This is done without loss of generality since both Ω_0 and Z_0 are known a priori and act only as a constant; they have no bearing on the optimal solution. The corresponding reward function is then given by

$$r_t \doteq \Delta X_t - \eta_2 \Omega_t^2 + \Omega_T \left(Z_T - \eta_1 \Omega_T \right) 1_{t=T}, \tag{6.12}$$

This particular representation derives from Equation 6.10 and Equation 6.11, but it is by no means unique — there are many possible ways one can define reward. Indeed, in later sections we will see examples of reward formulations that exploit the mark-to-market portfolio value (Equation 3.1). For $\gamma = 1$, we require only that the sum of rewards equals the term inside the expectation of Equation 6.11.

Example

Consider an instance of the optimal liquidation problem with the following parameterisation: an initial inventory of $\Omega_0 \doteq 10$ and price of $Z_0 \doteq 100$ with time increments of $\Delta t \doteq 0.005$. The volatility of the price process is taken as $\sigma_t \doteq 2 \; \forall t$ with constant temporary and permanent impact factors $k_t \doteq 0.1 \; \forall t$ and $b_t \doteq 0.01 \; \forall t$, respectively. A terminal penalty of $\eta_1 \doteq 1.0$ and running penalty of $\eta_2 \doteq 0.1$ were applied to the objective function.

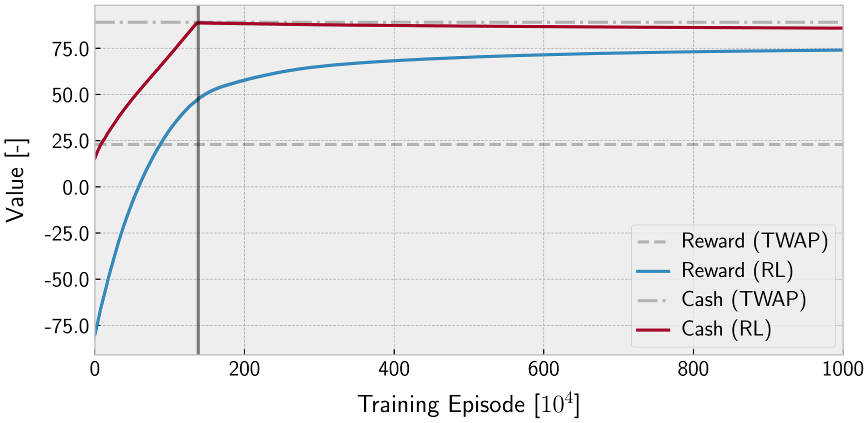
For a continuous problem such as this, we propose to learn a policy with the NAC-S(λ) algorithm using a folded Normal likelihood (Equation 6.4). The critic in this case was represented using a polynomial basis of 3rd order stacked over the policy's compatible features. Since it is known a priori that the problem is finite horizon, and that the time is included in s_t, we are free to choose a discount factor $\gamma \doteq 1$. The eligibility trace is constructed using an accumulating scheme with decay rate $\lambda \doteq 0.9$, and a learning rate of $\alpha_{\text{Critic}} \doteq 2 \times 10^{-5}$. The parameters of the policy itself were given by linear function approximators, also using 3rd order polynomial bases; all weights (including for the critic) were arbitrarily initialised with zeros. The policy gradient updates were applied every 20 steps with a learning of $\alpha_{\text{Policy}} \doteq 10^{-5}$.

The learning performance of the proposed methodology using a folded Normal policy is presented in Figure 6.4. We find that the policy improves in terms of both terminal cash and reward up until episode ~ 138. At this point, the total cash earnt matches that attained by a time-weighted average price execution strategy; this is illustrated only to aid comparison — we do not suggest that time-weighted average execution is a good strategy. From this point onwards the strategy begins to diverge from the benchmark, increasing reward at the cost of the expectation of cash. This corresponds to a more aggressive strategy that avoids holding large positions in order to minimise the impact of η_2, and ensuring that all inventory is liquidated by T to mitigate the terminal penalty.

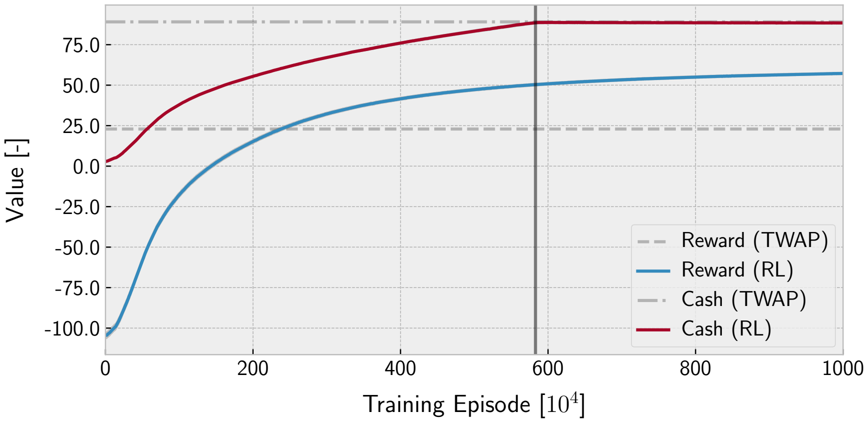
Interestingly, as claimed in Section 6.2, the use of a "natural" policy that respects the structure of $\mathcal{A}$ appears to yield improvements in learning efficiency. Figure 6.4 also shows the learning curve for a standard Gaussian policy in contrast with the folded Normal construction. Observe, in particular, the two vertical lines that highlight the points at which the two policies surpassed the cash benchmark. It took the Gaussian policy some ~ 400×10^4 episodes more to reach this level. It is plausible that this is caused by the boundary effects near zero due to improperly handled action clipping.

6.4 MARKET MAKING

We next consider market making, and propose to use the seminal model of Avellaneda and Stoikov [11] which has been studied by many others including Cartea, Donnelly,



(a) Folded Normal policy.



(b) Gaussian policy.

Figure 6.4: Learning performance of different policy classes on the optimal liquidation problem. The time-weighted average-price benchmark is illustrated in both cases alongside the terminal reward and cash attained by the strategies derived with RL. Each point corresponds to the sample mean over 1000 evaluation episodes with a (negligible) confidence window formed of the corrected, sample standard deviation.

and Jaimungal [31]. In this framework, the MM must trade a single asset for which the price, Z_t, evolves stochastically. In discrete-time,

$$Z_{t+1} = Z_t + \mu_t \Delta t + \sigma_t W_t, \tag{6.13}$$

where μ_t and σ_t are the *drift* and *volatility* coefficients, respectively. The randomness in this process comes from the sequence of independent, Normally-distributed random variables, W_t, each with mean zero and variance Δt; i.e. a random walk. The process begins with initial value Z_0 and continues until time T is reached.

The market maker interacts with the environment at each time by placing limit orders around Z_t to buy and sell a single unit of the asset. The prices at which the MM is willing to buy (bid) and sell (ask) are denoted by p_t^+ and p_t^-, respectively, and may be expressed as offsets from Z_t:

$$\delta_t^\pm = \pm[p_t^\pm - Z_t]; \tag{6.14}$$

these may be updated at each timestep at no cost to the agent. In general, $\delta_t^\pm \geqslant 0$ in order to ensure that the MM generates positive revenue from executing transactions, but this is not necessarily a hard constraint; see Section 3.3.2. Equivalently, we may define:

$$\psi_t \doteq \delta_t^+ + \delta_t^-, \tag{6.15}$$

and

$$\xi_t \doteq \frac{1}{2}\left(p_t^+ + p_t^-\right) - Z_t = \frac{1}{2}\left(\delta_t^+ - \delta_t^-\right), \tag{6.16}$$

called the *quoted spread* and *skew*, respectively. These relate to the MM's need for immediacy and bias in execution (i.e. $\nu^+ - \nu^-$) and have the advantage of greater interpretability.

In a given time increment, the probability that one or both of the agent's limit orders are executed depends on the liquidity in the market and the values $\delta_t^\pm$. Transactions occur when market orders arriving at random times have sufficient size to consume one of the agent's limit orders. In this case, these interactions, which are captured by $\nu_t^\pm$ in our nomenclature (Chapter 3), are modelled by independent Poisson processes with intensities

$$\lambda_t^\pm(x) \doteq A_t^\pm e^{-k_t^\pm x}, \tag{6.17}$$

where we define the shorthand notation $\lambda_t^\pm \doteq \lambda_t^\pm(\delta_t^\pm)$, and $A_t^\pm, k_t^\pm > 0$ to describe the *rate of arrival of market orders* and *distribution of volume in the book*, respectively. In the discrete setting, it follows that the probability of the agent's inventory changing in either direction is given by the values $\lambda_t^\pm$. This particular form derives from assumptions and observations on the structure and behaviour of limit order books which we omit here for brevity; see Avellaneda and Stoikov [11], Gould, Porter, Williams, McDonald, Fenn, and Howison [70], and Abergel, Anane, Chakraborti, Jedidi, and Toke [2] for more details. The dynamics of the agent's inventory process, or *holdings*, Ω_t (Definition 2), are then captured by the difference between these two point processes and Ω_0, which is known. The values of $\Omega_t \in \left[\underline{\Omega}, \overline{\Omega}\right]$ are also constrained such that that trading stops on the opposing side of the book when either limit is reached.

In this discrete-time setting, the evolution of the market maker's cash is given by the difference relation:

$$\begin{aligned} X_{t+1} &= X_t - p_t^+ \nu_t^+ + p_t^- \nu_t^-, \\ &= X_t + \delta_t^+ \nu_t^+ + \delta_t^- \nu_t^- - Z_t \nu_t. \end{aligned} \tag{6.18}$$

When expressed in terms of the controls $\delta_t^{\pm}$, the cash flow can be interpreted as a combination of the profit derived from charging the counterparty a premium of $\delta_t^{\pm}$, and the mark-to-market value of transacting the aggregate volume of ν_t. We can similarly derive a difference equation for the mark-to-market value of the agent's inventory. Noting the equivalence $\Omega_{t+1} Z_{t+1} = (\Omega_t + \nu_t)(Z_t + \Delta Z_t)$, a simple expansion yields the recursive process:

$$\Omega_{t+1} Z_{t+1} = \Omega_t Z_t + \Omega_t \Delta Z_t + \nu_t Z_t + \nu_t \Delta Z_t. \tag{6.19}$$

As introduced in Equation 3.1, the total (mark-to-market) value of the MM's portfolio may be expressed as $\Upsilon_t = X_t + \Omega_t Z_t$. Combining this with (6.18) and (6.19) above, it follows that the change in value of the MM's portfolio from $t \mapsto t+1$ is given by

$$\Delta \Upsilon_t = \underbrace{\delta_t^+ \nu_t^+ + \delta_t^- \nu_t^-}_{\text{Spread-PnL}} + \underbrace{\Omega_{t+1} \Delta Z_t}_{\text{Inventory-PnL}}. \tag{6.20}$$

This equation has a very similar form to the mark-to-market relation derived for the data-driven model in Section 5.6. Indeed, conceptually speaking they are indistinguishable, the only difference is in the spread term which no longer requires a correction for MOs.

Translating this framework into an MDP is accomplished by defining the state space $\mathcal{S} \in \mathbb{R}^2$ with $s_t \doteq (t, \Omega_t)$, and action space $\mathcal{A} \doteq \mathbb{R}^2$, where $a_t \doteq (\psi_t, \xi_t)$. For improved generalisation we may also apply a transformation of the state to $(t/T, \Omega_t^\dagger)$ for $\Omega^\dagger \doteq (\Omega_t - \underline{\Omega}_t)/(\overline{\Omega}_t - \underline{\Omega}_t)$. The objective is then defined to be

$$J(\pi) \doteq \mathbb{E}_{d_0, \pi}\left[\Upsilon_T - \eta_1 \Omega_T^2 - \eta_2 \sum_{t=0}^{T} \Omega_t^2\right], \tag{6.21}$$

which includes two penalty terms as in optimal liquidation problem. The first punishes strategies that reach T with a non-zero inventory, and the second disincentivises exposure to large inventory at each timestep. This is equivalent to having a reward function of the form:

$$r_t \doteq \Delta \Upsilon_t - \eta_2 \Omega_t^2 - \eta_1 \Omega_T^2 \mathbb{1}_{t=T}. \tag{6.22}$$

For studies using alternative definitions of reward see, e.g. Ganesh, Vadori, Xu, Zheng, Reddy, and Veloso [61] and Moody, Wu, Liao, and Saffell [116] or Part II.

In the special case that $\eta_1 = \eta_2 = 0$, this function yields a risk-neutral optimality criterion. Any non-zero penalty terms will give rise to risk-averse behaviour due to explicit sensitivity to inventory.

Remark. *One can quite legitimately view market making as a passive extension of the optimal execution problem. When $\eta_2 > 0$, the market maker is faced repeatedly with volatile execution targets whenever $|\Omega_t| > 0$. Similarly, when $\eta_1 > 0$ and T is finite, the agent must liquidate or acquire stock by the terminal time much in the same was as in Section 6.3. The additional challenge for a MM is that they must effectively interpolate between execution and profit generation simultaneously, whilst also quoting both ask and bid prices at all times.*

Example

Consider an instance of the market making problem with zero initial inventory, $\Omega_0 \doteq 0$, and prices evolving from $Z_0 \doteq 100$ with volatility $\sigma_t \doteq 2 \; \forall t$ and time increment $\Delta t \doteq 0.005$. The execution model is initialised with $A_t = 140 \; \forall t$ and $k_t = 1.5 \; \forall t$ — the same parameters as originally used by Avellaneda and Stoikov [11] — with constraints on trading to ensure that $\Omega_t \in [-50, 50]$ for all timesteps. To solve this MDP, we use the same fundamental algorithm as with the example in Section 6.3,

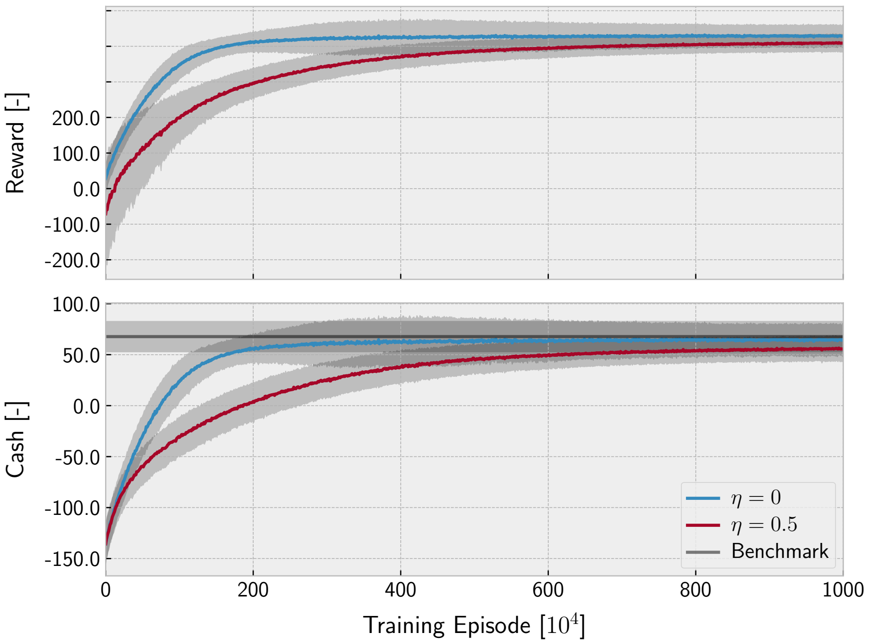


Figure 6.5: Learning performance of the NAC-S(λ) algorithm on the market making problem for $\eta \in \{0, 0.5\}$. The Avellaneda and Stoikov [11] solution (with $\gamma = 0.1$) is provided as a benchmark. Each point corresponds to the sample mean over 1000 evaluation episodes with a confidence window formed of the corrected, sample standard deviation.

but replace the policy likelihood with an isotropic, bivariate Gaussian distribution. Actions are then sampled as $(\psi, \xi) \sim \pi_\theta(\cdot \mid s)$ in any given state s. The parameters were kept mostly the same, save for reduced learning rates of $\alpha_{\text{Critic}} \doteq 2 \times 10^{-6}$ $\alpha_{\text{Policy}} \doteq 10^{-6}$.

Learning performance of the algorithm is illustrated in Figure 6.5 for two cases: $\eta_1 = 0$ (risk-neutral) and $\eta_1 = 0.5$ (risk-averse). Observe how the RL approach converges asymptotically to solutions that closely match the performance of the optimal strategy derived by Avellaneda and Stoikov [11] for exponential utilities; the small discrepancy is most likely caused by the stochastic nature of the policies and the subtle difference in objectives. As expected, the penalised strategy (i.e. for $\eta_1 > 0$) yields a lower variance on terminal wealth, but does so at the cost of a (very) small reduction in the mean.

6.5 PORTFOLIO OPTIMISATION

For portfolio optimisation, we consider an extended version of the model first proposed by Tamar, Di Castro, and Mannor [170]. This setup also features in the subsequent works of Bisi, Sabbioni, Vittori, Papini, and Restelli [22] and Spooner and Savani [159] (see Chapter 8), which present different perspectives on risk-sensitivity, and has become a standard test-bed for risk-sensitive RL. While simple, it highlights some of the key pathologies of portfolio optimisation that carry over to more realistic instances of the problem; such as the need for risk-sensitivity.

In this setting, the agent's portfolio consists of two types of asset: (i) a liquid asset such as cash holdings with a growth rate g^L; and (ii) an illiquid asset with time-dependent interest rate $g^I_t \in \{\overline{g}^I, \underline{g}^I\}$ that switches between two values stochastically; e.g. options. Unlike the original formulation of Tamar, Di Castro, and Mannor [170], we do not assume that g^I_t switches symmetrically. Instead, for added complexity, the illiquid growth rate is treated as a switching process with two states and fixed transition probabilities of $p_\uparrow$ and $p_\downarrow$. The sequence of values it takes thus forms a stochastic process:

$$g^I_{t+1} \doteq \begin{cases} \overline{g}^I & \text{w.p. } p_\uparrow \text{ if } g^I_t = \underline{g}^I, \\ \underline{g}^I & \text{w.p. } p_\downarrow \text{ if } g^I_t = \overline{g}^I, \\ g^I_t & \text{otherwise.} \end{cases} \tag{6.23}$$

At each timestep the agent chooses an amount, up to M, of the illiquid asset to purchase at a fixed cost per unit, c. At maturity, after N steps, this illiquid asset either defaults, with probability p_D, or is sold and converted into cash. We denote by $\Omega^{(i)}_t$ the investment in the illiquid asset with maturity in i timesteps; i.e. $\Omega^{(2)}_t$ will mature in 2 steps. By this definition we are treating each instance of the illiquid asset as unique depending on the maturity horizon. The evolution of this process, for $1 \leqslant i < N$, is thus given by

For options, this would also involve liquidating (or repurchasing) the underlying and converting the investment into cash.

$$\Omega^{(i)}_{t+1} \doteq \left(1 + g^I_t\right) \Omega^{(i+1)}_t,$$

and $\Omega^{(N)}_{t+1} \doteq a_t$ for $i = N$. At any given time, the total investment at risk is given by the sum $\sum_{i=1}^N \Omega^{(i)}_t$. The cash process then updates stochastically according to a liquid growth rate and the outcome of the Bernoulli random variable governing defaults:

$$X_{t+1} = \left(1 + g^L\right) X_t - c a_t + \begin{cases} \Omega^{(1)}_t & \text{w.p. } 1 - p_D, \\ 0 & \text{w.p. } p_D. \end{cases} \tag{6.24}$$

The problem specification above can now be cast explicitly into an MDP. The state space of the problem is embedded in $\mathcal{S} \doteq \mathbb{R}^{N+2}$, where the first entry denotes the allocation in the liquid asset, the next N are the allocations in the non-liquid asset class, and the final entry takes the value $g^\dagger_t \doteq g^I_t - \mathbb{E}_{t'<t}[g^I_{t'}]$. This last term determines which interest regime is active in the MDP in terms of the excess away from the expected value up to the current time t. This allows for better generalisation across different parameterisations of the problem while remaining an adapted process. Analytically, this is defined as

$$s_t \doteq \left[X_t, \boldsymbol{\Omega}^\top_t, g^\dagger_t\right]^\top. \tag{6.25}$$

The actions are then given by the M discrete choices over possible purchase orders, and the reward at each timestep is given by the log-return in the liquid asset in the interval $t \mapsto t+1$,

$$r_t \doteq \ln\left(X_{t+1}\right) - \ln\left(X_t\right), \tag{6.26}$$

as chosen by Bisi, Sabbioni, Vittori, Papini, and Restelli [22].

Example

As an example, let there be an instance of the portfolio problem where $g^L \doteq 0.005$, $\underline{g}^I \doteq 0.05$ and $\overline{g}^I \doteq 0.25$; with initial value $g^I_0 \doteq \underline{g}^I$. The illiquid growth rate then switches with probabilities $p_\downarrow \doteq 0.6$ and $p_\uparrow \doteq 0.1$. We will allow the agent to purchase up to $M \doteq 10$ units of the asset at a cost of $c \doteq 0.2/M$ each; this gives rise

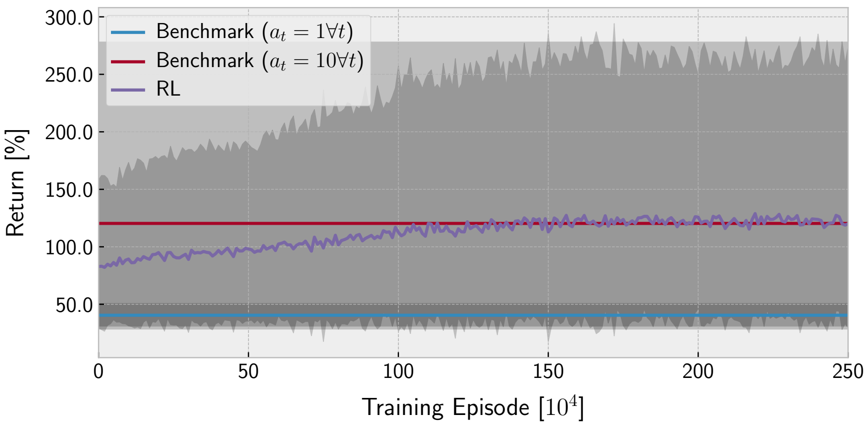


Figure 6.6: Learning performance of the NAC-S(λ) algorithm on the portfolio optimisation problem. Fixed passive and aggressive strategies are provided as a benchmark. Each point corresponds to the sample mean over 1000 evaluation episodes with a confidence window formed of the corrected, sample standard deviation.

to an action space of cardinality 11. The maturity horizon is set to $N = 4$ with a default probability of $p_D \doteq 0.1$. Each episode of the problem terminates at $T = 50$ with probability 1.

In this example, for the sake of variety, we make use of a different learning algorithm: traditional actor-critic using the TD-error as an estimate of the advantage function [143]; see Chapter 2. This is a highly standard approach and can often be seen in combination with deep function approximation in the literature. Here, we use a linear basis (i.e. a polynomial of order 1) over the state space for $\widehat{f}_\theta(s, a)$ in the policy, with repeated and independent weights for each action:

$$\boldsymbol{\phi}(s, a) \doteq \left[\boldsymbol{\phi}(s)^\top \circ \mathbf{1}^\top_{a=1}, \ldots, \boldsymbol{\phi}(s)^\top \circ \mathbf{1}^\top_{a=|\mathcal{A}|}\right]^\top, \tag{6.27}$$

where $\circ$ denotes the Hadamard product, and the state-dependent basis is given by $\boldsymbol{\phi}(s) \doteq [1, s_1, s_2, \ldots, s_N]$. Updates for the policy were then performed using a learning rate of $\alpha_{\text{Policy}} \doteq 5 \times 10^{-4}$. The value function approximator used in compute the TD-error was then defined as $\widehat{V}_\nu(s) \doteq \langle \nu, \boldsymbol{\phi}(s) \rangle$. This estimator was learnt in tandem with the policy using the iLSTD algorithm of Geramifard, Bowling, and Sutton [63], a learning rate of $\alpha_{\text{Critic}} \doteq 2 \times 10^{-6}$, and a discount factor of $\gamma = 1$; see Section 2.3.3 for details on least-squares methods.

The performance of this algorithm during training is illustrated in Figure 6.6 in terms of the return on investment. We find that the policy converges on a highly aggressive strategy that buys the maximum quantity of the asset at each timestep. Indeed, Figure 6.6 shows clearly that the agent achieves a mean and standard deviation of financial returns that are equal to the simple benchmark where $a_t \doteq 10 \forall t$. This suggests that the optimal strategy under the risk-neutral reward function (Equation 6.26) is simple. In Chapter 8 we extend these results to show that one can account for the downside risk of trading yielding a risk-sensitive strategy.

6.6 OPTIMAL CONSUMPTION

Our final problem setting is known as Merton's optimal consumption problem [112] and it is a special case of intertemporal portfolio optimisation. This domain is closely related to the previous example, but now the agent must also *consume* it's cash, X_t, in a optimal manner while simultaneously managing it's investment portfolio. This particular case has been largely unstudied in the RL literature in spite of the fact that it represents a broad class of real-world problems, such as retirement planning. As before, the agent's portfolio consists of two assets. The first is an illiquid, risky asset whose price, Z_t, evolves stochastically according to a discrete-time analogue of an Itô diffusion:

To the best of our knowledge, the work of Weissensteiner [182] is the only prior example.

$$Z_{t+1} = Z_t + \mu_t \Delta t + \sigma_t W_t, \tag{6.28}$$

where W_t is random walk with zero mean and variance of Δt. The second is a liquid (riskless) asset, cash, whose price grows at a fixed rate of $1 + g^L$. Here, however, the agent must specify two controls: (i) the proportion of it's wealth to invest between the risky and riskless assets; and (ii) an amount of it's cash to consume and permanently remove from the portfolio. The former quantity is expressed by the fraction

$$\Xi_t \doteq \frac{\Omega_t}{Z_t X_t + \Omega_t}, \tag{6.29}$$

and the latter we denote by C_t. The total value of the agent's portfolio can thus be expressed by the following difference equation:

$$\Upsilon_{t+1} \doteq \Upsilon_t + \left[\Omega_t Z_{t+1} + X_t \left(1 + g^L\right)\right] \Delta t - C_t. \tag{6.30}$$

The problem terminates when all the agent's wealth is consumed (i.e. when $\Upsilon_t = 0$) or the terminal timestep is reached. In the latter case, any remaining wealth that wasn't consumed is lost.

To highlight downside risk within the domain, we also extend the traditional model above to include the possibility of defaults. At each decision point there is a non-zero probability p_D that the risky asset's underlying “disappears”, the risky investment is lost, the problem terminates, and the remaining wealth in the liquid asset is consumed in it's entirety. It is precisely this setting that we study in detail in Chapter 8.

Similar variations on the original problem have been considered in the literature by, e.g. Puopolo [133].

This problem, both for $p_D = 0$ and $p_D > 0$, can once again be formulated as an MDP. First, observe that the state space of the problem is given by $\mathcal{S} \doteq \mathbb{R}^2_+$, where each state is given by the current time and the agent's remaining wealth: $s_t \doteq (t/T, X_t/X_0)$. The action space is defined as $\mathcal{A}(s_t) = [0, 1] \times [0, X_t]$, yielding the two controls $a_t \doteq (\Xi_t, C_t/X_t)$. The reward is then defined as the amount of cash consumed at each timestep: $r_t \doteq C_t/X_t$. In principle, one could instead use negative log-returns of $\{X_t\}$ as in the portfolio optimisation setting, but as long as the total is bounded we are free to make this choice.

7 INVARIANCE TO EPISTEMIC RISK

7.1 OUTLINE

Market making behaviour is synonymous with the act of providing liquidity to a market. This notion was introduced in Chapter 3 and studied in depth in Part II, so it is well established that an MM achieves this by continuously quoting prices to buy and sell an asset. The goal of an MM is thus to repeatedly earn the quoted spread by transacting in both directions. Of course, they cannot do this without exposing themselves to risk in the form of *adverse selection*. This derives from "toxic" agents exploiting technological and/or informational advantages, transacting with the MM such that its inventory is exposed to adverse price moves. This phenomenon, known ubiquitously as *inventory risk*, has been the subject of a great deal of research in optimal control, artificial intelligence and reinforcement learning (RL) literature; including the work presented in Part II. Yet, there are also other types of risk that are epistemic in nature.

A standard assumption in past work has been to suppose that the MM has perfect knowledge of market conditions, but this is clearly not the case. One must estimate a model's parameters from data, and in most cases (though, we posit all), any parametric model is an incomplete representation of market dynamics. Indeed, these are precisely the criticisms made in Part II of model-driven methods. Robustness to this type of model ambiguity has only recently received attention, with Cartea, Donnelly, and Jaimungal [31] extending optimal control approaches for the market making problem to address the risk of model misspecification. This chapter deals with the same type of epistemic risk, but taking a game theoretic approach. Rather than treat the environment as a static construct, we allow the parameters of the model to vary both inter- and intra-episode. We then train market making agents that are robust to strategically chosen market conditions through the use of adversarial RL.

The starting point of this work is the well-known single-agent mathematical model of market making of Avellaneda and Stoikov [11], which has been used extensively in quantitative finance [31, 36, 77, 78], and was specified in Chapter 6. Now the model is extended to introduce a "market player", the adversary, that can be thought of as a proxy for other market participants that would like to profit at the expense of the MM. The adversary controls the dynamics of the market environment — i.e. the values of b_t, $A_t^{\pm}$ and $k_t^{\pm}$ (see Section 6.4) — in a zero-sum game against the market maker. These parameters govern price and execution dynamics, and would naturally be expected to vary over time in real markets; or indeed react adversarially. We thus go beyond the fixed parametrisation of existing models — henceforth called the **Fixed** setting — with two extended learning scenarios:

RANDOMISED Episodes are initialised with an instance of the model whose parameters have been chosen independently and uniformly at random from the support.

STRATEGIC A setting in which the MM competes against an independent "market" learner whose objective is to *select parameters* from the support so as to *minimise* the performance of the market maker in a zero-sum game.

The Randomised and Strategic settings are, on the one hand, more realistic than the Fixed setting, but on the other hand, significantly more complex for the market making agent to learn in. In the following we show that market making strategies trained in each of these settings yield significantly different behaviour, and demonstrate striking benefits of our proposed Strategic learning scenario.

Contributions

The key contributions of this chapter are as follows:

(i) Introduce a game-theoretic adaptation of a standard mathematical model of market making. The adapted model is shown to be useful for training and evaluating MM strategies that are robust to epistemic risk (Section 7.3).

(ii) Propose an algorithm for adversarial RL in the spirit of RARL [130], and demonstrate its effectiveness in spite of the well known challenges associated with finding the Nash equilibria of Markov games [100] (Section 7.4 and Section 7.5).

(iii) Investigate the impact of three environmental settings (one adversarial) on learning market making. We show that training against a Strategic adversary *strictly dominates* the other two settings (Fixed and Randomised) in terms of a set of standard desiderata, including the Sharpe ratio (Section 7.5).

(iv) Prove that, in several key instances of the Strategic setting, the single-stage instantiation of our game has a Nash equilibrium resembling that found by adversarial training in the multi-stage game. We then confirm broader existence of (approximate) equilibria in the multi-stage game by empirical best response computations (Section 7.3 and Section 7.5).

7.2 RELATED WORK

OPTIMAL CONTROL AND MARKET MAKING. The theoretical study of market making originated from the pioneering work of Ho and Stoll [84], Glosten and Milgrom [66] and Grossman and Miller [75], among others. Subsequent work focused on characterising optimal behaviour under different market dynamics and contexts. Most relevant is the work of Avellaneda and Stoikov [11], who incorporated new insights into the dynamics of the limit order book to give a new market model, which is the one used in this chapter. They derived closed-form expressions for the optimal strategy of an MM with an exponential utility function when the MM has perfect knowledge of the model and its parameters. This same problem was then studied for other utility functions, including linear and quasilinear utilities [33, 36, 59, 77, 78]. As mentioned above, Cartea, Donnelly, and Jaimungal [31] study the impact of uncertainty in the model of Avellaneda and Stoikov [11]: they drop the assumption of perfect knowledge of market dynamics, and consider how an MM should optimally trade while being robust to possible misspecification. This type of *epistemic risk* is the primary focus of our chapter.

MACHINE LEARNING AND MARKET MAKING. Several papers have applied AI techniques to design automated market makers for financial markets. Chan and Shelton [40] focussed on the impact of noise from uninformed traders on the quoting behaviour of a market maker trained with reinforcement learning. Abernethy and Kale [3] used an *online learning* approach. More recently, Guéant and Manziuk [79] addressed scaling issues of finite difference approaches for high-dimensional, multi-asset market making using model-based RL. While the approach taken in this paper is also based on RL, unlike the majority of these works, our underlying market model is taken from the mathematical finance literature. There, models are typically analysed using methods from optimal control. To the best of our knowledge, we are the first to apply adversarial reinforcement learning (ARL) to derive trading strategies that are robust to epistemic risk.

A separate strand of work in AI and economics and computation has studied automated market makers for prediction markets, see the thesis of Othman [121] for example. While some similarities to the financial market making problem pertain, the focus in that strand of work focusses much more on price discovery and information aggregation.

RISK-SENSITIVE REINFORCEMENT LEARNING. Risk-sensitivity and safety in RL has been a highly active topic for some time. This is especially true in robotics where exploration is very costly. For example, Tamar, Di Castro, and Mannor [170] studied policy search in the presence of variance-based risk criteria, and Bellemare, Dabney, and Munos [15] presented a technique for learning the full distribution of (discounted) returns; see also Garcıa and Fernández [62]. These techniques are powerful, but can be complex to implement and can suffer from numerical instability. This is especially true when using exponential utility functions which, without careful consideration, may diverge early in training due to large negative rewards [103]. An alternative approach is to train agents in an adversarial setting [127, 130] in the form of a zero-sum game. These methods tackle the problem of epistemic risk by explicitly accounting for the misspecification between train- and test-time simulations. This robustness to test conditions and adversarial disturbances is especially relevant in financial problems and motivated the approach taken in this paper.

The problem of robustness has also been studied outside of the use of adversarial learning; see, e.g., [135].

7.3 TRADING GAMES

The market dynamics defined in Section 6.4 give rise to a zero-sum Markov game between the market maker (MM) and an adversary that acts as a proxy for all other market participants, and controls the parameters of the model. This construction forms the basis for the results presented in this chapter.

Definition 17 (Market Making Game). *The (undiscounted) Markov game between the MM and an adversary has T stages. At each stage, the MM chooses $\delta_t^{\pm}$ and the adversary $\{b_t, A_t^{\pm}, k_t^{\pm}\}$ based on the state s_t with transition dynamics as defined in Section 6.4. The resulting stage payoff is given by expected change in MtM value of the MM's portfolio: $\mathbb{E}[\Upsilon_T - \Upsilon_0]$. The total payoff paid by the adversary to the MM is the sum of the stage payoffs.*

An illustration of the market making game for $t \in \{0, 1, 2, \ldots\}$ is given in Figure 7.1, where each node depicts a state and each edge a possible transition. From this we can see clearly that the game has a (trinomial) tree structure due to the three possible innovations of the inventory of the MM: increase/decrease by one, or remain the same. Note, however, that the trinomiality only occurs while $\Omega \in (\underline{\Omega}, \overline{\Omega})$. When either the upper/lower boundary is reached, there are only two possible branches: remain the same, move away from the limit by one.

This concept often arises naturally in option pricing, so it's not unsurprising it showed up here.

7.3.1 *Single-Stage Analysis*

Consider an instance of the market making game (Definition 17) with $\mathsf{T} = 1$ — i.e. a single-stage variant of the game. This setting has the property of being stateless and zero-sum since both t and Ω_0 are given. As such, the payoff reduces to $\mathbb{E}[\Delta\Upsilon_0]$

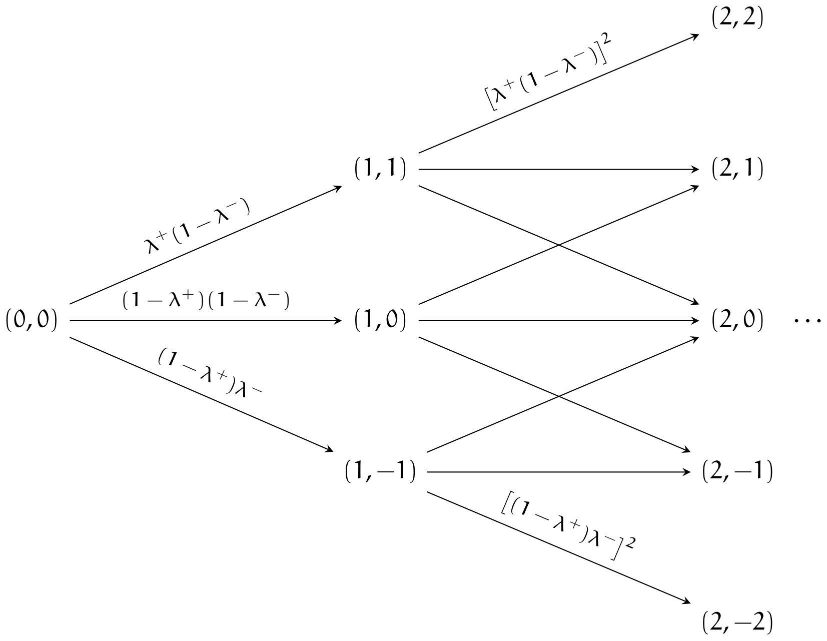


Figure 7.1: Trinomial tree of the multi-stage Market Making game with initial inventory of $\Omega_0 = 0$. Three stages of the game are depicted (for $t \in \{0, 1, 2\}$) with state-transition probabilities annotated along the edges.

which can be expanded into $\mathbb{E}[\Upsilon_1 - \Upsilon_0] = f(\delta^{\pm}; b, A^{\pm}, k^{\pm})$, where the function over strategy profiles is defined as

$$f(\delta^{\pm}; b, A^{\pm}, k^{\pm}) \doteq \underbrace{A^{+}(\delta^{+} - b)e^{-k^{+}\delta^{+}} + A^{-}(\delta^{-} + b)e^{-k^{-}\delta^{-}}}_{\text{Spread PnL}} + \underbrace{b\Omega.}_{\text{Inventory PnL}} \tag{7.1}$$

This expression is simply an expectation over the transition probabilities of the first three edges between $t = 0$ to $t = 1$ in Figure 7.1. As in the previous chapters, we recover the characteristic breakdown into spread for ask and bid, and the inventory PnL. Now, note that for certain parameter ranges, this function $f(\cdot)$ is concave in $\delta^{\pm}$ for which we introduce the following lemma.

Lemma 1 (Payoff Concavity in $\delta^{\pm}$)**.** *The payoff function (Equation 7.1) is a concave function of* $\delta^{\pm}$ *for* $\delta^{\pm} \in \left[0, \frac{2}{k^{\pm}} \mp b\right]$*, and strictly concave for* $\delta^{\pm} \in \left[0, \frac{2}{k^{\pm}} \mp b\right)$.

Proof. The first derivative of the payoff function (Equation 7.1) w.r.t. $\delta^{\pm}$ may be derived as follows:

$$\begin{aligned} \frac{\partial f}{\partial \delta^{\pm}} &= \frac{\partial}{\partial \delta^{\pm}} \lambda^{\pm}\left(\delta^{\pm} \mp b\right), \\ &= \lambda^{\pm} + \left(\delta^{\pm} \mp b\right)\frac{\partial \lambda^{\pm}}{\partial \delta^{\pm}}, && \text{by the chain rule} \\ &= \lambda^{\pm}\left[1 + k^{\pm}(\delta^{\pm} \pm b)\right]. && (7.2) \end{aligned}$$

Noting that the cross-derivatives go to zero, and then repeating a similar process to above then admits the diagonal Hessian matrix

$$\mathbf{H} = \begin{bmatrix} k^{+}\lambda^{+}\left[k^{+}\left(\delta^{+} - b\right) - 2\right] & 0 \\ 0 & k^{-}\lambda^{-}\left[k^{-}\left(\delta^{-} + b\right) - 2\right] \end{bmatrix}. \tag{7.3}$$

A sufficient condition for Equation 7.1 to be concave is that $\mathbf{H}$ is negative semi-definite, which is satisfied for $\delta^{\pm} \in \left[0, \frac{2}{k^{\pm}} \mp b\right]$. For strict concavity we require that $\mathbf{H}$ is negative definite, which is satisfied by extension for $\delta^{\pm} \in \left[0, \frac{2}{k^{\pm}} \mp b\right)$. This concludes the proof. ∎

It is clear also that this payoff function is linear in both b and $A^{\pm}$, implying that Equation 7.1 is both concave and convex with respect to these four variables. From this one can show that there exists an Nash equilibrium (NE) when $\delta^{\pm}$ and b are controlled strategically, and both $A^{\pm}$ and $k^{\pm}$ are fixed (i.e. constants of the game).

Theorem 3 (NE for fixed $A^{\pm}, k^{\pm}$)**.** *There is a pure strategy Nash equilibrium* $(\delta^{\pm}_{\star}, b_{\star})$ *for* $(\delta^{+}, \delta^{-}) \in [0, \frac{2}{k^{+}} - b] \times [0, \frac{2}{k^{-}} + b]$ *and* $b \in [\underline{b}, \overline{b}]$ *(with finite* $\underline{b}, \overline{b}$*),*

$$\delta^{\pm}_{\star} = \frac{1}{k^{\pm}} \pm b_{\star}; \quad b_{\star} = \begin{cases} \underline{b} & \Omega > 0, \\ \overline{b} & \Omega < 0, \end{cases} \tag{7.4}$$

which is unique for $|\Omega| > 0$*. When* $\Omega = 0$*, there is an equilibrium for every value* $b_{\star} \in [\underline{b}, \overline{b}]$.

Proof. Lemma 1 provides the conditions for which the payoff is a quasi-concave function: the intervals $\delta^{\pm} \in \left[0, \frac{2}{k^{\pm}} \mp b\right]$. This implies that any stationary points will be maxima and thus equating Equation 7.2 to zero gives the two possible solutions:

$$\lambda^{\pm} = 0, \quad \text{or} \quad k^{\pm}\left(\delta^{\pm} \pm b\right) = 1.$$

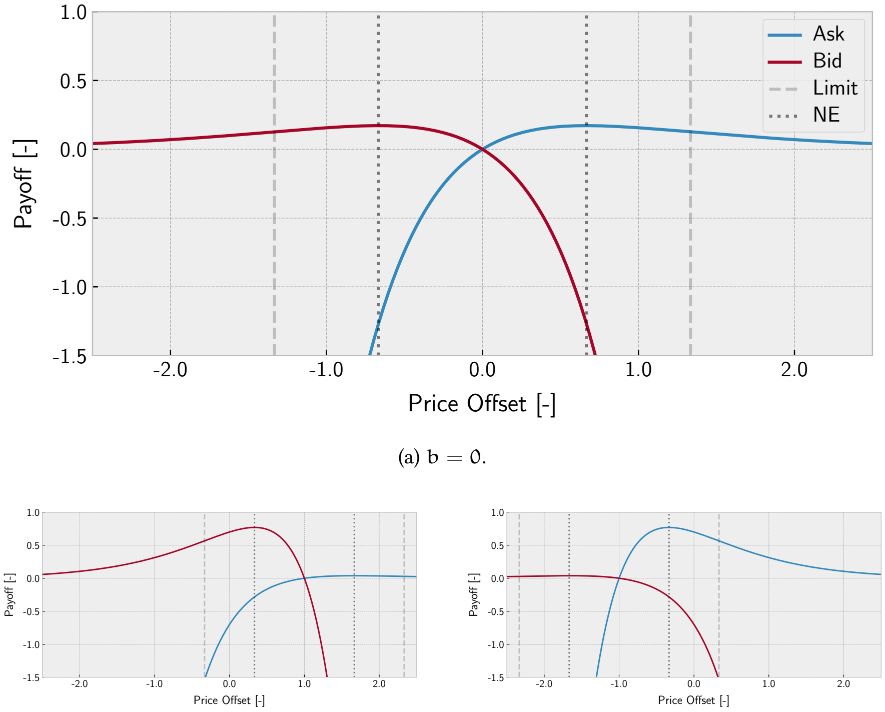
The former is ruled out immediately since the exponential function is strictly positive on the concave intervals; it is also clearly a minimum and only occurs in the limit as $\delta^{\pm} \to \infty$. It follows that the maximum is achieved at the values in Equation 7.4.

To prove that these correspond to a pure strategy NE of the game we show that the payoff is quasi-concave (resp. quasi-convex) in the MM's (resp. adversary's) strategy and then apply Sion's minimax theorem [154]. This is satisfied for the MM by Lemma 1, and by the linearity of the adversary's payoff with respect to b. As a result, there is a unique solution for $|\Omega| > 0$, and when $\Omega = 0$, there exists a continuum of solutions, all with equal payoff. ∎

The solution given in Equation 7.4 has a similar form to that of the optimal strategy under a linear utility with terminal inventory penalty [59], or equivalently that of a myopic agent with running penalty [33]. Figure 7.2 provides some intuition into the nature of the MM strategy for three fixed values of b. The (restricted) concavity — and thus uniqueness of the NE — as well as the impact of the adversary's attack is clear from this diagram. The former follows from uni-modality of a function with real codomain, and the latter from the skewing effect seen for $b \in \{-1, 1\}$. It can also be shown that the extension of Theorem 3 to an adversary with control over all five model parameters $\{b, A^{\pm}, k^{\pm}\}$ yields a similar result.

Theorem 4 (NE for general case)**.** *Take the game in Theorem 3 and add* $A^{\pm} \in \left[\underline{A}, \overline{A}\right]$ *and* $k^{\pm} \in \left[\underline{k}, \overline{k}\right]$ *to the adversary's strategy profile (with finite bounds). In this extension, there exists a pure strategy Nash equilibrium* $(\delta^{\pm}_{\star}, \{b_{\star}, A^{\pm}_{\star}, k^{\pm}_{\star}\})$ *of the MM game for* (7.4)*,* $A^{\pm}_{\star} = \underline{A}$ *and* $k^{\pm}_{\star} = \overline{k}$*. For* $|\Omega| > 0$ *this equilibrium is unique, and for* $\Omega = 0$ *there exists an equilibrium for every value* $b_{\star} \in [\underline{b}, \overline{b}]$.

Proof. For $(\delta^{\pm}_{\star}, \{b_{\star}, A^{\pm}_{\star}, k^{\pm}_{\star}\})$ to be an NE, both $\delta^{\pm}_{\star}$ and $\{b_{\star}, A^{\pm}_{\star}, k^{\pm}_{\star}\}$ must be best responses with respect to the opposing strategy in the profile. For the MM, we have from Theorem 3 that, for any fixed strategy played by the adversary, the optimal choice is given by $\delta^{\pm}_{\star}$ in Equation 7.4. For the adversary, we must show that the



(a) $b = 0$.

(b) $b = -1$.

(c) $b = 1$.

Figure 7.2: The MM's payoff (Equation 7.1) as a function of the price offsets for the ask, δ^+, and bid, $-\delta^-$ sides of the book. Each sub-figure corresponds to one of three values of b. The concave intervals (as derived in Lemma 1) are illustrated by the dashed line, and the optimal solution for the MM by the dotted line.

strategy $\{b_\star, A_\star^\pm, k_\star^\pm\}$ yields a payoff at least as high as any other strategy $\{b, A^\pm, k^\pm\}$. Concretely, we must show that the inequality

$$\underline{A}\left(\delta^+ + b_\star\right) e^{-\overline{k}\delta^+} + \underline{A}\left(\delta^- - b_\star\right) + e^{-\overline{k}\delta^-} + b_\star\Omega \leqslant \\ A^+\left(\delta^+ + b\right) e^{-k^+\delta^+} + A^-\left(\delta^- - b\right) e^{-k^-\delta^-} + b\Omega$$

holds for all $b \in [\underline{b}, \overline{b}]$, $A^\pm \in [\underline{A}, \overline{A}]$ and $k^\pm \in [\underline{k}, \overline{k}]$. A sufficient condition for this is that

$$b_\star\Omega \leqslant b\Omega, \tag{7.5}$$

and

$$\underline{A}(\delta^\pm \pm b_\star)e^{-\overline{k}\delta^\pm} \leqslant A^\pm(\delta^\pm \pm b)e^{-k^\pm\delta^\pm} \tag{7.6}$$

are *both* true.

The first of these requirements is always satisfied, since: $b_\star = \underline{b} \leqslant b$ for $\Omega > 0$ and $b_\star = \overline{b} \geqslant b$ for $\Omega < 0$; and clearly $0 \leqslant 0$. The same is also true of (7.6), since $\underline{A} \leqslant A^\pm$, $e^{-\overline{k}} \leqslant e^{-k^\pm}$ and $\delta^\pm \pm b_\star \leqslant \delta^\pm \pm b$ are satisfied for all $\Omega \in [\underline{\Omega}, \overline{\Omega}]$. We thus have that no unilateral deviation from the strategy profile $(\delta_\star^\pm, \{b_\star, A_\star^\pm, k_\star^\pm\})$ can yield a higher payoff for either player, and thus the profile constitutes an NE.

For uniqueness, we inherit the claims of Theorem 3 and need only show that $A_\star^\pm$ and $k_\star^\pm$ are unique. Inspecting (7.6), we can see that the inequality is strict for all $A^\pm \in (\underline{A}, \overline{A}]$ and $k^\pm \in [\underline{k}, \overline{k})$. Thus, the strategy $\{b_\star, A_\star^\pm, k_\star^\pm\}$ is unique and the proof is complete. ■

The uniqueness of the NE derived in Theorem 4 also tells us about the nature of the solution. In particular, we can show that the NE is stable to perturbations in the strategy profile.

Corollary 4.1. *The pure strategy Nash equilibrium prescribed in Theorem 4 for $|\Omega| > 0$ is stable to perturbations in the profile itself. The equilibria for $\Omega = 0$ are not.*

Proof. Stability of an NE is satisfied if, for a small perturbation of a player's profile, the following both hold: (i) the perturbed player achieves a strictly lower payoff; and (ii) the opposing player's best response is unchanged. By this definition, there is no unique solution for the adversary for $\Omega = 0$, and thus the first requirement is not met. For $|\Omega| > 0$, the conditions are met due to the concavity/convexity of the payoff (Lemma 1) and the uniqueness of the NE strategy profile, respectively, concluding the proof. ■

This has an interesting interpretation in the context of RL. That is, if a learning method successfully converges, approximately, to the NE, then small exploratory actions about this point will not lead to divergence in terms of the equilibrium. This is important since convergence is typically guaranteed only under the assumption that the policy explores sufficiently, even in the limit.

7.3.2 *Multi-Stage Analysis*

Consider another special case of the market maker (MM) game in which $T = 2$ (see Figure 7.1) and the total expected payoff for the MM is given by:

$$\begin{aligned} f\left(\delta^{\pm}; \mathbf{b}, \mathbf{A}^{\pm}, \mathbf{k}^{\pm}\right) &= b_0\Omega_0 + b_1\mathbb{E}\left[\Omega_1\right] + \sum_{t=0}^{1} \lambda_t^{+}(\delta_t^{+} - b_t) + \lambda^{-}(\delta_t^{-} + b_t), \\ &= \underbrace{b_0\Omega_0 + b_1\left[\Omega_0 + \lambda_0^{-}(1-\lambda_0^{+}) - \lambda_0^{+}(1-\lambda_0^{-})\right]}_{\text{Inventory PnL}} \\ &\quad + \underbrace{\sum_{t=0}^{1} \lambda_t^{+}(\delta_t^{+} - b_t) + \lambda^{-}(\delta_t^{-} + b_t)\,.}_{\text{Spread PnL}} \end{aligned} \tag{7.7}$$

Here we see for the first time how spread PnL depends only on the time t, and is thus *independent of the inventory* Ω_t *or it's future values.* The optimal choice — ignoring momentarily the inventory PnL — thus depends only on the adversary's strategy and can be solved at each timestep entirely myopically. Multi-step planning is only required because of the inventory term which couples the strategy played at time t with the state and strategy played at subsequent timesteps $t' > t$. This speaks to the intuition around inventory risk and why adverse selection is an important topic in the study of market making.

In terms of equilibria, we can clearly deduce from the single-state analysis that there exists at least one subgame perfect NE in the full multi-stage game (i.e. for any T) whereby both players simply play (7.4) at each stage. This follows from standard backwards induction arguments. Enumerating all possible NE for a Markov game with large T, however, is computationally hard and numerical methods only have guarantees for verifying their existence and reachability in special cases; let alone assert with any confidence what equilibria may exist at all. We leave it as interesting future work to expand on these theoretical results and whether any guarantees can be established for the multi-stage setting.

7.4 ADVERSARIAL TRAINING

As outlined in the previous section, a single-stage analysis is informative but unrealistic. On the other hand, the multi-stage analysis quickly becomes non-trivial to probe analytically and the results that can be derived are not particularly deep or insightful. In this section we investigate a range of multi-stage settings with different restrictions on the adversary, and explore how adversarial training (a numerical method) can be used to find market making strategies that are robust to strategic attacks in the environment dynamics. In so doing, we will show that adversarial training is an effective method for deriving trading strategies that are epistemically robust.

First, let us define the following three environments (Markov games), each of which, in turn, afford increasing freedom to the adversary to control the market's dynamics:

Definition 18 (Fixed adversary)**.** *The simplest possible adversary always plays the same fixed strategy:* $b_t = 0$, $A_t^{\pm} = 140$ *and* $k_t^{\pm} = 1.5$ *for all times* $t \in [0, T)$*; these values match those originally used by Avellaneda and Stoikov [11]. This amounts to a* single-agent learning setting with stationary transition dynamics.

Definition 19 (Random adversary)**.** *The second type of adversary instantiates each episode with parameters chosen independently and uniformly at random from the* *ranges:* $b_t = b \in [-5, 5]$, $A_t^{\pm} = A \in [105, 175]$ *and* $k_t^{\pm} = k \in [1.125, 1.875]$*; note that these intervals are centred on the values for the fixed adversary with diameters driven by experimentation. These are chosen at the start of each episode and remain fixed until the terminal (actionable) timestep,* $T - 1$*. This is analogous to* single-agent RL with non-stationary transition dynamics.

Definition 20 (Strategic adversary)**.** *The final type of adversary chooses the model parameters* $\{b_t, A_t^{\pm}, k_t^{\pm}\}$ *(bounded as in Definition 19) at* each step of the game*. This represents a* fully-adversarial and adaptive learning environment*, and unlike the models presented in the related work [31], the source of risk here is* exogenous *and* reactive *to the quotes of the MM.*

The principle of adversarial learning — as with other successful applications [68] — is that if the MM plays a strategy that is not robust, then this can (and ideally will, in training) be exploited by the adversary. If an NE strategy is played by the MM, then their strategy is robust and cannot be exploited. While there are no guarantees that an NE will be reached using ARL, we show in Section 7.5 via empirical best response computations that our approach consistently converges to reasonable approximations thereof. Moreover, we show that the derived strategies consistently outperform past approaches in terms of absolute performance *and* robustness to model ambiguity.

Robustness of this kind (through the use of ARL) was first introduced by Pinto, Davidson, Sukthankar, and Gupta [130] who demonstrated its effectiveness across a number of standard OpenAI gym domains. We adapt their RARL algorithm to support incremental actor-critic based methods and facilitate *asynchronous training*, though many of the features remain the same. The adversary is trained in parallel with the market maker, is afforded the same access to state — *including the inventory of the MM*, Ω_t — and uses the same (actor-critic) algorithm for learning. In effect, we make the assumption that the market is highly efficient and able to recover a information from its interactions with the MM. This is a more interesting setting to study since it explores the impact of inventory risk in the adversarial setting. This approach is similar to the stochastic fictitious play method of Pérolat, Piot, and Pietquin [127] for multi-stage games.

7.5 EXPERIMENTS

Both the MM and adversary agents use the NAC-S(λ) algorithm, a natural actor-critic method [173] for stochastic policies (i.e. mixed strategies) using semi-gradient SARSA(λ) [139] for policy evaluation. The value functions are represented by *compatible* [129] radial basis function networks of 100 (uniformly distributed) Gaussian prototypes with *accumulating* eligibility traces [162]. For more details on these methods we refer the reader back to Chapter 2.

The MM learns a bivariate Normal policy for ψ_t (Equation 6.15) and ξ_t (Equation 6.16) with a diagonal covariance matrix. The mean and variance vectors are modelled by linear function approximators using 3rd-order polynomial bases [96]. Both variances and the mean of the spread term, ψ_t, are kept positive via a softplus transformation. The adversary learns a Beta policy [42], shifted and scaled to cover the market parameter intervals. The two shape parameters are learnt the same as for the variance of the Normal distribution above, with a translation of $+1$ to ensure unimodality and concavity.

In each of the experiments to follow, the value function was pre-trained for 1000 episodes (with a learning rate of 10^{-3}) to reduce variance in early policy updates. Both the value function and policy were then trained for 10^6 episodes, with policy updates every 100 time steps, and a learning rate of 10^{-4} for both the critic and

Table 7.1: Performance and characteristics of market makers trained and evaluated against the FIXED adversary.

η_1	η_2	Terminal wealth	Sharpe ratio	Terminal inventory	Average spread
0.0	0.0	49.9 ± 15.3	3.26	0.53 ± 7.54	1.42 ± 0.02
1.0	0.0	53.8 ± 9.1	5.88	-0.04 ± 1.19	1.76 ± 0.02
0.5	0.0	53.6 ± 11.8	4.39	0.03 ± 1.24	1.66 ± 0.03
0.1	0.0	55.4 ± 11.0	5.01	-0.17 ± 1.87	1.42 ± 0.02
0.01	0.0	51.6 ± 13.9	3.70	0.88 ± 4.37	1.42 ± 0.02
0.0	0.01	60.6 ± 6.7	9.02	0.01 ± 1.44	1.60 ± 0.02
0.0	0.001	60.2 ± 7.6	7.94	0.14 ± 2.97	1.44 ± 0.02

policy. The value function was configured to learn $\lambda = 0.97$ returns. The starting time was chosen uniformly at random from the interval $t_0 \in [0.0, 0.95]$, with starting price $Z_0 = 100$ and inventory $\Omega_0 \in [\underline{\Omega} = -50, \overline{\Omega} = 50]$. Innovations in Z_t occurred with fixed volatility $\sigma = 2$ for the interval $[t_0, 1]$ with increment $\Delta t = 0.005$.

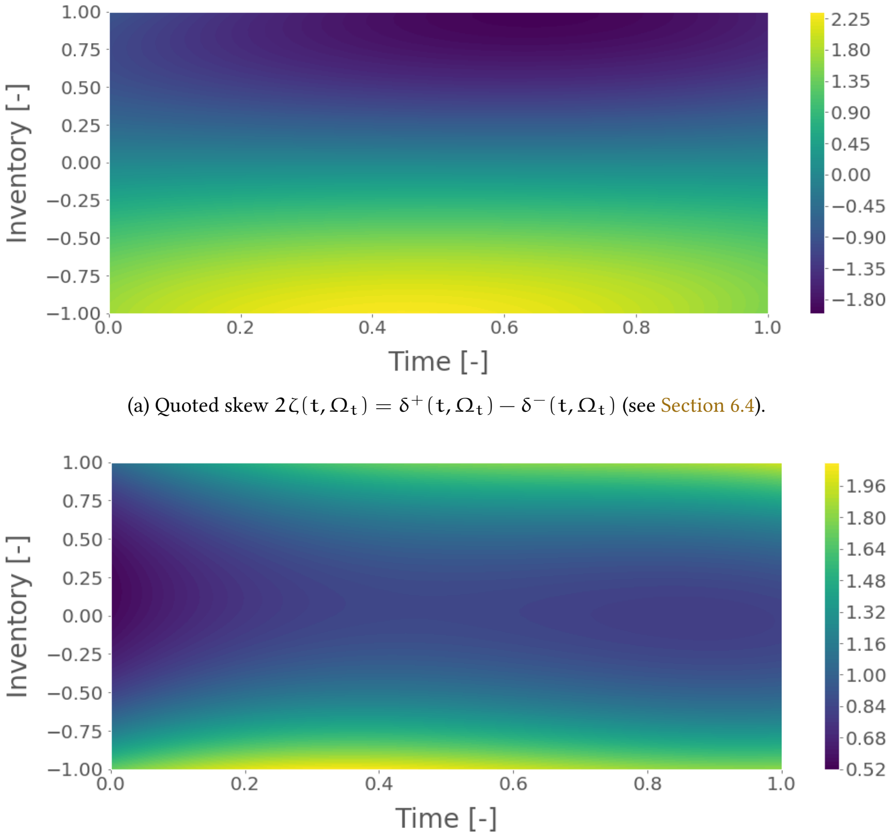
7.5.1 *Fixed Setting*

We first trained MMs against the Fixed adversary; i.e. a standard single-agent learning environment. Both the risk-neutral and risk-averse formulations of Equation 6.22 were used with risk parameters $\eta_1 \in \{1, 0.5, 0.1, 0.01\}$ and $\eta_2 \in \{0.01, 0.001\}$. Table 7.1 summarises the performance for the resulting agents. To provide some further intuition, we illustrate one of the learnt policies in Figure 7.3. In this case, the agent learnt to offset its price asymmetrically as a function of inventory, with increasing intensity as we approach the terminal time.

7.5.2 *Randomised Setting*

Next, MMs were trained in an environment with a Randomised adversary; a simple extension to the training procedure that aims to develop robustness to epistemic risk. To compare with earlier results, the strategies were also tested against the Fixed adversary — a summary of which, for the same set of risk parameters, is given in Table 7.2.

The impact on test performance in the face of model ambiguity was then evaluated by comparing market makers trained on the Fixed adversary with those trained against the Randomised adversary. Specifically, out-of-sample tests were carried out in an environment with a Randomised adversary. This means that the model dynamics at *test-time* were different from those at training time. While not explicitly adversarial, this misspecification of the model represents a non-trivial challenge for robustness. Overall, we found that market makers trained against the Fixed adversary exhibited no change in average wealth creation, but an increase of 98.1% in the variance across all risk parametrisations (see Table 7.1). On the other hand, market makers originally trained against the Randomised adversary yielded a lower average increase in the variance of 86.0%. The Randomised adversary clearly helps, but the sensitivity to changes in market dynamics in both cases are significant and would lead to strategies with a Sharpe ratio that is half its originally quoted value. As we will see next, this is not the case for strategies trained using a Strategic adversary.



(a) Quoted skew $2\zeta(t, \Omega_t) = \delta^+(t, \Omega_t) - \delta^-(t, \Omega_t)$ (see Section 6.4).

(b) Quoted spread $\psi(t, \Omega_t) = \delta^+(t, \Omega_t) + \delta^-(t, \Omega_t)$ (see Section 6.4).

Figure 7.3: Most probable (modal) action for the risk-averse Gaussian policy learnt using NAC-S(λ) with $\eta_1 = 0$ and $\eta_2 = 0.01$. Time is measured as a proportion of the time limit (i.e. 0.8 corresponds to 80% of the episode), and inventory is measured as a signed fraction of imposed upper/lower bounds.

Table 7.2: Performance and characteristics of market makers trained against the RANDOM adversary and evaluated in the FIXED environment.

η_1	η_2	Terminal wealth	Sharpe ratio	Terminal inventory	Average spread
0.0	0.0	49.6 ± 14.9	3.33	0.36 ± 7.14	1.36 ± 0.05
1.0	0.0	53.6 ± 8.0	6.72	0.02 ± 1.09	1.87 ± 0.02
0.5	0.0	55.5 ± 8.9	6.21	-0.03 ± 1.21	1.68 ± 0.02
0.1	0.0	55.1 ± 10.7	5.16	-0.18 ± 1.56	1.46 ± 0.03
0.01	0.0	54.2 ± 11.8	4.60	-0.63 ± 3.93	1.47 ± 0.02
0.0	0.01	61.3 ± 6.7	9.15	-0.07 ± 1.32	1.60 ± 0.02
0.0	0.001	607 ± 7.3	8.30	-0.07 ± 2.62	1.44 ± 0.02

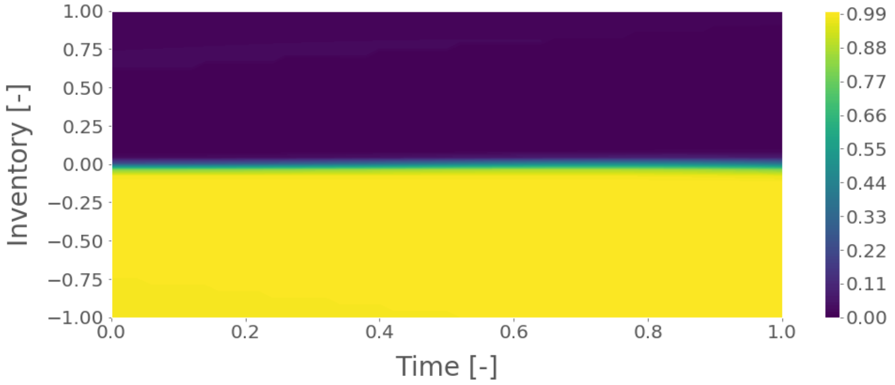


Figure 7.4: Policy learnt by the adversary for manipulating price against the market maker with $\eta_1 = \eta_2 = 0$. The solution takes the form of continuous approximation of the binary solution derived in Theorem 3.

Table 7.3: Performance and characteristics of market makers trained against the STRATEGIC adversary (with varying degrees of control) and evaluated in the FIXED environment.

Adversary	η_1	η_2	Terminal wealth	Sharpe ratio	Terminal inventory	Average spread
$\{b\}$	0	0	61.2 ± 6.9	8.87	0.05 ± 2.14	1.43 ± 0.01
$\{A^\pm\}$	0	0	47.1 ± 16.8	2.80	2.51 ± 7.72	1.46 ± 0.02
$\{k^\pm\}$	0	0	48.5 ± 16.1	3.02	0.60 ± 7.91	1.45 ± 0.02
$\{b, A^\pm, k^\pm\}$	0	0	61.6 ± 6.6	9.30	-0.05 ± 1.93	1.44 ± 0.02
$\{b, A^\pm, k^\pm\}$	1.0	0	57.4 ± 6.7	8.51	-0.02 ± 0.97	1.75 ± 0.02
	0.5	0	60.2 ± 6.8	8.84	-0.07 ± 1.04	1.60 ± 0.01
	0.1	0	61.5 ± 6.6	9.25	-0.05 ± 1.37	1.49 ± 0.02
	0.01	0	61.7 ± 6.6	9.28	-0.09 ± 1.89	1.49 ± 0.01
	0	0.01	60.6 ± 6.6	9.11	-0.03 ± 1.19	1.65 ± 0.02
	0	0.001	61.22 ± 6.5	9.45	0.0 ± 1.71	1.44 ± 0.01

7.5.3 *STRATEGIC Setting*

First consider an adversary that controls the drift b_t only — a direct multi-stage extension of the game instance analysed in Theorem 3. With risk-neutral rewards, we found that the adversary learns a time-independent binary policy (Figure 7.4) that is identical to the strategy in the corresponding single-stage game; see Section 7.3. We also found that the strategy learnt by the MM in this setting generates profits and associated Sharpe ratios in excess of all other strategies seen thus far when tested against the FIXED adversary (see Table 7.3). This is true also when comparing with tests run against the RANDOMISED or STRATEGIC adversaries, suggesting that the adversarially trained MM is indeed more robust to test-time model discrepancies.

This, however, does not extend to STRATEGIC adversaries with control over either *only* $A_t^\pm$ or *only* $k_t^\pm$. In these cases, performance was found to be *no better* than the corresponding MMs trained in the FIXED setting with a conventional learning setup. The intuition for this again derives from the single-stage analysis. That is, the adversary almost surely chooses a strategy that minimises $A_t^\pm$ (equivalently maximises $k_t^\pm$) in order to decrease the probability of execution, thus decreasing the profits of the MM derived from execution and its ability to manage inventory effectively. The control afforded to the adversary must be coupled in some way with

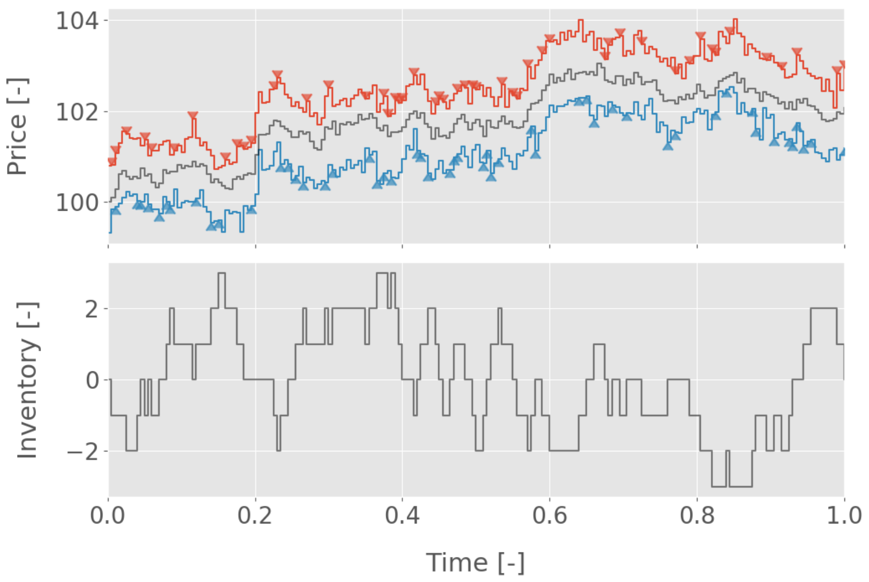


Figure 7.5: Sample rollout of an adversarially trained market making strategy. Quoted ask (red) and bid (blue) prices are shown around the mid-price. Executions are illustrated using arrows and the resulting inventory process is illustrated in the lower plot.

sources of variance — such as inventory speculation — in order for robustness to correspond to practicable forms of risk-aversion.

The natural subsequent question to pose is whether an adversary with simultaneous control over $\{b_t, A_t^\pm, k_t^\pm\}$ produces strategies outperforming those where the adversary controls b_t alone. This is certainly plausible since combining all five model parameters could lead to more interesting strategies, such as driving inventory up/down only to switch the drift at the peak (i.e. pump and dump). We investigated this by training an adversary with control over all five parameters and the resulting performance can be found in Table 7.3. This shows an improvement in the Sharpe ratio of 0.27 and lower variance on terminal wealth. Interestingly, these MMs also quote tighter spreads on average — the values even approaching that of the risk-neutral MM trained against a FIXED adversary. This indicates that the strategies are able to achieve epistemic risk aversion *without* charging more to counterparties. An example rollout of the strategy is given in Figure 7.5.

Exploring the impact of varying risk parameters of the reward function, η_1 and η_2 (Equation 6.22), we found that in all cases MM strategies trained against a STRATEGIC adversary with a risk-averse reward outperformed their counterparts in Table 7.1 and Table 7.2. It is unclear, however, if changes to the reward function away from the risk-neutral variant actually improved the strategy in general. Excluding when $\eta_2 = 0.001$, all values appear to do worse than for an adversarially trained MM with risk-neutral reward. It may well be that the addition of inventory penalty terms actually boosts the power of the adversary and results in strategies that try to avoid trading at all, a frequent problem in this domain.

TRAINING DYNAMICS Closer examination of the evolution of the market maker's policy during training also reveals some intriguing properties about how the two agents drive one another's behaviour. Figure 7.6 illustrates an example in which the price skewing of the MM oscillates in the early stages of learning. To begin, the agent quotes prices that are independent of it's inventory, and simply increases the probability of executing on one side of the market. After a few timesteps this yields

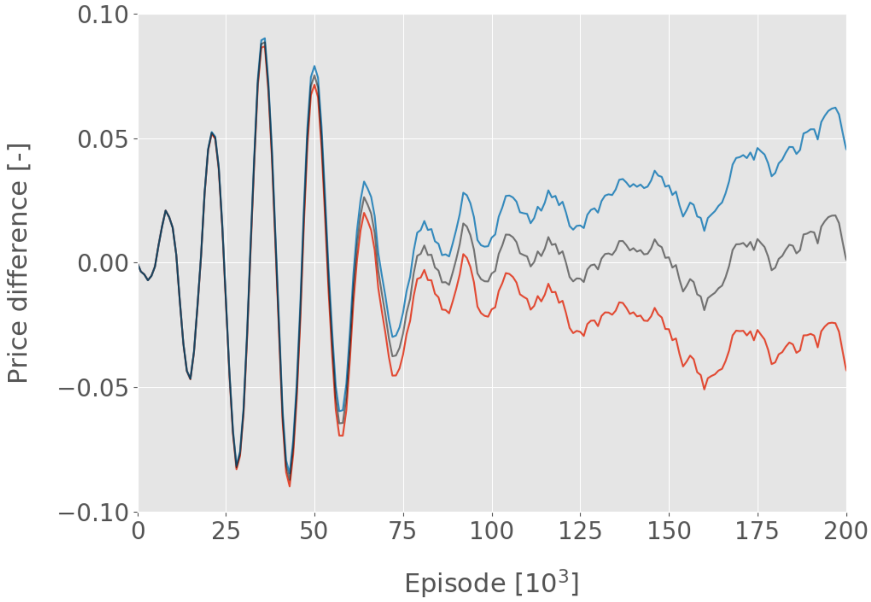


Figure 7.6: Oscillatory behaviour in the best response dynamics between directionally biased market making strategies in the early stages of training in the Strategic setting. Each curve corresponds to the evolution of the modal value of the policy's skew factor, η_2, during training for the state $s = (0, \Omega)$. Three cases are considered: when the agent's inventory is neutral $\Omega = 0$ (grey), bullish $\Omega = 5$ (red) or bearish $\Omega = -5$ (blue).

an inventory at the upper/lower limit depending on the sign of the adversary's chosen drift, b. If the adversary chooses a positive drift, so then the MM exploits this by accumulating a positive inventory and speculating on future price changes. Around episode 75000, the policy enters a new regime in which the strategy is robust to *any* drift the adversary can throw at the MM. This takes the form of a skewing strategy that protects against adverse price movements and induces mean reversion in the inventory process. It is in this regime that the MM begins converging directly towards to subgame perfect NE derived in Section 7.3.

VERIFICATION OF APPROXIMATE EQUILIBRIA Holding the strategy of one player fixed, we empirically computed the best response against it by training. We found consistently that neither the trader nor adversary deviated from their policy. This suggests that our ARL method were finding reasonable approximate Nash equilibria. While we do not provide a full theoretical analysis of the stochastic game, these findings are corroborated by those in Section 7.3, since, as seen in Figure 7.4, the learned policy in the multi-stage setting corresponds closely to the equilibrium strategy from the single-stage case that was presented in Section 7.3.

7.6 CONCLUSIONS

In this chapter we have introduced a new approach for learning trading strategies with ARL. The learned strategies are robust to the discrepancies between the market model in training and testing. We show that our approach leads to strategies that outperform non-adversarially trained strategies in terms of PnL and Sharpe ratio, and have comparable spread efficiency. This is shown to be the case for out-of-sample tests in all three of the considered settings: Fixed, Randomised, and Strategic. In other words, our learned strategies are not only more robust to misspecification, but also dominate in overall performance.

In some special cases we show that the learned strategies correspond to Nash equilibria of the corresponding single-stage game. More widely, we empirically show that the learned strategies correspond to approximate equilibria in the multi-stage Markov game.

Finally, we remark that, while our paper focuses on market making, the approach can be applied to other trading scenarios such as optimal execution and statistical arbitrage, where we believe it is likely to offer similar benefits. Further, it is important to acknowledge that this methodology has significant implications for safety of RL in finance. Training strategies that are explicitly robust to model misspecification makes deployment in the real-world considerably more practicable.

8 ROBUSTNESS TO ALEATORIC RISK

8.1 OUTLINE

rl! (**rl!**) solves the problem of how to act optimally in a potentially unknown environment. While it does this very well in many cases, it has become increasingly clear that uncertainty about the environment — both epistemic and aleatoric in nature — can have severe consequences on the performance of our algorithms. While many problems can be solved by maximising the expected returns alone, it is rarely sufficient, and shies away from many of the subtleties of the real-world. In fields such as finance and health, the mitigation of risk is absolutely foundational, and the lack of practical methods is one of the biggest roadblocks in wider adoption of RL. Now, recent developments in risk-sensitive RL have started to enable practitioners to design algorithms to tackle their problems. However, many of these approaches rely on full trajectory rollouts, and most only consider variance-related risk criteria which:

(i) are not suited to all domains; and

(ii) are often non-trivial to estimate in an online setting.

One rarely has the luxury of ready access to high-quality data, and humans' definition of risk is highly nuanced [146, 176].

This observation is not unique and indeed many fields have questioned the use of symmetric risk measures to correctly capture human preferences. Markowitz himself noted, for example, that "semi-variance seems a more plausible measure of risk than variance, since it is only concerned with adverse deviations" [109]. Yet, apart from Tamar, Chow, Ghavamzadeh, and Mannor [169], who introduced semi-deviation as a possible measure of risk, very little work has been done to address this gap in RL research. Furthermore, of those that do, even fewer still consider the question of how to learn an incremental approximation, instead opting to directly estimate policy gradients with Monte-Carlo sampling.

Contributions

The **first contribution** of this chapter lies in the development of the lower partial moment (LPM) — i.e. the expected value of observations falling below some threshold — as an effective downside risk measure *that can be approximated efficiently through temporal-difference (TD) learning*. This insight derives from the sub-additivity of the $\max\{\cdot, \cdot\}$ function and enables us to define a recursive bound on the LPM that serves as a proxy in constrained policy optimisation. We are able to prove that the associated Bellman operator is a contraction, and analyse the variance on the transformed reward that emerges from the approximation to gain insight into the stability of the proposed algorithm.

The **second key contribution** is to show that the reward-constrained policy optimisation framework of Tessler, Mankowitz, and Mannor [172] can be extended to use natural policy gradients. In so doing, we show that the classical policy gradient of Sutton, McAllester, Singh, and Mansour [164] can be generalised to an arbitrary linear combination of estimators, each individually satisfying the compatible function approximation requirements. While multi-objective problems in RL are notoriously hard to solve [108], natural gradients are known to address some of the issues associated with convergence to local minima; we posit that this is particularly effective in the multi-objective setting. The resulting algorithm, used alongside our LPM estimation procedure and generalised compatibility result is easy to implement and is shown to be highly effective in a number of empirical problem settings.

8.2 RELATED WORK

Past work on risk-sensitivity and robustness in RL can be split into those that tackle epistemic uncertainty, and those that tackle aleatoric uncertainty – which is the focus of this chapter. Aleatoric risk (the risk inherent to a problem) has received much attention in the literature. For example, in 2001, Moody and Saffell devised an incremental formulation of the Sharpe ratio for on-line learning. Shen, Tobia, Sommer, and Obermayer [149] later designed a host of value-based methods using utility functions (see also [98]), and work by Tamar, Di Castro, and Mannor [171] and Sherstan, Bennett, Young, Ashley, White, White, and Sutton [150] even tackle the estimation of the variance on returns; a contribution closely related to those in this chapter. More recently, a large body of work that uses policy gradient methods for risk-sensitive RL has emerged [22, 169, 170, 172]. Epistemic risk (the risk associated with, e.g. known model inconsistencies) has also been addressed, though to a lesser extent [93, 130] (see also Chapter 7). There also exists a distinct but closely related field called "safe RL" which includes approaches for safe exploration; see the excellent survey by Garcıa and Fernández [62].

8.3 CONSTRAINED MDPS

Constrained MDPs are a generalisation of MDPs to problems in which the policy is subject to a secondary set of optimality criteria which encode behavioural requirements beyond the expected return [6]. These constraints are represented by a penalty function $c(s, a)$ (akin to the reward function), and constraint functions

$$C^\gamma_\pi(s) = \lim_{n\to\infty} \mathbb{E}_\pi\left[\sum_{k=0}^{n} \gamma^k c(s_{t+k}, a_{t+k}) \,\middle|\, s_t = s\right], \tag{8.1}$$

and

$$C^\gamma_\pi(s, a) = \lim_{n\to\infty} \mathbb{E}_\pi\left[\sum_{k=0}^{n} \gamma^k c(s_{t+k}, a_{t+k}) \,\middle|\, s_t = s, a_t = a\right], \tag{8.2}$$

over the realised penalties (with some abuse of notation). These two functions are equivalent to the discounted value functions defined in Equation 2.19 and Equation 2.20 in which the discounted return has simply been replaced with the discounted sum of the penalty function realisations. As in Tessler, Mankowitz, and Mannor [172], we denote the objective associated with the constraint function by the "penalty-to-go": $J_c(\pi) = \mathbb{E}_{d^\gamma_\pi,\pi}[c(s, a)]$. The problem of optimising a policy $\pi \in \Pi_s$ thus becomes constrained optimisation problem with mean-risk flavour,

$$\begin{aligned} \max_{\pi\in\Pi_s} \quad & J_r(\pi), \\ \text{subject to} \quad & J_c(\pi) \leqslant \nu, \end{aligned} \tag{8.3}$$

where $\nu \in \mathbb{R}$ is a user-specified threshold parameter.

Constrained optimisation problems of this type are typically recast as saddle-point problems by Lagrange relaxation [18] to give

$$\min_{\lambda\geqslant 0} \max_{\pi} \mathcal{L}(\lambda, \pi) = \min_{\lambda\geqslant 0} \max_{\pi} \left[J_r(\pi) - \lambda \left(J_c(\pi) - \nu\right)\right], \tag{8.4}$$

where $\mathcal{L}(\lambda, \pi)$ denotes the Lagrangian, and $\lambda \geqslant 0$ the Lagrange multiplier. *Feasible solutions* are those that satisfy the constraint — e.g. those for which $J_c(\pi) \leqslant \nu$ — the existence of which depends on the problem domain and choice of the penalty $c(s, a)$ and threshold ν. Any policy that is not a feasible solution is considered sub-optimal.

Approaches to solving problems of this kind revolve around the derivation and estimation of the gradients of $\mathcal{L}(\lambda, \pi_\theta)$ with respect to the parameters of a continuously differentiable policy $\pi_\theta \in \Pi_{s,\theta}$ and the multiplier λ [20, 24]. That is, we restrict ourselves to a class of policies that can be updated through stochastic gradient descent and derive the corresponding policy gradient update. For the model specified above, the Lagrangian $\mathcal{L}(\lambda, \theta)$ (note we have replaced π with θ) has derivatives

$$\nabla_\lambda \mathcal{L}(\lambda, \theta) = \nu - C_\pi^\gamma(s), \tag{8.5}$$

and

$$\nabla_\theta \mathcal{L}(\lambda, \theta) = \mathbb{E}_\pi[[V_\pi^\gamma(s) - \lambda C_\pi^\gamma(s)] \, \nabla_\theta \ln \pi_\theta(a \mid s)] \, , \tag{8.6}$$

which are obtained using the log-likelihood trick [172, 183].

8.3.1 *Reward Constrained Policy Optimisation*

Recent work by Tessler, Mankowitz, and Mannor [172] extended previous methods for constrained problems like (8.4) to handle *general constraints* without prior knowledge while remaining invariant to reward scaling. Their algorithm, named reward constrained policy optimisation (RCPO), uses samples obtained through simulations of the MDP to estimate (8.5) and (8.6), and update λ and θ using multi-timescale stochastic approximation:

$$\lambda_{k+1} \leftarrow \Gamma_\lambda \left[\lambda_k - \alpha_\lambda(k) \nabla_\lambda \mathcal{L}(\lambda, \theta)\right] , \tag{8.7}$$

and

$$\theta_{k+1} \leftarrow \Gamma_\theta \left[\theta_k - \alpha_\theta(k) \nabla_\theta \mathcal{L}(\lambda, \theta)\right] , \tag{8.8}$$

where Γ_θ projects the weights onto a compact, convex set, and Γ_λ projects λ into $[0, \bar{\lambda}]$. When the policy update is performed at a faster rate than for λ (as well as other standard Robbins-Monro assumptions [138] — see Assumptions 1 through 3 in [172]), then by Theorem 1 and Lemma 1 of Tessler, Mankowitz, and Mannor [172], the iterates converge to a local optimum almost surely and this fixed point is a feasible solution.

The reward constrained policy optimisation framework offers a powerful methodology for optimising behaviours in a contrained MDP. Most importantly, it allows us to express the learning objective naturally in terms of the constraint functions, and it removes the need to perform laborious scaling and tuning of the rewards and penalties; this is a key advantage of dual approaches. As we will show in Section 8.6, this algorithm may also be extended to leverage natural gradients (see Section 2.4.4) which further improves the stability and performance of the approach.

8.4 DOWNSIDE RISK MEASURES

In general, the definition of the penalty function, $c(s, a)$, depends on the problem setting and desired behaviour. This need not always be motivated by risk. For example, in robotics problems, it may take the form of a cost applied to policies with a large jerk or snap in order to encourage smooth motion. In economics and health problems, the constraint is typically based on some measure of risk/dispersion associated with the uncertainty in the outcome, such as the variance. In many real world applications, it is particularly appropriate (and natural) to consider *downside* *risk*, such as the dispersion of returns below a target threshold, or the likelihood of Black Swan events. Intuitively, we may think of a *general risk measure* as a measure of "distance" between risky situations and those that are risk-free, when both favourable and unfavourable discrepancies are accounted for equally. A *downside risk measure*, on the other hand, only accounts for deviations that contribute unfavourably to risk [47, 53].

8.4.1 *Partial Moments*

Partial moments were first introduced as a means of measuring the probability-weighted deviations below (or above) a target threshold τ. These feature prominently in finance and statistical modelling as a means of defining (asymmetrically) risk-adjusted metrics of performance [57, 147, 155, 161]. Our definition of partial moments, stated below, follows the original formulation of Fishburn [58].

Definition 21. *Let* $\tau \in \mathbb{R}$ *denote a target value, then the* m^{th}*-order partial moments of the random variable* X *about* τ *are given by*

$$\mathbb{M}_{+}^{m}\left[X|\tau\right] \doteq \mathbb{E}\left[(\tau - X)_{+}^{m}\right] \quad \textit{and} \quad \mathbb{M}_{-}^{m}\left[X|\tau\right] \doteq \mathbb{E}\left[(X - \tau)_{+}^{m}\right], \tag{8.9}$$

where $(x)_{+} \doteq \max\{x, 0\}$ *and* $m \in [1, \infty)$.

The two quantities $\mathbb{M}_{-}^{m}\left[X|\tau\right]$ and $\mathbb{M}_{+}^{m}\left[X|\tau\right]$ are known as the *lower* and *upper* partial moments (LPM/UPM), respectively. When the target is chosen to be the mean — i.e. for $\tau = \mathbb{E}\left[X\right]$ — we refer to them as the *centralised partial moments*, and typically drop τ from the notation for brevity. For example, the semi-variance is given by the centralised, second-order LPM, $\mathbb{M}_{-}^{2}\left[X\right]$.

Unlike the expectation operator, the partial moment operators in (8.9) are non-linear functions of the input and satisfy very few of the properties that make expected values well behaved. Of particular relevance to this work is the fact that they are non-additive. This presents a challenge in the context of approximation since we cannot directly apply the Robbins-Monro algorithm [138]. As we will show in Section 8.5, however, we can estimate an upper bound for the first partial moment, for which we introduce the following key properties in Lemma 2 and Lemma 3 below. These are well known mathematical facts and we repeat the proofs here merely for completeness.

Lemma 2 (Max/min decomposition). *The maximum of two terms,* $\max\{x, y\}$*, can be expressed as* $f(x, y) \doteq (|x - y| + x + y)/2$*. Similarly,* $\min\{x, y\} = g(x, y) \doteq (x + y - |x - y|)/2$*.*

Proof. Recall that $|z| = z$ if $z \geqslant 0$, and $|z| = -z$ if $z < 0$. Thus, if $x \geqslant y$ then $|x - y| = x - y$, $f(x, y)$ reduces to x and $g(x, y)$ reduces to y; similarly, if $x < y$, then $|x - y| = y - x$, then $f(x, y)$ takes the value y and $g(x, y)$ becomes x. This means that both $f(x, y)$ and $g(x, y)$ satisfy the properties of the max and min and thus the proof is complete. ■

Lemma 3 (Subadditivity of partial moments). *Consider a pair of real-valued random variables* X *and* Y*, and a fixed, additive target* $\tau \doteq \tau_X + \tau_Y$*. Then for* $m = 1$*, the partial moment is* subadditive *in* X *and* Y*, with*

$$\mathbb{M}_{\pm}[X + Y \mid \tau] \leqslant \mathbb{M}_{\pm}[X \mid \tau_X] + \mathbb{M}_{\pm}[Y \mid \tau_Y]\,. \tag{8.10}$$

Proof. Consider the lower partial moment, expressing the inner term as a function of real and absolute values $\left[|\tau - X - Y| + \tau - X - Y\right]/2$ (follows from Lemma 2). By the subadditivity of the absolute function (triangle inequality), it follows that:

$$(\tau - X - Y)_{+} = (\tau_X - X + \tau_Y - Y)_{+} \leqslant (\tau_X - X)_{+} + (\tau_Y - Y)_{+}\,. \tag{8.11}$$

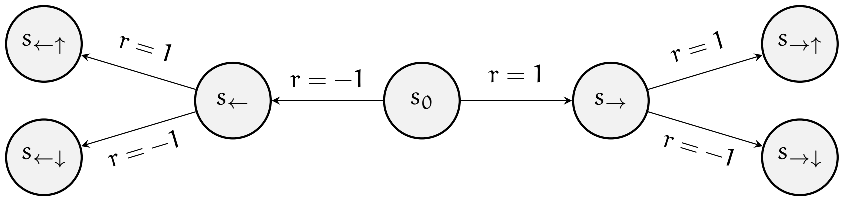
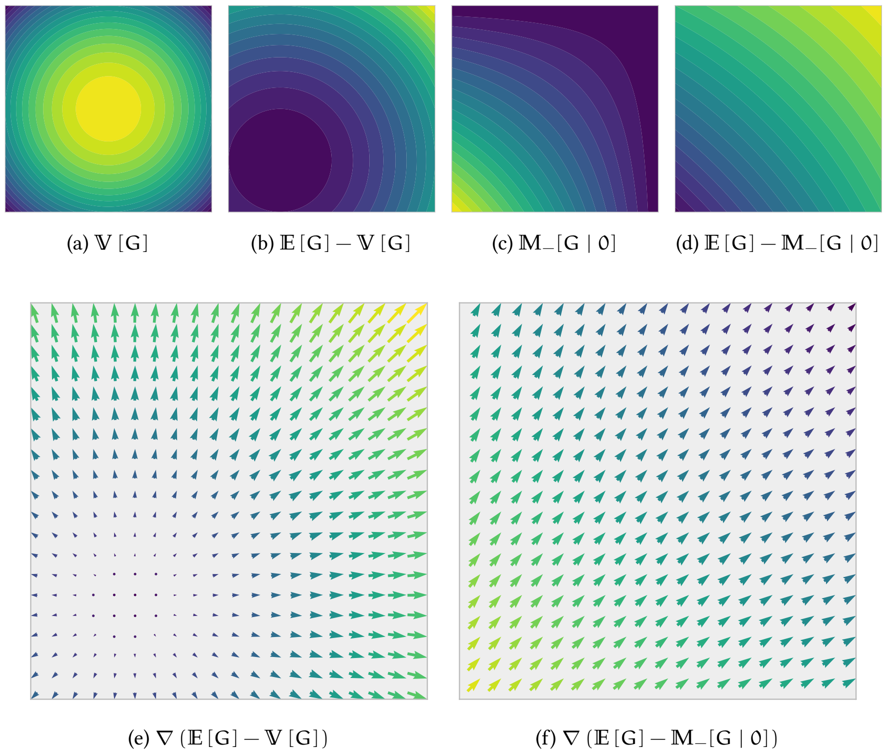


Figure 8.1: Simple MDP with two actions and 7 states; the terminal state is omitted.



(a) $\mathbb{V}[G]$ (b) $\mathbb{E}[G] - \mathbb{V}[G]$ (c) $\mathbb{M}_{-}[G \mid 0]$ (d) $\mathbb{E}[G] - \mathbb{M}_{-}[G \mid 0]$

(e) $\nabla(\mathbb{E}[G] - \mathbb{V}[G])$ (f) $\nabla(\mathbb{E}[G] - \mathbb{M}_{-}[G \mid 0])$

Figure 8.2: Moments of the return G generated by the MDP in Figure 8.1. The x-axis corresponds to $\theta_1 \in [0, 1]$, and the y-axis to $\theta_2 \in [0, 1]$. Higher values are in yellow, and lower values in dark blue.

By the linearity of the expectation operator, we arrive at (8.10). This result may also be derived for the upper partial moment by the same logic. ■

MOTIVATING EXAMPLE. Why is this so important? Consider the MDP in Figure 8.2, with stochastic policy parameterised by $\theta \in [0, 1]^2$ such that $\pi_\theta(\rightarrow \mid s_0) \doteq \theta_1$ and $\pi_\theta(\uparrow \mid s_{\leftarrow}) = \pi_\theta(\uparrow \mid s_{\rightarrow}) \doteq \theta_2$. As shown by Tamar, Di Castro, and Mannor [170], even in a simple problem such as this, the space of solutions for a mean-variance criterion is non-convex. Indeed, Figure 8.2 shows that the solution space exhibits local-optima for the deterministic policies $\theta_{1,2} \in \{(0, 0), (1, 0), (0, 1)\}$. On the other hand, the lower partial moment only exhibits a single optimum at the correct solution of $\theta_1 = \theta_2 = 1$. While this of course says nothing of the general case, it does suggest that partial moments have a valid place in risk-averse RL, and may in some instances lead to more intuitive results; especially in terms of human motivations.

The extremal values correspond to the three minima in variance seen in Figure 1 of Tamar, Di Castro, and Mannor [170].

8.5 POLICY EVALUATION

The same follows for the UPM, though it's validity in promoting risk-sensitivity is unclear.

Our objective in this section is now to derive an incremental, temporal-difference (TD) prediction algorithm for the first LPM of the return distribution, G_t. To begin, let $\rho[\tau](s,a)$ denote the first LPM of G_t with respect to a target function $\tau(s,a)$, starting from state-action pair (s,a):

$$\rho[\tau](s,a) \doteq \mathbb{M}_-[G_t \mid S_t = s, A_t = a, \tau]\,, \tag{8.12}$$

where the *centralised moments* are shortened to $\rho(s,a)$. For a given target, this function can be learnt trivially through Monte-Carlo (MC) estimation using batches of sample trajectories. Indeed, we can even learn the higher-order moments using such an approach. However, while this yields an unbiased estimate of the LPM, it comes at the cost of increased variance and decreased sample efficiency [162] — something we want to avoid. This is especially pertinent in risk-sensitive RL which is often concerned with (already) highly stochastic domains. The challenge is that (8.12) is a non-linear function of G_t which does not have a direct recursive form amenable to TD-learning.

Rather than learn the LPM directly, we propose to learn a proxy in the form of an upper bound. To begin, we note that by Lemma 3, the LPM of the return distribution satisfies

$$\rho[\tau](s,a) \leqslant \mathbb{M}_-[r_{t+1} \mid \tau_r(s,a)] + \gamma\mathbb{M}_-\left[G_{t+1} \,\middle|\, \mathbb{E}_\pi[\tau(s',a')]\right], \tag{8.13}$$

for $\tau(s,a) = \tau_r(s,a) + \gamma\mathbb{E}_\pi[\tau(s',a')]$. Unravelling the final term ad infinitum yields a geometric series of bounds on the reward moments. This sum admits a recursive form which we define as

$$\bar{\rho}[\tau](s,a) \doteq \mathbb{M}_-[r_{t+1} \mid \tau_r] + \gamma\bar{\rho}[\tau](s',a'), \tag{8.14}$$

This bears a resemblance to the reward-volatility objective of Bisi, Sabbioni, Vittori, Papini, and Restelli [22].

which is, precisely, an action-value function with non-linear reward transformation: $g(r) = (\tau_r - r)_+$. This means *we are now free to use any prediction algorithm to perform the actual TD updates*, such as SARSA(λ) or GQ(λ) [105]. We need only choose τ_r to satisfy the requirements of the problem; perhaps minimising the error between (8.12) and (8.14). For example, a fixed target yields the expression $\tau_r(s,a) = (1-\gamma)\tau$. Alternatively, a centralised variant would be given by $\tau_r(s,a) = r(s,a)$. This freedom to choose a target function affords us a great deal of flexibility in designing downside risk metrics.

8.5.1 *Convergence*

As observed by Hasselt, Quan, Hessel, Xu, Borsa, and Barreto [83], Bellman equations with non-linear reward transformations (as in Equation 8.14) carry over all standard convergence results under the assumption that the transformation is bounded. This is trivially satisfied when the rewards themselves are bounded [19]. This means that the associated Bellman operator is a contraction, and that the proxy (8.14) converges with stochastic approximation under the standard Robbins-Monro conditions [138].

8.5.2 *Variance Analysis*

To study the variance of the proposed proxy action-value function we first analyse the variance on functions of random variables. Specifically, one can show that the variance on the absolute value of a random variable is at most that of the original, untransformed quantity.

Lemma 4 (Variance bound for absolute values)**.** *For any random variable* X *we have that* $\mathbb{V}[|X|] \leqslant \mathbb{V}[X]$.

Proof.

$$\begin{aligned} \mathbb{V}[X] - \mathbb{V}[|X|] &= \mathbb{E}\left[X^2\right] - \mathbb{E}[X]^2 - \mathbb{E}\left[|X|^2\right] + \mathbb{E}[|X|]^2 \\ &= \mathbb{E}[|X|]^2 - \mathbb{E}[X]^2 && |X|^2 = X^2 \\ &= (\mathbb{E}[|X|] - \mathbb{E}[X])(\mathbb{E}[|X|] + \mathbb{E}[X]) \\ &\geqslant 0 && \mathbb{E}[|X|] \geqslant \mathbb{E}[X] \end{aligned}$$

■

This result also means that the covariance between the transformed and original random variable is bounded to the interval $[-\mathbb{V}[X], \mathbb{V}[X]]$.

Corollary 4.2 (Covariance bound for absolute values)**.** *The covariance between any random variable* X *and it's absolute value* $|X|$ *is bounded:* $-\mathbb{V}[X] \leqslant \mathrm{Cov}[|X|, X] \leqslant \mathbb{V}[X]$.

Proof. By the Cauchy-Schwarz inequality we have that $\mathrm{Cov}[|X|, X]^2 \leqslant \mathbb{V}[|X|]\,\mathbb{V}[X]$, and from Lemma 4 we know that $\mathbb{V}[|X|] \leqslant \mathbb{V}[X]$. The claim follows. ■

Lemma 4 can now be used to show that the variance on the positive part of a random variable — i.e. $X_+ \doteq (c - X)_+$ — is also bounded.

Lemma 5. *For any random variable,* X*, with support on a subset of the real line,* $\operatorname{supp} X \subseteq \mathbb{R}$*, we have that* $\mathbb{V}\left[(c - X)_+\right] \leqslant \mathbb{V}[X]$ *for arbitrary constant* $c \in \operatorname{supp} X$.

Proof. Recall that, by Lemma 2, the term $(c - X)_+$ may be decomposed into the sum $(|c - X| + c - X)/2$. From the standard definitions of variance, we have that

$$\mathbb{V}[(c - X)_+] = \mathbb{V}[|c - X|]/4 + \mathbb{V}[c - X]/4 + \mathrm{Cov}[|c - X|, c - X]/2.$$

Using the result of Lemma 4 and Lemma 5, we can bound each individual value in this expression. This means that $\mathbb{V}[(c - X)_+] \leqslant \mathbb{V}[c - X]$, and thus that $\mathbb{V}[(c - X)_+] \leqslant \mathbb{V}[X]$ since c is a constant and $\mathbb{V}[-X] = \mathbb{V}[X]$. ■

The outcome of Lemma 5 can be used to show that the non-linear reward term in Equation 8.14 exhibits a lower variance than that of the original reward: $\mathbb{V}[(\tau_r - r)^+] \leqslant \mathbb{V}[r]$. By the same logic, we must have that the variance on the positive part of the return is also bounded, such that $\mathbb{V}\left[(c - G)_+\right] \leqslant \mathbb{V}[G]$. We posit that in most realistic cases, this implies that the variance on the learning procedure for the LPM proxy is at most that of the $\widehat{Q}(s, a)$ function. Since the Bellman equations are contractions, we know then that SARSA(λ), for example, converges asymptotically and globally to the correct values under standard conditions (greedy in the limit of infinite exploration, and the Robbins-Monro conditions). As noted by Pendrith and Ryan [125], this means that the Bellman targets for $\widehat{Q}(s, a)$ and $\widehat{\overline{\rho}}(s, a)$ will be dominated by the reward terms. This can bee seen in the expression below:

$$\mathbb{V}\left[r + \gamma \hat{f}\right] = \mathbb{V}[r] + \gamma^2 \mathbb{V}\left[\hat{f}\right] + 2\gamma\,\mathrm{Cov}\left[r, \hat{f}\right].$$

In the limit as $\hat{f}(s, a) \to \mathbb{E}[f(s, a)]$, so the variance on the estimator $\hat{f}(s, a)$ decreases to zero and the only remaining term is $\mathbb{V}[r]$. By Lemma 5, the variance on the $\widehat{Q}(s, a)$ target will therefore be greater than for $\widehat{\overline{\rho}}(s, a)$ as training proceeds and the estimates converge. This suggests that the estimate for the LPM proxy will be stable and will converge at a rate bounded by the same algorithm estimating $\widehat{Q}(s, a)$.

8.6 POLICY OPTIMISATION

In the previous section we saw how the upper bound on the LPM of the return can be learnt effectively in an incremental fashion. Putting this to use now requires that we integrate our estimator into a constrained policy optimisation framework. This is particularly simple in the case of RCPO, for which we incorporate (8.14) into the penalised reward function introduced in Definition 3 of Tessler, Mankowitz, and Mannor [172]. Following their template, we may derive a whole class of actor-critic algorithms that optimise for a downside risk-adjusted objective. Crucially, if the two value function estimators $\widehat{Q}(s,a)$ and $\widehat{\rho}[\tau](s,a)$ are *compatible* with the policy parameterisation [164], then we may extend RCPO to use natural policy gradients (see Chapter 2). We call the resulting algorithm natural reward constrained policy optimisation (NRCPO) for which the existence hinges on Theorem 5 below.

Theorem 5 (Additive policy gradient). *Consider an objective given by the weighted sum of* n *state-action functions,* $J(\theta) \doteq \sum_{i=1}^{n} c_i J_i(\theta) \doteq \mathbb{E}_{d_0,\pi_\theta}\left[\sum_{i=1}^{n} c_i f_i(s,a)\right]$, *where* c_i *are constants. Then, for a corresponding set of* approximators, $\widehat{f}_i(s,a)$, *if the following conditions hold:*

(i) that all $\widehat{f}_i(s,a)$ *are compatible with the policy, such that*

$$\frac{\partial \widehat{f}_i(s,a)}{\partial \boldsymbol{w}_i} = \frac{1}{\pi_\theta(a \mid s)} \frac{\partial \pi_\theta(a \mid s)}{\partial \theta}, \tag{8.15}$$

(ii) and that each $\widehat{f}_i(s,a)$ *minimises the mean-squared error*

$$\mathcal{E}_i \doteq \mathbb{E}_\pi\left[\left(\widehat{f}_i(s,a) - f_i(s,a)\right)^2\right], \tag{8.16}$$

then

$$\int_{\mathcal{S}} d_{\pi_\theta}(s) \int_{\mathcal{A}(s)} \frac{\partial \pi_\theta(a \mid s)}{\partial \theta} \sum_{i=1}^{n} c_i \widehat{f}_i(s,a) \; da \; ds \tag{8.17}$$

is an unbiased estimate of the policy gradient $\nabla_\theta J(\theta)$.

Proof. Let $f(s,a)$ be an arbitrary function of state and action and let there exist a corresponding approximator $\widehat{f}(s,a)$ with weights $\boldsymbol{w}_f$. The MSE between the true function and the approximation is given by

$$\mathcal{E}_f = \int_{\mathcal{S}} d_{\pi_\theta}(s) \int_{\mathcal{A}(s)} \pi_\theta(a \mid s) \left[\widehat{f}(s,a) - f(s,a)\right]^2 \; da \; ds. \tag{8.18}$$

If $\widehat{f}(s,a)$ fulfils requirement (8.15), then the derivative of the MSE is given by

$$\begin{aligned} \frac{\partial \mathcal{E}_f}{\partial \boldsymbol{w}_f} &= 2 \int_{\mathcal{S}} d_{\pi_\theta}(s) \int_{\mathcal{A}(s)} \pi_\theta(a \mid s) \frac{\partial \widehat{f}(s,a)}{\partial \boldsymbol{w}_f} \left[\widehat{f}(s,a) - f(s,a)\right] \; da \; ds, \\ &= 2 \int_{\mathcal{S}} d_{\pi_\theta}(s) \int_{\mathcal{A}(s)} \frac{\partial \pi_\theta(a \mid s)}{\partial \theta} \left[\widehat{f}(s,a) - f(s,a)\right] \; da \; ds. \end{aligned}$$

If we then assume that the learning method minimises the MSE defined above (i.e. requirement (8.16)), then the weights $\boldsymbol{w}_f$ yield the unique stationary point of the MSE. Equating the derivative above to zero thus results in the equality

$$\int_{\mathcal{S}} d_{\pi_\theta}(s) \int_{\mathcal{A}(s)} \frac{\partial \pi_\theta(a \mid s)}{\partial \theta} f(s,a) \; da \; ds = \int_{\mathcal{S}} d_{\pi_\theta}(s) \int_{\mathcal{A}(s)} \frac{\partial \pi_\theta(a \mid s)}{\partial \theta} \widehat{f}(s,a) \; da \; ds,$$

which implies that the true value functions in the policy gradient can be replaced with the MSE-minimising approximations without introducing bias. This is simply the classic result of Sutton, McAllester, Singh, and Mansour [164].

Now, for an additive objective function of the form $J(\theta) = \sum_{i=1}^{n} c_i J_i(\theta)$, we have that

$$\frac{\partial J(\theta)}{\partial \theta} = \sum_{i=1}^{n} c_i \frac{\partial J_i(\theta)}{\partial \theta} \tag{8.19}$$

which follows from the linearity of differentiation. From the policy gradient theorem [164], and the compatible function approximation result above, we know that each of the differential terms may be expressed by an integral of the form

$$\frac{\partial J_i(\theta)}{\partial \theta} = \int_{\mathcal{S}} d_{\pi_\theta}(s) \int_{\mathcal{A}(s)} \frac{\partial \pi_\theta(a \mid s)}{\partial \theta} \widehat{f}_i(s, a) \, da \, ds. \tag{8.20}$$

Combining (8.19) and (8.20), it follows from the linearity of integration (sum rule) that the policy gradient of $J(\theta)$ is given by (8.21) and the proof is complete. ■

This result states the conditions under which the individual functions, $f_i(s, a)$, in the gradient of an additive objective may be replaced by approximators, $\widehat{f}_i(s, a)$, without introducing bias in the estimate. A large class of problems fall under this umbrella and, as such, many different policy gradients may be derived. In the case of RCPO, the gradient can be recovered by instantiating Theorem 5 as in the corollary below.

Corollary 5.1. *The policy gradient $\nabla_\theta \mathcal{L}(\lambda, \theta)$ may be expressed as*

$$\frac{\partial \mathcal{L}(\lambda, \theta)}{\partial \theta} = \int_{\mathcal{S}} d_{\pi_\theta}(s) \int_{\mathcal{A}(s)} \frac{\partial \pi_\theta(a \mid s)}{\partial \theta} \left[\widehat{Q}(s, a) - \lambda \widehat{C}(s, a)\right] da \, ds, \tag{8.21}$$

where $\widehat{Q}(s, a)$ and $\widehat{C}(s, a)$ are function approximators satisfying the conditions of Theorem 5.

Proof. Instantiate Theorem 5 with $f_1(s, a) = Q_\pi(s, a), c_1 = 1, f_2(s, a) = C_\pi(s, a) - \nu$ and $c_2 = -\lambda$. ■

The updates for NRCPO may now be derived by combining Corollary 5.1 with the NAC [129] framework. This amounts to replacing $\nabla_\theta \mathcal{L}(\lambda, \theta)$ with $\widetilde{\nabla_\theta} \mathcal{L}(\lambda, \theta)$ in Equation 8.8, which can then be expressed as

$$\theta_{k+1} \leftarrow \Gamma_\theta \underbrace{\left[\theta_k + \alpha_\theta(k) \widetilde{\nabla}_\theta \mathcal{L}(\lambda, \theta)\right]}_{\theta_k + \alpha_\theta(k)(w_q - \lambda w_C)}.$$

This algorithm is simple to implement and is also computationally efficient since $\widetilde{\nabla}_\theta(\lambda, \theta)$ can be calculated, for any k, with a single vector-vector addition; i.e. the complexity is linear in $|\theta|$. As a final step, we can integrate the action-value introduced in Section 8.5 into this formulation by assigning $C_\pi(s, a) = \overline{\rho}_\pi(s, a)$. The result is a family of downside risk-averse policy optimisation algorithms, parameterised simply by ν and $\tau_r(s, a)$.

8.6.1 *Convergence*

The convergence of vanilla RCPO, in the absence of function approximation, was proven by Tessler, Mankowitz, and Mannor [172] in the following Theorem.

Theorem 6 (Tessler, Mankowitz, and Mannor [172]). *Assume that the learning rates satisfy $\sum_k^\infty \alpha_\theta(k) = \sum_k^\infty \alpha_\lambda(k) = \infty$, $\sum_k^\infty \left[\alpha_\theta(k)^2 + \alpha_\lambda(k)^2\right] < \infty$ and $\alpha_\lambda(k)/\alpha_\theta(k) \xrightarrow{k\to\infty} 0$. Then, under standard assumptions of iterates and bounded noise [25], the iterates (θ_k, λ_k) converge to a fixed point almost surely.*

Proof. See the original proof of Tessler, Mankowitz, and Mannor [172]. The result derives from standard two-timescale stochastic approximation arguments and the analysis by Borkar [25]. ■

This result can be extended by the same arguments, and the definition of natural gradients [7], to the NRCPO framework. We show this formally in Corollary 6.1 below which also asserts that replacing the vanilla gradient with the natural gradient does not change the solution set, and therefore it even has the same set of fixed points.

Corollary 6.1. *With $\widetilde{\nabla}_\theta \mathcal{L}(\lambda_k, \theta_k)$ in place of $\nabla_\theta \mathcal{L}(\lambda_k, \theta_k)$ in Equation 8.8, the iterates (θ_k, λ_k) converge to a fixed point almost surely under the same assumptions as Theorem 6. Futher, the solution set remains the same.*

Proof. Under the two-timescale assumption of Tessler, Mankowitz, and Mannor [172], we can assume that λ is constant. The ODE for θ now takes the form

$$\nabla_t \theta_t = \Gamma_\theta \left[\widetilde{\nabla}_\theta \mathcal{L}(\lambda, \theta_t)\right],$$

where Γ_θ projects the weights onto the set $\Theta \doteq \prod_{i=1}^k \left[\underline{\theta}_i, \overline{\theta}_i\right]$. Following the standard arguments of Borkar [25], we can consider the learning process as a noisy discretisation of the ODE above, where

$$\theta_{k+1} = \Gamma_\theta \left[\theta_k + \eta_\theta(k) \widetilde{\nabla}_\theta \mathcal{L}(\lambda, \theta_k)\right]. \tag{8.22}$$

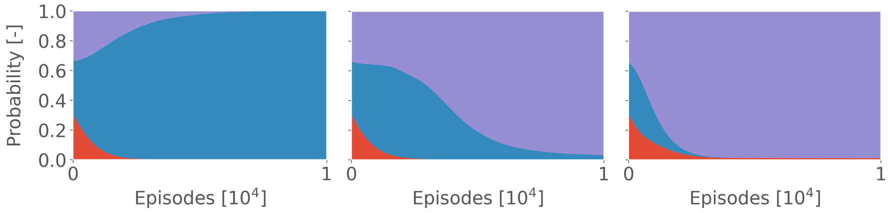
Under stochastic approximation arguments [25], and the fact that the natural gradient here is the steepest ascent direction (see Theorem 1 of Amari [7]), this process converges to a local optimum in the asymptotic limit. The remainder follows directly from the original proof of Tessler, Mankowitz, and Mannor [172], showing that the two-timescale algorithm converges to a local saddle point.

To prove that the solution set is the same, we need only look at the definition of the natural gradient. First, note that $\widetilde{\nabla}_\theta J(\theta) = G^{-1}(\theta) \nabla_\theta J(\theta)$, where $G^{-1}(\theta)$ is the inverse of the Fisher information matrix under the parameterised policy. We know that the fixed points of this approximation scheme, if they exist, are located at the points where $\widetilde{\nabla}_\theta J(\theta) = 0$. Thus, pre-multiplying by $G(\theta)$, we arrive back at the same condition as for the vanilla policy gradient and have thus shown that the solution set is invariant to this transformation. ■

It follows from similar logic to Lemma 1 of Tessler, Mankowitz, and Mannor [172] that NRCPO, a second-order method, also converges on a fixed point that is a feasible solution, assuming: (i) the value $v_\pi(s)$ is bounded for all policies $\pi \in \Pi$; and (ii) every local minimum of $J_c(\theta)$ is a feasible solution. Indeed, following the claims in Corollary 6.1, all results presented by Tessler, Mankowitz, and Mannor [172] carry over to NRCPO under the same conditions.

8.7 EXPERIMENTS

We now present evaluations of the proposed NRCPO algorithm on three experimental domains using variations on $\tau(s, a)$ in the LPM proxy (Equation 8.14). In all cases, the hyper-parameters were chosen through a combination of intuition and trial-and-error.



(a) $J(\theta) = \mathbb{E}[r]$ (b) $J(\theta) = \mathbb{E}[r] - 2\mathbb{M}_{-}[r]$ (c) $J(\theta) = \mathbb{E}[r] - \mathbb{M}_{-}^{2}[r]$

Figure 8.3: Evolution of Boltzmann policies' selection probabilities for arms A (red), B (blue) and C (purple). Each curve represents a normalised average over 100 independent trials.

8.7.1 *Multi-Armed Bandit*

The first problem setting — taken from Tamar, Chow, Ghavamzadeh, and Mannor [169] — is a *3-armed bandit* with rewards distributed according to: $r_A \sim \mathcal{N}(1, 1)$; $r_B \sim \mathcal{N}(4, 6)$; and $r_C \sim \text{Pareto}(1, 1.5)$. The expected rewards from each of the arms are thus 1, 4 and 3, respectively. The optimal solution for a risk-neutral agent is to choose the second arm, but it is apparent that agents sensitive to negative values should choose the third arm since the Pareto distribution's support is bounded from below.

In this experiment we considered a Gibbs policy of the form

$$\pi_\theta(a \mid s) = \frac{e^{\theta_a}}{\sum_{a'} e^{\theta_{a'}}},$$

where each action corresponded to a unique choice over the three arms $a \in \mathcal{A} \doteq \{A, B, C\}$. The value functions were then represented by linear function approximators of the form

$$\widehat{Q}(s, a) = \left\langle \boldsymbol{w}_Q, \nabla_\theta \ln \pi_\theta(a \mid s) \right\rangle + v_Q,$$

and

$$\widehat{\bar{\rho}}(s, a) = \left\langle \boldsymbol{w}_{\bar{\rho}}, \nabla_\theta \ln \pi_\theta(a \mid s) \right\rangle + v_{\bar{\rho}},$$

which are compatible with the policy by construction. The canonical SARSA algorithm was used for policy evaluation with learning rate $\alpha_{\text{Critic}} \doteq 0.005$. The policy updates were performed every 100 samples with a learning rate of $\alpha_{\text{Policy}} \doteq 0.001$.

RESULTS The proposed methods were evaluated by training and evaluating three different Boltzmann policies on the multi-armed bandit problem. The first (Figure 8.3a) was trained using a standard variant of NAC, the latter two (Figure 8.3b and Figure 8.3c) used a stateless version of NRCPO with first and second LPMs as risk measures, respectively; for simplicity, we assume a constant value for the Lagrange multiplier λ. The results show that after ~ 5000 samples, both risk-averse policies have converged on arm C. This highlights: (i) the flexibility of our approach — changing the particular moment and weight in the objective function is trivial; and (ii) the improvements in efficiency that can be gained from incremental algorithms compared to Monte-Carlo methods. See, for example, the approach of Tamar, Chow, Ghavamzadeh, and Mannor [169] which used *10000 samples per gradient estimate*, requiring a total of $\sim 10^5$ sample trajectories before convergence.

8.7.2 *Portfolio Optimisation*

We now consider the portfolio optimisation problem introduced in Section 6.5 using a very similar setup. The policy's likelihood function was defined by a Gibbs distribution and, similarly, we chose a first-order polynomial approximation over the state, as in Equation 6.27. In this case, however, compatible function approximators of the form:

$$\widehat{Q}(s, a) = \left\langle \boldsymbol{w}_Q, \nabla_\theta \ln \pi_\theta(a \mid s) \right\rangle + \left\langle \boldsymbol{v}_Q, \boldsymbol{\phi}(s) \right\rangle, \tag{8.23}$$

and

$$\widehat{\overline{\rho}}(s, a) = \left\langle \boldsymbol{w}_{\overline{\rho}}, \nabla_\theta \ln \pi_\theta(a \mid s) \right\rangle + \left\langle \boldsymbol{v}_{\overline{\rho}}, \boldsymbol{\phi}(s) \right\rangle \tag{8.24}$$

were used. The canonical SARSA(λ) algorithm was used for policy evaluation with learning rate $\alpha_{\text{Critic}} \doteq 0.0001$, discount factor of $\gamma \doteq 0.99$ and accumulating trace with decay rate $\lambda \doteq 1$ (forgive the abuse of notation with respect to the Lagrange multiplier). The policy updates were performed every 200 time steps with $\alpha_{\text{Policy}} = 0.0001$ and the value-function and Lagrange multiplier (with learning rate $\alpha_{\text{Lagrange}} \doteq 0.001$) were pre-trained for 1000 episodes against the initial policy for improved stability.

The portfolio optimisation domain itself was configured as follows: a liquid growth rate of $g^L \doteq 0.005$, illiquid rates $\overline{g}^I = 0.25$ and $\underline{g}^I = 0.05$, and switching probabilities $p_\uparrow \doteq 0.1$ and $p_\downarrow \doteq 0.6$. The probability of default was set to $p_D \doteq 0.1$. Orders sizes were capped at $M \doteq 10$ with a cost per unit of $c \doteq 0.2/M$ and maturity time of $N \doteq 4$ steps. Every episode length was simulated for 50 time steps. These are the same parameters as used previously, repeated here for convenience.

RESULTS Figure 8.4 shows how the performance of our LPM variant of NRCPO performed on the portfolio optimisation problem; in this case the centralised LPM was used as a target, i.e. let $\tau_r(s, a) \doteq r(s, a)$. We observe the emergence of a “frontier” of solutions which trade-off maximisation of the expected return with minimisation of the risk term in the objective. As the threshold ν (see Equation 8.3) increases, thereby increasing the tolerance to risk, so too do we observe a tendency for solutions with a higher mean, higher LPM and more extreme minima. From this we can conclude that minimisation of the proxy (8.14) does have the desired effect of reducing the LPM, validating the *practical value of the bound.*

8.7.3 *Optimal Consumption*

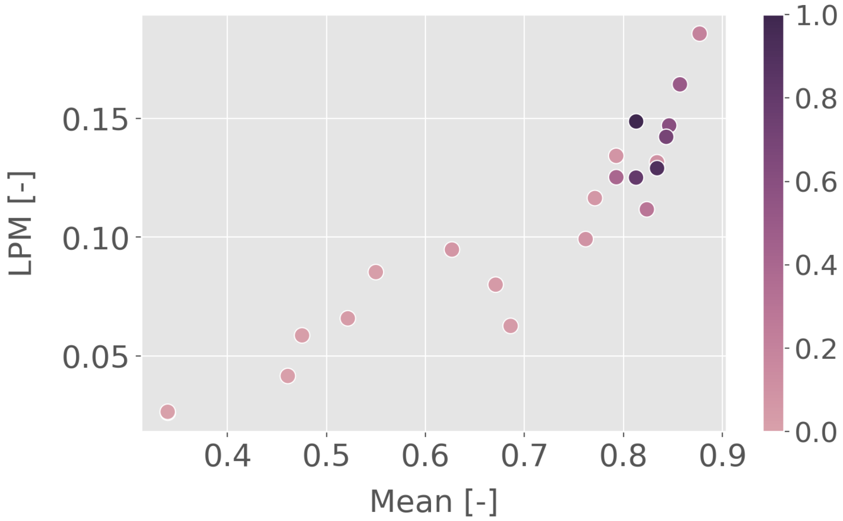
The optimal consumption problem defined in Section 6.6 has an action space with mixed support. To handle this, we used a policy with likelihood function given by the product of a Normal distribution and a Beta distribution,

$$\pi_\theta(\boldsymbol{a} \mid s) = \pi^{(1)}_{\theta_1}(a_1|s)\, \pi^{(2)}_{\theta_2}(a_2|s), \tag{8.25}$$

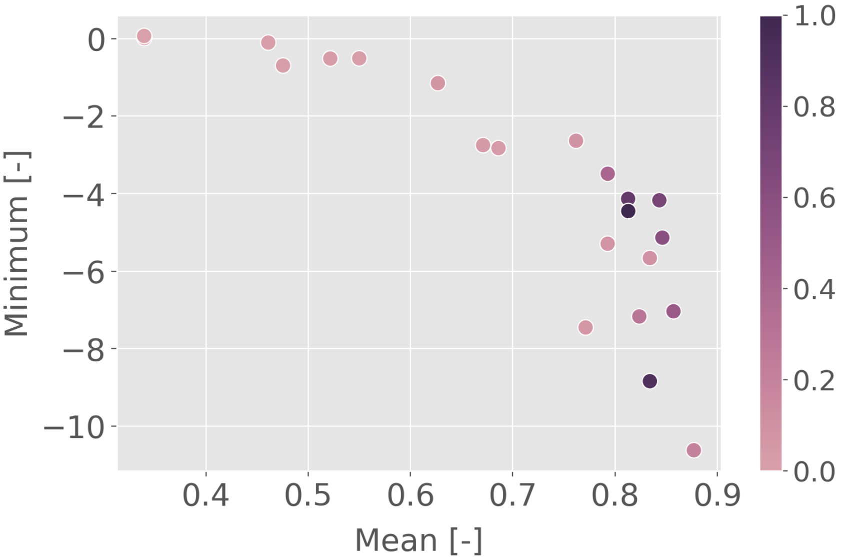
where $\pi^{(1)}_{\theta_1}(a_1|s)$ and $\pi^{(2)}_{\theta_2}(a_2|s)$ are defined in Equation 6.1 and Equation 6.1, respectively. In this case, $\widehat{\mu}$ was represented by a linear function approximator with third-order Fourier basis, and $\widehat{\sigma}$, $\widehat{\alpha}$ and $\widehat{\beta}$ were given by the same as $\widehat{\mu}$ but mapped through a softplus transformation to maintain positivity. Both $\widehat{\alpha}$ and $\widehat{\beta}$ were also shifted by a value 1 to maintain unimodality of the Beta distribution.

The value functions associated with $\pi_\theta(\boldsymbol{a} \mid s)$ were represented by linear function approximators

$$\widehat{Q}(s, a) = \left\langle \boldsymbol{w}_Q, \nabla_\theta \ln \pi_\theta(a \mid s) \right\rangle + \left\langle \boldsymbol{v}_Q, \boldsymbol{\phi}(s) \right\rangle, \tag{8.26}$$



(a) Mean vs. the LPM of returns.



(b) Mean vs. the minimum observed log return.

Figure 8.4: Performance of portfolio optimisation solutions for varying thresholds $\nu \in [0, 1]$. Each point was generated by evaluating the policy over 10^4 trials following training.

and

$$\widehat{\overline{\rho}}(s, a) = \left\langle \boldsymbol{w}_{\overline{\rho}}, \nabla_{\theta} \ln \pi_{\theta}(a \,|\, s) \right\rangle + \left\langle \boldsymbol{v}_{\overline{\rho}}, \boldsymbol{\phi}(s) \right\rangle, \tag{8.27}$$

using the compatible bases of the policy and the same Fourier basis as for the policy. SARSA(λ) was used for policy evaluation with learning rate $\alpha_{\text{Critic}} \doteq 0.00001$, discount factor of $\gamma \doteq 1$ and accumulating trace with decay rate $\lambda \doteq 0.97$. The policy updates were performed every 1000 time steps with $\alpha_{\text{Policy}} \doteq 0.00001$. As in the previous experiment, the value-function and Lagrange multiplier ($\alpha_{\text{Lagrange}} \doteq 0.0025$) were pre-trained for 1000 episodes.

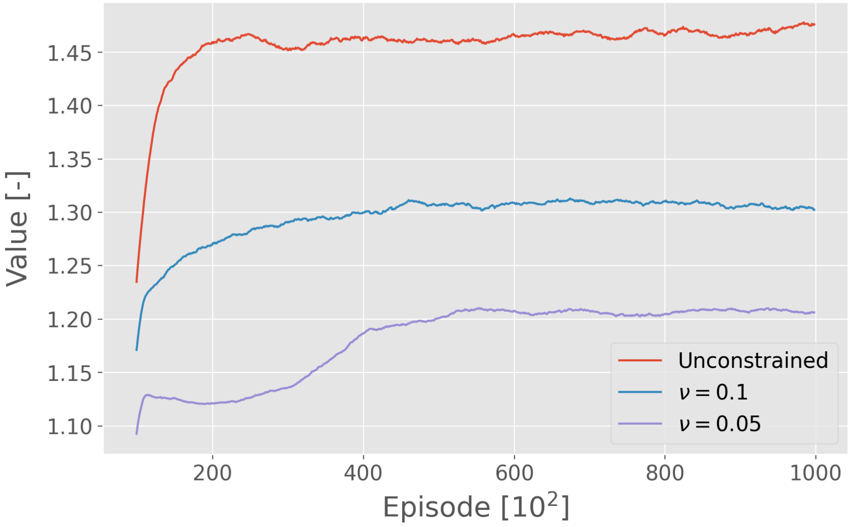
Unlike before, we defined a custom target function by leveraging prior knowledge of the problem: we set $\tau_r(s, a) \doteq W_t \, \Delta t (T - t)$. This has the interpretation of the expected reward generated by an agent that consumes it's wealth at a fixed rate. Unrolling the recursive definition of $\tau(s, a)$, we have an implied target of W_t for all states. In other words, we associate a higher penalty with those policies that underperform said reasonable "benchmark" and finish the episode having consumed less wealth than the initial investment.

The domain itself was configured as follows: a gain of $g^L \doteq 0.05$ for the liquid asset; a risky asset whose price evolved with drift $\mu_t \doteq 1$, volatility $\sigma_t \doteq 0.25$ and random walk W_t. The wealth of the agent was initialised with value $W_0 \doteq 1$ and time increment of $\Delta t \doteq 0.005$. The probability of default at each time step was then set to $p_D \doteq 0.0015$.

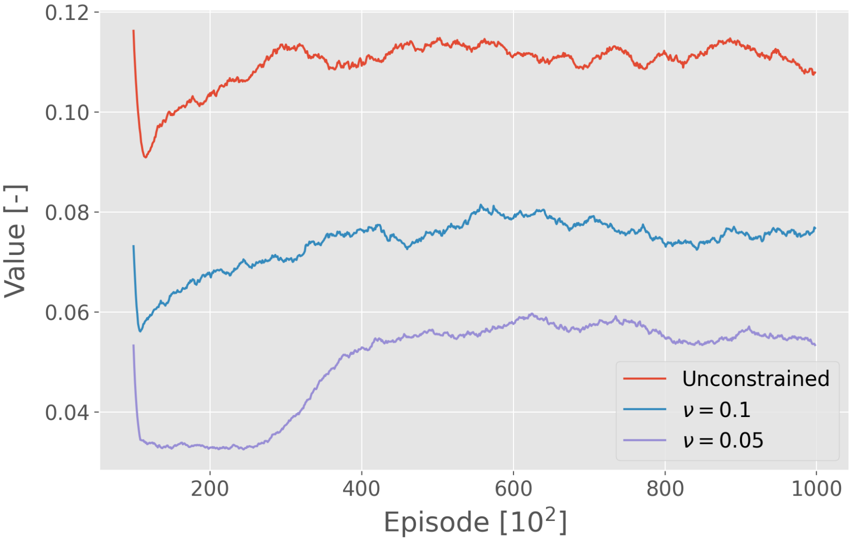
RESULTS Figure 8.5 shows how performance of the NRCPO algorithm evolved during training. Each curve was generated by sampling 100 out-of-sample trajectories every 100 training episodes to estimate performance statistics of the learn policy; both in- and out-of-sample models used the same parameters. As in the previous section, we observe how decreasing ν leads to increasing risk-aversion in the form of a lower mean and LPM. In all cases the algorithm was able to identify a *feasible solution* and exhibited highly stable learning. An important conclusion to take from this is that the flexibility to choose $\tau_r(s, a)$ affords us a great deal of control over the behaviour of the policy. In this case, we *only penalise downside risk associated with losses.* Furthermore, NRCPO removes the need for calibrating the multiplier λ, which can be very hard to tune [4]. This makes the approach *highly practical for many real-world problems.*

8.8 CONCLUSIONS

In this chapter we have put forward two key ideas. First, that partial moments offer a tractable alternative to conventional metrics such as variance or conditional value at risk. We show that our proxy has a simple interpretation and enjoys favourable reward variance. Second, we demonstrate how an existing method in constrained policy optimisation can be extended to leverage natural gradients, an algorithm we call NRCPO. The combination of these two developments is a methodology for deriving downside risk-averse policies with a great deal of flexibility and sample efficiency. In future work we hope to address questions on computational complexity, and establish whether these methods could be applied to multi-agent systems. We also intend to explore the intersection of methods targeting aleatoric risk and those presented in Chapter 7 for handling epistemic uncertainty to address questions around whether the risk-sensitivity learnt via this method also performs well under train-test ambiguity.



(a) $J_r(\theta)$



(b) $J_c(\theta)$

Figure 8.5: Evolution of performance of optimal consumption solutions for $\nu \in \{0.05, 0.1, \infty\}$. Each curve was generated by evaluating the policy for 100 trials every 100 training episodes, with a simple moving average of period 100 applied for clarity.

Part IV

EPILOGUE

9 CONCLUSION

9.1 LOOKING BACK

This thesis has demonstrated two core claims:

(i) **Epistemic uncertainty** — i.e. uncertainty derived from misspecification of the model compared with reality — can be managed effectively through direct LOB reconstruction in a computational efficient manner, and through game theoretic adaptations of parameterised stochastic models; and

(ii) **Aleatoric uncertainty** — i.e. uncertainty that is intrinsic to the problem domain — can be handled through careful construction of reward functions, and through risk-sensitive RL, while still retaining interpretability and simplicity of implementation.

These claims are supported by the data-driven and model-driven analyses conducted in Part II and Part III, respectively. The insights developed were intentionally aligned with the four questions posed at the beginning of the thesis (Section 1.2 of Part I) which are repeated below for posterity. A summary of the conclusions drawn from each chapter as associated with each of the research agendas is then provided in the sections to follow.

	Aleatoric Uncertainty	
Epistemic Uncertainty	**Q1**: How do we exploit data in LOB reconstruction to minimise train-test model ambiguity?	**Q2**: What techniques are required to apply RL to realistic settings and promote risk-sensitivity via the reward function?
	Q3: Is is possible to derive epistemically robust strategies from improperly specified models?	**Q4**: Can risk-sensitive RL be extended to support human-interpretable objectives that aren't possible to specify in the reward?

Q1 It is well understood that full order book reconstruction is a highly accurate method of simulating a financial market. In factual scenarios, there is exactly zero bias between the true events and the reconstruction; up to the limit given by the level of data available (Section 4.2). This allows us to compute important quantities needed to represent the *public* state of the market and even assess the predictive power of each feature over various time horizons. The real limitation of this approach lies in the inability to simulate counterfactual scenarios, as discussed in Section 4.5. Market impact, in particular, is near impossible to accurately account for due to the path dependency of the LOB, regardless of the fidelity of the data. This is a fundamental limitation of market replay. To address this, we proposed a technique called *shadowing* which allows us to replay historical transactions against artificial orders. This can be done pessimistically, optimistically, or in the volume-weighted scheme — and in scenarios where the ego agent's orders are small, it can be very effective. As part of answering this question, we also provided a codebase in C++ that can be used to perform experiments like those conducted in Chapter 5. This has since been used by a number of academics for further research, and is accessible here: https://github.com/tspooner/rl_markets.

Q2 The second question revolved around algorithmic performance and the handling of aleatoric risk in the data-driven setting covered in Part II. The former is addressed in Section 5.6 which examined credit assignment, the objective horizon and bias-variance reduction as a means of improving baseline performance. To begin, we showed that eligibility traces significantly improve robustness and that, in highly stochastic domains such as this, off-policy algorithms can behave very inconsistently; a likely reflection of the deadly triad phenomenon characterised by Sutton and Barto [162]. Another key contribution here was to show that one can overcome the variance of value-based methods due to noisy features by means of factored representations. Through carefully engineered state augmentation — i.e. to use multiple basis functions with varying feature-complexities — we were able to generate highly competitive algorithms compared to state-of-the-art benchmarks.

The second part of Chapter 5 studied constrained behaviour with risk-sensitive reward construction (Section 5.7). The key insight here was to observe that the inventory term in the mark-to-market decomposition (Definition 14) is the main driver of risk. We were thus able to demonstrate that Definition 16 — a function that asymmetrically penalises speculative reward — is a highly effective choice that targets only the negative part of inventory PnL. This approach provides a granular method of tuning the strategy to any level of risk aversion and has since featured in follow-up research, including a paper by Ganesh, Vadori, Xu, Zheng, Reddy, and Veloso [61]. In summary, we have introduced a practical, effective and interpretable family of value-based control algorithms for market making in discrete action-spaces.

Q3 In the second half of the thesis we focussed on techniques for improving the epistemic robustness of strategies that were learnt in a model-based simulation. The real value of this is simple: it allows us to extract as much "bang-for-our-buck" as possible from the assumed market dynamics. Still, it remained an open question as to how one does this in a meaningful way. Our contribution, and answer to research question, was to cast the single-agent learning problem as a zero-sum stochastic game between the trader and the market. The result, as shown in Chapter 7, is that any minimax solution — i.e. Nash equilibrium — will correspond to an adversarially robust trading strategy. A validation was provided by comparing our variant of adversarial reinforcement learning (ARL) to two traditional algorithms, and performing an analysis of the game theoretical aspects of the problem setting. Concretely, our method was shown to dominate in all cases and directly corresponds to the solutions predicted by the single-stage analysis.

Q4 The final question of the thesis asked how one incorporates human-intuitive risk criteria into the objective of an RL agent. Historically the risk-sensitivity in RL has been tackled using symmetric, variance-based measures which penalise behaviours that give rise to a large dispersion in the return distribution [170] or in the stepwise rewards [22]. This, as we have argued, however, fails to align with "human" concepts of risk in trading in which we are only concerned with *downside deviation*. Motivated by the normative concepts of Tversky and Kahneman [176], we proposed partial moments as a theoretical framework for estimating the expected losses against an arbitrary benchmark. We show that the non-linearity can be circumvented using a bound based on the triangle inequality of absolute values. It is shown that this can be learnt in a stable and incremental manner, can be incorporated into existing algorithms for risk-sensitive RL, and results in a natural extension of previous methods that is more suitable to the financial setting.

In the course of answering **Q4**, we were also able to derive an extension of the policy gradient theorem to linearly decomposable objectives. Assuming each of the inner terms are compatible function approximators, then we could show that natural policy gradients translate elegantly to risk-sensitive RL. The resulting algorithm, NRCPO, is a highly efficient algorithm for learning constrained policies in MDPs.

Altogether, these contributions make a significant step towards improved robustness and interpretability of RL-based trading paradigms. While much remains to be done, I firmly believe that the techniques developed in this thesis will have practical use in industry, as well as prompt further interesting research in academia. Some of these directions are outlined in the following, and final section.

9.2 LOOKING AHEAD

While we have addressed some important issues at the intersection of algorithmic trading and reinforcement learning much is still left to be done. The space of ideas is vast and we cannot hope to possibly cover them all here. Given the time, one would of course cover a myriad directions. However, there are a number of key areas that would be of great interest to explore that follow on directly from the contributions of this thesis. In the two sections below we summarise some of these and how they relate to the limitations of the work presented over each of the last two parts.

9.2.1 *Data-Driven Trading*

In the context of data-driven trading, we have covered key questions around credit assignment, time horizons, bias-variance trade-offs, risk-sensitivity and state-augmentation. All of these contributed to a performant final algorithm, but many open questions remain.

MARKET FRICTIONS How do market frictions such as transaction fees and latency affect both the performance of the algorithm and the nature of the optimal solution? This is particularly important to practitioners, and even more so to those that do not operate as a designated market maker. The former has been studied in detail in the stochastic optimal control literature, but the structure of fees is typically somewhat simplified. Exploring the impact of realistic, hierarchical fee structures would provide real insight not only to traders themselves, but also the exchanges who design the market mechanisms. The latter, on the other hand, has not received as much attention. This is primarily because fees mostly affect high-frequency traders who operate at the sub-millisecond level. It stands to reason that building a deep understanding of the impact of latency using realistic, physical network models would be of great importance to the field.

COUNTERFACTUAL SIMULATIONS A key limitation of our data-driven approach — even with shadowing — is the inability to simulate counterfactual scenarios. Relaxing the assumption of negligible market impact adds a great deal of complexity to the simulation. The reason is that path dependency and hysteresis can give rise to very different market conditions when we condition on both the past market state *and* the artificial orders. This is akin to the well-known butterfly effect of dynamical systems. This problem has been identified in other work [44, 179] but it remains unclear whether there is exists a truly robust solution. It is possible that an interpolation-based approach would be effective, but we leave it to future work to explore these areas.

HIERARCHICAL ACTION-SPACES It became increasingly clear during the research that the space of trading strategy is vast. While we maintain that the simplicity of a discrete action-space is valuable, it is also limiting. This was partially addressed in Part III with the extensions to continuous action-spaces. However, hierarchical representations would appear to be a much more natural approach. For example, the agent could choose between different order types at the top-level, and then have a secondary, conditioned policy that solves for the optimal quantity and price to trade at. This relates very closely with the research in RL on parameterised action-spaces [110]. Developing a parsimonious strategy specification that supports hierarchical decision-making would be a fantastic contribution to the area.

EPISTEMIC UNCERTAINTY Another important direction of study would be to explore the application of adversarial RL in a fully data-driven simulation. The presents a number of key challenges around specification, and would exacerbate the issues mentioned previously on the mis-match between simulated and true measures under perturbations. It thus follows that this direction of research could address some of the more important aspects of the trading and simulation. It would also be of great interest to rigorously define the market replay setting in a measure-theoretic framework. This would allows better integration with the insights from Chapter 7 and of Cartea, Donnelly, and Jaimungal [31], of which we postulate there is an equivalence.

9.2.2 *Model-Driven Trading*

While some make use of RL as an intelligent numerical solver for optimal control problems [79], this viewpoint, we argue, is severely short-sighted. As we have seen in Chapter 7 and Chapter 8, there is a rich space of research directions that spawn out of the intersection between stochastic modelling and RL. In this thesis we covered some particularly interesting cases, but we summarise below some extensions to these, and novel directions that would be especially fascinating to explore.

NON-STATIONARITY AND AGENT-BASED MODELLING It is clear from Chapter 7 that introducing non-stationarity to the market making problem considerably increases the complexity of the task, both for theoretical and empirical analysis. However, in the case of two-player zero-sum stochastic games (as studied here), we also have a number of crucial guarantees that ensure, for instance, the existence of equilibria. A fascinating next step would be to look at the dynamics and behaviours that can be obtained in a full multi-agent system. Does the agent become increasingly robust to strategic opponents in the market? What can we say about the existence and stability of solutions in the stochastic game and one-shot projections? A closely related concept is thus task generalisation.

TASK GENERALISATION An emerging paradigm of contextual [81] and parameterised [94] MDPs facilitate the learning of policies that generalise across tasks. These tasks may be known explicitly a priori [90], or may even be based on latent states that the agent must infer [186]. This idea is incredibly powerful, with the latter setting being especially well suited to the trading domain. For example, one could design a policy that conditions it',s behaviour on estimates of a latent regime of the market, such as the time of day (a simple example) or the reversion dynamics of prices (less simple).

These research directions, of course, could be applied in the data-driven setting as well. However, we do suggest that work on these particular questions focus on the model-driven setting where the true dynamics are known in advance and can be controlled. This allows us to explore precisely when and how the methods break down. Once one has established sufficient insight, moving to a data-driven platform is the logical next step.

NON-CONVEX RISK-SENSITIVE RL In Chapter 8 we explored risk-sensitive RL when the risk measure was non-convex due to a rectification operation. To address this we derived a bound and showed that minimising this proxy was sufficient to achieve the desired behaviour. However, this leaves a somewhat sour taste in the mouth. Why can't we simply learn the true risk measure directly using temporal-difference (TD) methods? As discussed by Hasselt, Quan, Hessel, Xu, Borsa, and Barreto [83], Bellman equations can be generalised beyond the linear definitions we usually consider in the literature. The contraction property in this case relies on Lipschitz continuity, but can be satisfied. It would be incredibly valuable to the community to provide a thorough evaluation of methods that are able to learn risk metrics that are not well-behaved but better model human rationality. In the same way that Tamar, Chow, Ghavamzadeh, and Mannor [169] generalised policy gradients to coherent risk measures, is it possible to achieve a similar result while relaxing these assumptions? Furthermore, how do we perform prediction when the non-linearities are present in the value functions?

BIBLIOGRAPHY

[1] Pieter Abbeel and Andrew Y Ng. 'Exploration and Apprenticeship Learning in Reinforcement Learning'. In: *Proc. of ICML*. 2005, pp. 1–8.

[2] Frédéric Abergel, Marouane Anane, Anirban Chakraborti, Aymen Jedidi, and Ioane Muni Toke. *Limit Order Books*. Cambridge University Press, 2016.

[3] Jacob D. Abernethy and Satyen Kale. 'Adaptive Market Making via Online Learning'. In: *Proc. of NeurIPS*. 2013.

[4] Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. 'Constrained Policy Optimization'. In: *Proc. of ICML*. Vol. 70. 2017, pp. 22–31.

[5] Robert Almgren and Neil Chriss. 'Optimal Execution of Portfolio Transactions'. In: *Journal of Risk* 3 (2001), pp. 5–40.

[6] Eitan Altman. *Constrained Markov Decision Processes*. Vol. 7. CRC Press, 1999.

[7] Shun-Ichi Amari. 'Natural Gradient Works Efficiently in Learning'. In: *Neural Computation* 10.2 (1998), pp. 251–276.

[8] Torben G Andersen and Oleg Bondarenko. 'Reflecting on the VPIN Dispute'. In: *Journal of Financial Markets* 17 (2014), pp. 53–64.

[9] András Antos, Csaba Szepesvári, and Rémi Munos. 'Learning near-optimal policies with Bellman-residual minimization based fitted policy iteration and a single sample path'. In: *Machine Learning* 71.1 (2008), pp. 89–129.

[10] Sanjeev Arora, Elad Hazan, and Satyen Kale. 'The multiplicative weights update method: a meta-algorithm and applications'. In: *Theory of Computing* 8.1 (2012), pp. 121–164.

[11] Marco Avellaneda and Sasha Stoikov. 'High-frequency trading in a limit order book'. In: *Quantitative Finance* 8.3 (2008), pp. 217–224.

[12] Louis Bachelier. 'Théorie de la spéculation'. In: *Annales scientifiques de l'École normale supérieure*. Vol. 17. 1900, pp. 21–86.

[13] Leemon Baird. 'Residual Algorithms: Reinforcement Learning with Function Approximation'. In: *Machine Learning*. Elsevier, 1995, pp. 30–37.

[14] Thomas Bayes. 'LII. An essay towards solving a problem in the doctrine of chances. By the late Rev. Mr. Bayes, FRS communicated by Mr. Price, in a letter to John Canton, AMFR S'. In: *Philosophical Transactions of the Royal Society of London* 53 (1763), pp. 370–418.

[15] Marc G Bellemare, Will Dabney, and Rémi Munos. 'A Distributional Perspective on Reinforcement Learning'. In: *Proc. of ICML*. 2017.

[16] Marc G Bellemare, Georg Ostrovski, Arthur Guez, Philip S Thomas, and Rémi Munos. 'Increasing the Action Gap: New Operators for Reinforcement Learning'. In: *Proc. of AAAI*. 2016.

[17] Richard Bellman. *Dynamic Programming*. Priceton University Press, 1957.

[18] Dimitri P Bertsekas. 'Nonlinear Programming'. In: *Journal of the Operational Research Society* 48.3 (1997), p. 334.

[19] Dimitri P Bertsekas and John N Tsitsiklis. *Neuro-Dynamic Programming*. Athena Scientific, 1996.

[20] Shalabh Bhatnagar and K Lakshmanan. 'An Online Actor-Critic Algorithm with Function Approximation for Constrained Markov Decision Processes'. In: *Journal of Optimization Theory and Applications* 153.3 (2012), pp. 688–708.

[21] Joydeep Bhattacharjee. *Practical Machine Learning with Rust: Creating Intelligent Applications in Rust*. Apress, 2019.

[22] Lorenzo Bisi, Luca Sabbioni, Edoardo Vittori, Matteo Papini, and Marcello Restelli. 'Risk-Averse Trust Region Optimization for Reward-Volatility Reduction'. In: *arXiv preprint arXiv:1912.03193* (2019).

[23] Fischer Black and Myron Scholes. 'The pricing of options and corporate liabilities'. In: *Journal of Political Economy* 81.3 (1973), pp. 637–654.

[24] Vivek S Borkar. 'An actor-critic algorithm for constrained Markov decision processes'. In: *Systems & Control Letters* 54.3 (2005), pp. 207–213.

[25] Vivek S Borkar. *Stochastic Approximation: A Dynamical Systems Viewpoint*. Vol. 48. Springer, 2009.

[26] Justin A Boyan. 'Least-Squares Temporal Difference Learning'. In: *Proc. of ICML*. Citeseer. 1999, pp. 49–56.

[27] Justin A Boyan. 'Technical Update: Least-Squares Temporal Difference Learning'. In: *Machine Learning* 49.2-3 (2002), pp. 233–246.

[28] Steven J Bradtke and Andrew G Barto. 'Linear least-squares algorithms for temporal difference learning'. In: *Machine Learning* 22.1-3 (1996), pp. 33–57.

[29] Aseem Brahma, Mithun Chakraborty, Sanmay Das, Allen Lavoie, and Malik Magdon-Ismail. 'A Bayesian market maker'. In: *Proc. of EC*. New York, New York, USA, 2012, pp. 215–232.

[30] René Carmona and Kevin Webster. 'The self-financing equation in limit order book markets'. In: *Finance and Stochastics* 23.3 (2019), pp. 729–759.

[31] Álvaro Cartea, Ryan Donnelly, and Sebastian Jaimungal. 'Algorithmic Trading with Model Uncertainty'. In: *SIAM Journal on Financial Mathematics* 8.1 (2017), pp. 635–671.

[32] Alvaro Cartea, Ryan Donnelly, and Sebastian Jaimungal. 'Enhancing Trading Strategies with Order Book Signals'. In: *Applied Mathematical Finance* 25.1 (2018), pp. 1–35.

[33] Álvaro Cartea and Sebastian Jaimungal. *Order-Flow and Liquidity Provision*. Working Paper. 2015.

[34] Álvaro Cartea and Sebastian Jaimungal. 'Risk Metrics and Fine Tuning of High-Frequency Trading Strategies'. In: *Mathematical Finance* 25.3 (2015), pp. 576–611.

[35] Álvaro Cartea, Sebastian Jaimungal, and Damir Kinzebulatov. 'Algorithmic Trading with Learning'. In: *International Journal of Theoretical and Applied Finance* 19.04 (2016).

[36] Álvaro Cartea, Sebastian Jaimungal, and José Penalva. *Algorithmic and High-Frequency Trading*. Cambridge University Press, 2015.

[37] Álvaro Cartea, Sebastian Jaimungal, and Jason Ricci. 'Buy Low Sell High: A High Frequency Trading Perspective'. In: *SIAM Journal on Financial Mathematics* 5.1 (2014), pp. 415–444.

[38] Tanmoy Chakraborty and Michael Kearns. 'Market Making and Mean Reversion'. In: *Proc. of EC*. 2011, pp. 307–314.

[39] David Chambers and Rasheed Saleuddin. 'Commodity option pricing efficiency before Black, Scholes, and Merton'. In: *The Economic History Review* 73.2 (2020), pp. 540–564.

[40] Nicholas Tung Chan and Christian Shelton. *An Electronic Market-Maker*. Working Paper. 2001.

[41] Nuttapong Chentanez, Andrew G Barto, and Satinder P Singh. 'Intrinsically Motivated Reinforcement Learning'. In: *Proc. of NeurIPS*. 2005, pp. 1281–1288.

[42] Po-Wei Chou, Daniel Maturana, and Sebastian Scherer. 'Improving Stochastic Policy Gradients in Continuous Control with Deep Reinforcement Learning Using the Beta Distribution'. In: *Proc. of ICML*. 2017.

[43] Lonnie Chrisman. 'Reinforcement learning with perceptual aliasing: The perceptual distinctions approach'. In: *Proc. of AAAI*. Vol. 1992. Citeseer. 1992, pp. 183–188.

[44] Hugh L Christensen, Richard E Turner, Simon I Hill, and Simon J Godsill. 'Rebuilding the limit order book: sequential Bayesian inference on hidden states'. In: *Quantitative Finance* 13.11 (2013), pp. 1779–1799.

[45] Dave Cliff. 'ZIP60: an enhanced variant of the ZIP trading algorithm'. In: *IEEE International Conference on E-Commerce Technology*. 2006.

[46] Harald Cramér. *Mathematical Methods of Statistics*. Vol. 9. Princeton University Press, 1946.

[47] Jon Danielsson, Jean-Pierre Zigrand, Bjørn N Jorgensen, Mandira Sarma, and CG de Vries. *Consistent Measures of Risk*. Working Paper. 2006.

[48] Christoph Dann, Gerhard Neumann, Jan Peters, et al. 'Policy Evaluation with Temporal Differences: A Survey and Comparison'. In: *JMLR* 15 (2014), pp. 809–883.

[49] Peter Dayan. 'Reinforcement comparison'. In: *Connectionist Models*. Elsevier, 1990, pp. 45–51.

[50] Peter Dayan and Geoffrey E Hinton. 'Feudal reinforcement learning'. In: *Proc. of NeurIPS*. 1993, pp. 271–278.

[51] Frederik Michel Dekking, Cornelis Kraaikamp, Hendrik Paul Lopuhaä, and Ludolf Erwin Meester. *A Modern Introduction to Probability and Statistics: Understanding Why and How*. Springer Science & Business Media, 2005.

[52] M. A H Dempster and V. Leemans. 'An automated FX trading system using adaptive reinforcement learning'. In: *Expert Systems with Applications* 30.3 (2006), pp. 543–552.

[53] Jan Dhaene, Steven Vanduffel, Qihe Tang, Marc Goovaerts, Rob Kaas, and David Vyncke. *Solvency capital, risk measures and comonotonicity: a review*. Working Paper. 2004, pp. 1–33.

[54] Pei-yong Duan and Hui-he Shao. 'Multiple Hyperball CMAC Structure for Large Dimension Mapping'. In: *Proc. of International Federation of Automated Control* 32.2 (1999), pp. 5237–5242.

[55] David Easley, Marcos M López de Prado, and Maureen O'Hara. 'The Volume Clock: Insights into the High-Frequency Paradigm'. In: *The Journal of Portfolio Management* 39.1 (2012), pp. 19–29.

[56] Albert Einstein. 'On the method of theoretical physics'. In: *Philosophy of Science* 1.2 (1934), pp. 163–169.

[57] Simone Farinelli and Luisa Tibiletti. 'Sharpe thinking in asset ranking with one-sided measures'. In: *European Journal of Operational Research* 185.3 (2008), pp. 1542–1547.

[58] Peter C Fishburn. 'Mean-Risk Analysis with Risk Associated with Below-Target Returns'. In: *The American Economic Review* 67.2 (1977), pp. 116–126.

[59] Pietro Fodra and Mauricio Labadie. 'High-frequency market-making with inventory constraints and directional bets'. In: *arXiv preprint arXiv:1206.4810* (2012).

[60] Yasuhiro Fujita and Shin-ichi Maeda. 'Clipped Action Policy Gradient'. In: *Proc. of ICML*. 2018.

[61] Sumitra Ganesh, Nelson Vadori, Mengda Xu, Hua Zheng, Prashant Reddy, and Manuela Veloso. 'Reinforcement Learning for Market Making in a Multi-agent Dealer Market'. In: *arXiv:1911.05892* (2019).

[62] Javier Garcıa and Fernando Fernández. 'A comprehensive survey on safe reinforcement learning'. In: *JMLR* 16.1 (2015), pp. 1437–1480.

[63] Alborz Geramifard, Michael Bowling, and Richard S Sutton. 'Incremental Least-Squares Temporal Difference Learning'. In: *Proc. of NCAI*. Vol. 21. 1. 2006, p. 356.

[64] Freddie Gibbs. *Freddie Gibbs | The Bootleg Kev Podcast (Episode 1)*. Youtube. 2020. URL: https://youtu.be/aelpi6tahuI?t=476.

[65] J Willard Gibbs. 'Fourier's series'. In: *Nature* 59.1522 (1898), pp. 200–200.

[66] Lawrence R Glosten and Paul R Milgrom. 'Bid, Ask and Transaction Prices in a Specialist Market with Heterogeneously Informed Traders'. In: *Journal of Financial Economics* 14.1 (1985), pp. 71–100.

[67] Dhananjay K. Gode and Shyam Sunder. 'Allocative Efficiency of Markets with Zero-Intelligence Traders: Market as a Partial Substitute for Individual Rationality'. In: *Journal of Political Economy* 101.1 (1993), pp. 119–137.

[68] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 'Generative Adversarial Nets'. In: *Proc. of NeurIPS*. 2014.

[69] Martin D Gould and Julius Bonart. 'Queue Imbalance as a One-Tick-Ahead Price Predictor in a Limit Order Book'. In: *Market Microstructure and Liquidity* 2.2 (2016), p. 1650006.

[70] Martin D Gould, Mason A Porter, Stacy Williams, Mark McDonald, Daniel J Fenn, and Sam D Howison. 'Limit Order Books'. In: *Quantitative Finance* 13.11 (2013), pp. 1709–1742.

[71] Ronald L Graham, Donald E Knuth, Oren Patashnik, and Stanley Liu. 'Concrete Mathematics: A Foundation for Computer Science'. In: *Computers in Physics* 3.5 (1989), pp. 106–107.

[72] Evan Greensmith, Peter L Bartlett, and Jonathan Baxter. 'Variance Reduction Techniques for Gradient Estimates in Reinforcement Learning'. In: *JMLR* 5 (2004), pp. 1471–1530.

[73] Ivo Grondman. 'Online Model Learning Algorithms for Actor-Critic Control'. PhD thesis. Dutch Institute for Systems and Control, 2015.

[74] Ivo Grondman, Lucian Busoniu, Gabriel AD Lopes, and Robert Babuska. 'A survey of actor-critic reinforcement learning: Standard and natural policy gradients'. In: *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 42.6 (2012), pp. 1291–1307.

[75] Sanford J Grossman and Merton H Miller. 'Liquidity and Market Structure'. In: *The Journal of Finance* 43.3 (1988), pp. 617–633.

[76] Marek Grzes and Daniel Kudenko. 'Reward Shaping and Mixed Resolution Function Approximation'. In: *Developments in Intelligent Agent Technologies and Multi-Agent Systems*. 2010. Chap. 7.

[77] Olivier Guéant. 'Optimal market making'. In: *Applied Mathematical Finance* 24.2 (2017), pp. 112–154.

[78] Olivier Guéant, Charles-Albert Lehalle, and Joaquin Fernandez-Tapia. 'Dealing with the Inventory Risk: A solution to the market making problem'. In: *Mathematics and Financial Economics* 7.4 (2011), pp. 477–507.

[79] Olivier Guéant and Iuliia Manziuk. 'Deep reinforcement learning for market making in corporate bonds: beating the curse of dimensionality'. In: *Applied Mathematical Finance* 26.5 (2019), pp. 387–452.

[80] Fabien Guilbaud and Huyen Pham. 'Optimal High Frequency Trading with Limit and Market Orders'. In: *CoRR* abs/1106.5040 (2011).

[81] Assaf Hallak, Dotan Di Castro, and Shie Mannor. 'Contextual markov decision processes'. In: *arXiv preprint arXiv:1502.02259* (2015).

[82] Hado V Hasselt. 'Double Q-learning'. In: *Proc. of NeurIPS*. 2010, pp. 2613–2621.

[83] Hado van Hasselt, John Quan, Matteo Hessel, Zhongwen Xu, Diana Borsa, and Andre Barreto. 'General non-linear Bellman equations'. In: *arXiv preprint arXiv:1907.03687* (2019).

[84] Thomas Ho and Hans R Stoll. 'Optimal Dealer Pricing Under Transactions and Return Uncertainty'. In: *Journal of Financial Economics* 9.1 (1981), pp. 47–73.

[85] Kurt Hornik, Maxwell Stinchcombe, Halbert White, et al. 'Multilayer Feedforward Networks are Universal Approximators'. In: *Neural Networks* 2.5 (1989), pp. 359–366.

[86] Barry Johnson. *Algorithmic Trading & DMA: An introduction to direct access trading strategies*. 4Myeloma Press London, 2010.

[87] Sham M Kakade. 'A Natural Policy Gradient'. In: *Proc. of NeurIPS*. 2001, pp. 1531–1538.

[88] Sham M Kakade. 'A Natural Policy Gradient'. In: *Proc. of NeurIPS*. 2002.

[89] Michael Kearns and Satinder Singh. 'Near-optimal Reinforcement Learning in Polynomial Time'. In: *Machine Learning* 49.2-3 (2002), pp. 209–232.

[90] Taylor W Killian, Samuel Daulton, George Konidaris, and Finale Doshi-Velez. 'Robust and efficient transfer learning with hidden parameter markov decision processes'. In: *Proc. of NeurIPS*. 2017, pp. 6250–6261.

[91] Andrei Kirilenko, Albert S Kyle, Mehrdad Samadi, and Tugkan Tuzun. 'The flash crash: High-frequency trading in an electronic market'. In: *The Journal of Finance* 72.3 (2017), pp. 967–998.

[92] Achim Klenke. *Probability Theory: A Comprehensive Course*. Springer Science & Business Media, 2013.

[93] Richard Klima, Daan Bloembergen, Michael Kaisers, and Karl Tuyls. 'Robust Temporal Difference Learning for Critical Domains'. In: *Proc. of AAMAS*. 2019, pp. 350–358.

[94] George Konidaris and Finale Doshi-Velez. 'Hidden parameter Markov decision processes: an emerging paradigm for modeling families of related tasks'. In: *Proc. of AAAI*. 2014.

[95] George Konidaris, Sarah Osentoski, and Philip Thomas. 'Value Function Approximation in Reinforcement Learning using the Fourier Basis'. In: *Proc. of AAAI.* 2011.

[96] Michail G Lagoudakis and Ronald Parr. 'Least-Squares Policy Iteration'. In: *JMLR* 4 (2003), pp. 1107–1149.

[97] Sophie Laruelle and Charles-albert Lehalle. *Market microstructure in practice.* World Scientific, 2018.

[98] Germain Lefebvre, Maël Lebreton, Florent Meyniel, Sacha Bourgeois-Gironde, and Stefano Palminteri. 'Behavioural and neural characterization of optimistic reinforcement learning'. In: *Nature Human Behaviour* 1.4 (2017), pp. 1–9.

[99] Alexander Lipton, Umberto Pesavento, and Michael G Sotiropoulos. 'Trade Arrival Dynamics and Quote Imbalance in a Limit Order Book'. In: *arXiv preprint arXiv:1312.0514* (2013).

[100] Michael L Littman. 'Markov games as a framework for multi-agent reinforcement learning'. In: *Proc. of ICML.* 1994.

[101] Andrew W Lo. 'The statistics of Sharpe ratios'. In: *Financial Analysts Journal* 58.4 (2002), pp. 36–52.

[102] Donald MacKenzie and Taylor Spears. '"The formula that killed Wall Street": The Gaussian copula and modelling practices in investment banking'. In: *Social Studies of Science* 44.3 (2014), pp. 393–417.

[103] C. J. Maddison, D. Lawson, G. Tucker, N. Heess, M. Norouzi, A. Mnih, A. Doucet, and Y. W. Teh. 'Particle Value Functions'. In: *arXiv preprint arXiv:1703.05820* (2017).

[104] Ananth Madhavan. 'Market Microstructure: A Survey'. In: *Journal of Financial Markets* 3.3 (2000), pp. 205–258.

[105] Hamid Reza Maei and Richard S Sutton. 'GQ(λ): A general gradient algorithm for temporal-difference prediction learning with eligibility traces'. In: *Proc. of AGI.* Atlantis Press. 2010.

[106] Hamid Reza Maei, Csaba Szepesvári, Shalabh Bhatnagar, and Richard S Sutton. 'Toward Off-Policy Learning Control with Function Approximation'. In: *Proc. of ICML.* 2010.

[107] Benoit Mandelbrot and Howard M Taylor. 'On the distribution of stock price differences'. In: *Operations Research* 15.6 (1967), pp. 1057–1062.

[108] Shie Mannor and John N Tsitsiklis. 'Algorithmic Aspects of Mean-Variance Optimization in Markov Decision Processes'. In: *European Journal of Operational Research* 231.3 (2013), pp. 645–653.

[109] Harry M Markowitz. 'Foundations of Portfolio Theory'. In: *The Journal of Finance* 46.2 (1991), pp. 469–477.

[110] Warwick Masson, Pravesh Ranchod, and George Konidaris. 'Reinforcement learning with parameterized actions'. In: *Proc. of AAAI.* 2016, pp. 1934–1940.

[111] Francisco S Melo, Sean P Meyn, and M Isabel Ribeiro. 'An Analysis of Reinforcement Learning with Function Approximation'. In: *Proc. of ICML.* 2008, pp. 664–671.

[112] Robert C Merton. 'Lifetime Portfolio Selection under Uncertainty: The Continuous-Time Case'. In: *The Review of Economics and Statistics* (1969), pp. 247–257.

[113] Robert C Merton. 'Theory of rational option pricing'. In: *The Bell Journal of Economics and Management Science* (1973), pp. 141–183.

[114] John E Moody and Matthew Saffell. ‘Reinforcement Learning for Trading’. In: *Proc. of NeurIPS*. 1999, pp. 917–923.

[115] John Moody and Matthew Saffell. ‘Learning to Trade via Direct Reinforcement’. In: *IEEE Transactions on Neural Networks* 12.4 (2001), pp. 875–889.

[116] John Moody, Lizhong Wu, Yuansong Liao, and Matthew Saffell. ‘Performance Functions and Reinforcement Learning for Ttrading Systems and Portfolios’. In: *Journal of Forecasting* 17.5-6 (1998), pp. 441–470.

[117] Lyndon Moore and Steve Juh. ‘Derivative pricing 60 years before Black–Scholes: evidence from the Johannesburg Stock Exchange’. In: *The Journal of Finance* 61.6 (2006), pp. 3069–3098.

[118] Rémi Munos. ‘A Study of Reinforcement Learning in the Continuous Case by the Means of Viscosity Solutions’. In: *Machine Learning* 40.3 (2000), pp. 265–299.

[119] Yuriy Nevmyvaka, Yi Feng, and Michael Kearns. ‘Reinforcement Learning for Optimized Trade Execution’. In: *Proc. of ICML*. 2006.

[120] Emmy Noether. ‘Invariante Variationsprobleme’. ger. In: *Nachrichten von der Gesellschaft der Wissenschaften zu Göttingen, Mathematisch-Physikalische Klasse* (1918), pp. 235–257.

[121] Abraham Othman. ‘Automated Market Making: Theory and Practice’. PhD thesis. CMU, 2012.

[122] Abraham Othman, David M Pennock, Daniel M Reeves, and Tuomas Sandholm. ‘A Practical Liquidity-Sensitive Automated Market Maker’. In: *ACM Transactions on Economics and Computation* 1.3 (2013), pp. 1–25.

[123] Maureen O’Hara. ‘High Frequency Market Microstructure’. In: *Journal of Financial Economics* 116.2 (2015), pp. 257–270.

[124] Judea Pearl and Dana Mackenzie. *The book of why: the new science of cause and effect*. Basic Books, 2018.

[125] Mark D. Pendrith and Malcolm Ryan. ‘Estimator Variance in Reinforcement Learning: Theoretical Problems and Practical Solutions’. In: 1997.

[126] Carlota Perez. *Technological revolutions and financial capital*. Edward Elgar Publishing, 2003.

[127] Julien Pérolat, Bilal Piot, and Olivier Pietquin. ‘Actor-Critic Fictitious Play in Simultaneous Move Multistage Games’. In: *Proc. of AISTATS*. 2018.

[128] Jan Peters. *Policy Gradient Methods for Control Applications*. Working Paper. 2002.

[129] Jan Peters and Stefan Schaal. ‘Natural Actor-Critic’. In: *Neurocomputing* 71.7-9 (2008), pp. 1180–1190.

[130] Lerrel Pinto, James Davidson, Rahul Sukthankar, and Abhinav Gupta. ‘Robust Adversarial Reinforcement Learning’. In: *Proc. of ICML*. Vol. 70. 2017, pp. 2817–2826.

[131] Doina Precup. ‘Eligibility traces for off-policy policy evaluation’. In: *Computer Science Department Faculty Publication Series* (2000), p. 80.

[132] Doina Precup, Richard S Sutton, and Sanjoy Dasgupta. ‘Off-policy temporal-difference learning with function approximation’. In: *Proc. of ICML*. 2001, pp. 417–424.

[133] Giovanni Walter Puopolo. ‘Portfolio Selection with Transaction Costs and Default Risk’. In: *Managerial Finance* (2017).

[134] Martin L Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, 1994.

[135] Aravind Rajeswaran, Sarvjeet Ghotra, Balaraman Ravindran, and Sergey Levine. 'EPOpt: Learning Robust Neural Network Policies using Model Ensembles'. In: *Proc. of ICLR*. 2017.

[136] C Radhakrishna Rao. 'Information and the accuracy attainable in the estimation of statistical parameters'. In: vol. 20. 1945, pp. 78–90.

[137] Martin Reck. 'Xetra: the evolution of an electronic market'. In: *Electronic Markets* (2020), pp. 1–5.

[138] Herbert Robbins and Sutton Monro. 'A Stochastic Approximation Method'. In: *The Annals of Mathematical Statistics* (1951), pp. 400–407.

[139] Gavin A Rummery and Mahesan Niranjan. *On-line Q-learning Using Connectionist Systems*. Working Paper. Department of Engineering, University of Cambridge, 1994.

[140] *"Rust and machine learning #4: practical tools (Ep. 110)"*. URL: https://datascienceathome.com/rust-and-machine-learning-4-practical-tools-ep-110.

[141] *Rust*. URL: https://www.rust-lang.org.

[142] Moonkyung Ryu, Yinlam Chow, Ross Anderson, Christian Tjandraatmadja, and Craig Boutilier. 'CALQ: Continuous Action Q-Learning'. In: *arXiv preprint arXiv:1909.12397* (2019).

[143] John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. 'High-Dimensional Continuous Control using Generalized Advantage Estimation'. In: *Proc. of ICLR*. 2016.

[144] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 'Proximal Policy Optimization Algorithms'. In: *arXiv preprint arXiv:1707.06347* (2017).

[145] L. Julian Schvartzman and Michael P. Wellman. 'Stronger CDA Strategies through Empirical Game-Theoretic Analysis and Reinforcement Learning'. In: *Proc. of AAMAS* (2009), pp. 249–256.

[146] Eldar Shafir and Robyn A LeBoeuf. 'Rationality'. In: *Annual Review of Psychology* 53.1 (2002), pp. 491–517.

[147] Alexander Shapiro, Darinka Dentcheva, and Andrzej Ruszczyński. *Lectures on Stochastic Programming: Modeling and Theory*. SIAM, 2014.

[148] C R Shelton. 'Importance Sampling for Reinforcement Learning with Multiple Objectives'. PhD thesis. Massachusetts Institute of Technology, 2001.

[149] Yun Shen, Michael J Tobia, Tobias Sommer, and Klaus Obermayer. 'Risk-Sensitive Reinforcement Learning'. In: *Neural Computation* 26.7 (2014), pp. 1298–1328.

[150] Craig Sherstan, Brendan Bennett, Kenny Young, Dylan R Ashley, Adam White, Martha White, and Richard S Sutton. 'Directly Estimating the Variance of the λ-return using Temporal-Difference Methods'. In: *arXiv preprint arXiv:1801.08287* (2018).

[151] Alexander A. Sherstov and Peter Stone. 'Three Automated Stock-Trading Agents: A Comparative Study'. In: *Agent Mediated Electronic Commerce {VI}: Theories for and Engineering of Distributed Mechanisms and Systems (AMEC 2004)*. Vol. 3435. 2004, pp. 173–187.

[152] Alexander A. Sherstov and Peter Stone. 'Function Approximation via Tile Coding: Automating Parameter Choice'. In: *Abstraction, Reformulation and Approximation*. 2005, pp. 194–205.

[153] Satinder P Singh, Tommi Jaakkola, and Michael I Jordan. 'Learning Without State-Estimation in Partially Observable Markovian Decision Processes'. In: *Machine Learning*. Elsevier, 1994, pp. 284–292.

[154] Maurice Sion. 'On General Minimax Theorems'. In: *Pacific Journal of Mathematics* 8.1 (1958), pp. 171–176.

[155] Frank A Sortino and Lee N Price. 'Performance Measurement in a Downside Risk Framework'. In: *The Journal of Investing* 3.3 (1994), pp. 59–64.

[156] Thomas Spooner, John Fearnley, Rahul Savani, and Andreas Koukorinis. 'Market Making via Reinforcement Learning'. In: *Proc. of AAMAS*. 2018, pp. 434–442.

[157] Thomas Spooner, Anne E Jones, John Fearnley, Rahul Savani, Joanne Turner, and Matthew Baylis. 'Bayesian optimisation of restriction zones for bluetongue control'. In: *Scientific Reports* 10.1 (2020), pp. 1–18.

[158] Thomas Spooner and Rahul Savani. 'Robust Market Making via Adversarial Reinforcement Learning'. In: *Proc. of IJCAI*. Special Track on AI in FinTech. July 2020, pp. 4590–4596.

[159] Thomas Spooner and Rahul Savani. 'A Natural Actor-Critic Algorithm with Downside Risk Constraints'. URL: https://arxiv.org/abs/2007.04203.

[160] Thomas Spooner, Nelson Vadori, and Sumitra Ganesh. 'Causal Policy Gradients: Leveraging Structure for Efficient Learning in (Factored) MOMDPs'. In: *arXiv preprint arXiv:2102.10362* (2021).

[161] SM Sunoj and N Vipin. 'Some properties of conditional partial moments in the context of stochastic modelling'. In: *Statistical Papers* (2019), pp. 1–29.

[162] Richard S Sutton and Andrew G Barto. *Reinforcement Learning: An Introduction*. MIT Press, 2018.

[163] Richard S Sutton, Hamid Reza Maei, Doina Precup, Shalabh Bhatnagar, David Silver, Csaba Szepesvári, and Eric Wiewiora. 'Fast Gradient-Descent Methods for Temporal-Difference Learning with Linear Function Approximation'. In: *Proc. of ICML*. 2009, pp. 993–1000.

[164] Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. 'Policy Gradient Methods for Reinforcement Learning with Function Approximation'. In: *Proc. of NeurIPS*. 2000.

[165] Richard S Sutton, Joseph Modayil, Michael Delp, Thomas Degris, Patrick M Pilarski, Adam White, and Doina Precup. 'Horde: A Scalable Real-time Architecture for Learning Knowledge from Unsupervised Sensorimotor Interaction'. In: *Proc. of AAMAS*. 2011, pp. 761–768.

[166] Richard S Sutton, Doina Precup, and Satinder Singh. 'Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning'. In: *Artificial Intelligence* 112.1-2 (1999), pp. 181–211.

[167] Csaba Szepesvári. 'Algorithms for Reinforcement Learning'. In: *Synthesis Lectures on Artificial Intelligence and Machine Learning* 4.1 (2010), pp. 1–103.

[168] Csaba Szepesvári and William D Smart. 'Interpolation-Based Q-Learning'. In: *Proc. of ICML*. 2004, p. 100.

[169] Aviv Tamar, Yinlam Chow, Mohammad Ghavamzadeh, and Shie Mannor. 'Policy Gradient for Coherent Risk Measures'. In: *Proc. of NeurIPS*. 2015, pp. 1468–1476.

[170] Aviv Tamar, Dotan Di Castro, and Shie Mannor. 'Policy Gradients with Variance Related Risk Criteria'. In: *Proc. of ICML*. 2012.

[171] Aviv Tamar, Dotan Di Castro, and Shie Mannor. 'Learning the Variance of the Reward-To-Go'. In: *JMLR* 17.1 (2016), pp. 361–396.

[172] Chen Tessler, Daniel J. Mankowitz, and Shie Mannor. 'Reward Constrained Policy Optimization'. In: *Proc. of ICLR*. 2019.

[173] Philip Thomas. 'Bias in Natural Actor-Critic Algorithms'. In: *Proc. of ICML*. 2014.

[174] Hunter S. Thompson. *The Proud Highway: Saga of a Desperate Southern Gentleman*. Vol. 1. Bloomsbury, 2011.

[175] John N Tsitsiklis and Benjamin Van Roy. 'An analysis of temporal-difference learning with function approximation'. In: *IEEE Transactions on Automatic Control* 42.5 (1997), pp. 674–690.

[176] Amos Tversky and Daniel Kahneman. 'Prospect Theory: An Analysis of Decision Under Risk'. In: *Econometrica* 47.2 (1979), pp. 263–291.

[177] Harm Van Seijen, Hado Van Hasselt, Shimon Whiteson, and Marco Wiering. 'A Theoretical and Empirical Analysis of Expected Sarsa'. In: *IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning*. 2009, pp. 177–184.

[178] Svitlana Vyetrenko, David Byrd, Nick Petosa, Mahmoud Mahfouz, Danial Dervovic, Manuela Veloso, and Tucker Hybinette Balch. 'Get Real: Realism Metrics for Robust Limit Order Book Market Simulations'. In: *arXiv preprint arXiv:1912.04941* (2019).

[179] Svitlana Vyetrenko and Shaojie Xu. 'Risk-Sensitive Compact Decision Trees for Autonomous Execution in Presence of Simulated Market Response'. In: *arXiv:1906.02312* (2019).

[180] Perukrishnen Vytelingum, Dave Cliff, and Nicholas R Jennings. 'Strategic Bidding in Continuous Double Auctions'. In: *Artificial Intelligence* 172.14 (2008), pp. 1700–1729.

[181] Christopher JCH Watkins and Peter Dayan. 'Q-Learning'. In: *Machine Learning* 8.3-4 (1992), pp. 279–292.

[182] Alex Weissensteiner. 'A Q-Learning Approach to Derive Optimal Consumption and Investment Strategies'. In: *IEEE Transactions on Neural Networks* 20.8 (2009), pp. 1234–1243.

[183] Ronald J Williams. 'Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning'. In: *Machine Learning* 8.3-4 (1992), pp. 229–256.

[184] Wing-Keung Wong, Meher Manzur, and Boon-Kiat Chew. 'How rewarding is technical analysis? Evidence from Singapore stock market'. In: *Applied Financial Economics* 13.7 (2003), pp. 543–551.

[185] Xin Xu, Han-gen He, and Dewen Hu. 'Efficient reinforcement learning using recursive least-squares methods'. In: *Journal of Artificial Intelligence Research* 16 (2002), pp. 259–292.

[186] Luisa Zintgraf, Kyriacos Shiarlis, Maximilian Igl, Sebastian Schulze, Yarin Gal, Katja Hofmann, and Shimon Whiteson. ‘VariBAD: A Very Good Method for Bayes-Adaptive Deep RL via Meta-Learning’. In: *arXiv preprint arXiv:1910.08348* (2019).