Computer Science and Artificial Intelligence Laboratory
Technical Report

MIT-CSAIL-TR-2006-051
CBCL-262

July 12, 2006

Dealers, Insiders and Bandits: Learning and its Effects on Market Outcomes

Sanmay Das

massachusetts institute of technology, cambridge, ma 02139 usa — www.csail.mit.edu

CSAIL

Dealers, Insiders and Bandits: Learning and its Effects on Market Outcomes

by

Sanmay Das

A.B. in Computer Science (2001), Harvard College
S.M. in Electrical Engineering and Computer Science (2003), Massachusetts Institute of Technology

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2006

Author . . . .
Department of Electrical Engineering and Computer Science
May 17, 2006

Certified by . . . .
Tomaso Poggio
Eugene McDermott Professor
Thesis Supervisor

Certified by . . . .
Andrew W. Lo
Harris and Harris Group Professor
Thesis Supervisor

Accepted by . . . .
Arthur C. Smith
Chairman, Department Committee on Graduate Students

Dealers, Insiders and Bandits: Learning and its Effects on Market Outcomes

by

Sanmay Das

Submitted to the Department of Electrical Engineering and Computer Science
on May 17, 2006, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

This thesis seeks to contribute to the understanding of markets populated by boundedly rational agents who learn from experience. Bounded rationality and learning have both been the focus of much research in computer science, economics and finance theory. However, we are at a critical stage in defining the direction of future research in these areas. It is now clear that realistic learning problems faced by agents in market environments are often too hard to solve in a classically rational fashion. At the same time, the greatly increased computational power available today allows us to develop and analyze richer market models and to evaluate different learning procedures and algorithms within these models. The danger is that the ease with which complex markets can be simulated could lead to a plethora of models that attempt to explain every known fact about different markets. The first two chapters of this thesis define a principled approach to studying learning in rich models of market environments, and the rest of the thesis provides a proof of concept by demonstrating the applicability of this approach in modeling settings drawn from two different broad domains, financial market microstructure and search theory.

In the domain of market microstructure, this thesis extends two important models from the theoretical finance literature. The third chapter introduces an algorithm for setting prices in dealer markets based on the model of Glosten and Milgrom (1985), and produces predictions about the behavior of prices in securities markets. In some cases, these results confirm economic intuitions in a significantly more complex setting (like the existence of a local profit maximum for a monopolistic market-maker) and in others they can be used to provide quantitative guesses for variables such as *rates of convergence* to efficient market conditions following price jumps that provide insider information. The fourth chapter studies the problem faced by a trader with insider information in Kyle's (1985) model. I show how the insider trading problem can be usefully analyzed from the perspective of reinforcement learning when some important market parameters are unknown, and that the equilibrium behavior of an insider who knows these parameters can be learned by one who does not, but also that the time scale of convergence to the equilibrium behavior may be impractical, and agents with limited time horizons may be better off using approximate algorithms that do not converge to equilibrium behavior.

The fifth and sixth chapters relate to search problems. Chapter 5 introduces models for a class of problems in which there is a search "season" prior to hiring or matching, like academic job markets. It solves for expected values in many cases, and studies the difference between a "high information" process where applicants are immediately told when they have been rejected and a "low information" process where employers do not send any signal when they reject an applicant. The most important intuition to emerge from the results is that the relative benefit of the high information process is much greater when applicants do not know their own "attractiveness," which implies that search markets might be able to eliminate inefficiencies effectively by providing good information, and we do not always have to think about redesigning markets as a whole. Chapter 6 studies two-sided search explicitly and introduces a new class of multi-agent learning problems, *two-sided bandit problems*, that capture the learning and decision problems of agents in matching markets in which agents must learn their preferences. It also empirically studies outcomes under different periodwise matching mechanisms and shows that some basic intuitions about the asymptotic stability of matchings are preserved in the model. For example, when agents are matched in each period using the Gale-Shapley algorithm, asymptotic outcomes are always stable, while a matching mechanism that induces a stopping problem for some agents leads to the lowest probabilities of stability.

By contributing to the state of the art in modeling different domains using computational techniques, this thesis demonstrates the success of the approach to modeling complex economic and social systems that is prescribed in the first two chapters.

Thesis Supervisor: Tomaso Poggio
Title: Eugene McDermott Professor

Thesis Supervisor: Andrew W. Lo
Title: Harris and Harris Group Professor

Preface

Parts of this thesis are based on published work, and some parts are joint work with others. In particular:

- Chapter 3 contains material that appears in *Quantitative Finance* (April 2005), under the title "A Learning Market-Maker in the Glosten-Milgrom Model."

- Chapter 4 contains material presented at the Neural Information Processing Systems Workshop on Machine Learning in Finance (December 2005) under the title "Learning to Trade with Insider Information."

- Chapter 5 represents joint work with John N. Tsitsiklis.

- Chapter 6 is based in part on material (joint with Emir Kamenica) that appears in the *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI 2005)* under the title "Two-Sided Bandits and the Dating Market."

I use the plural pronoun "we" in chapters 5 and 6 and when referring to work from those chapters.

Acknowledgments

I would like to thank my committee – each one of them brings a different perspective to the table, and each of these perspectives has been incredibly useful in shaping my thoughts and this dissertation. Tommy Poggio brought me into a lab full of neuroscientists, computer vision researchers, and statisticians, and made sure that I never felt out of place. He has provided a home-away-from home for many at CBCL, and I am very grateful for his support, his scientific insights, and his ability to see the big picture and the important questions. Andrew Lo got me interested in finance when I started graduate school, and he has continued to be a tremendous source of knowledge and advice. Looking back, I am glad that it is virtually impossible to slip a dicey oversimplification or a lazy piece of analysis by him! Leslie Kaelbling's infectious enthusiasm, her wide grasp of research in machine learning and artificial intelligence, and her willingness to discuss the somewhat abstract philosophy of bounded rationality with a somewhat naïve graduate student have kept me excited about AI in general and work on bounded rationality in particular. John Tsitsiklis has shown me how to think about problems from a different perspective. I am amazed by his ability to frame interesting questions that can be analyzed fruitfully without having to resort to my favorite tool, simulation!

While almost all of the ideas in this thesis are the result of discussions with many people, the fifth and sixth chapters are products of explicit collaborations with John N. Tsitsiklis and with Emir Kamenica respectively. I am grateful for the opportunity to have worked with them. Sayan Mukherjee served as an unofficial advisor for the first half of my graduate career, and I am thankful for his advice and ideas. I am also indebted to Andrea Caponnetto, Jishnu Das, Ranen Das, Jennifer Dlugosz, Emir Kamenica, Adlar Kim, and Tarun Ramadorai for conversations that shaped my thinking on the ideas contained here. I would also like to thank Barbara Grosz and Avi Pfeffer, who supervised my college research projects and set me on the path to graduate school, and Charles Elkan, who taught the first two AI classes I ever took.

I'd like to thank Mary Pat Fitzgerald for always looking out for the students and postdocs at CBCL, and making sure that the ship runs smoothly, and Svetlana Sussman for help and support at LFE. I am grateful to the Department of Brain and Cognitive Sciences for having interesting students, hosting interesting speakers and providing interesting beers

at Department Tea most Fridays. I thank all the members of CBCL, in particular Tony Ezzat, Gadi Geiger, Ulf Knoblich, and Thomas Serre for making it such a wonderful place to be!

My Ph.D. research has been supported by an MIT Presidential Fellowship and by grants from Merrill-Lynch, National Science Foundation (ITR/IM) Contract No. IIS-0085836 and National Science Foundation (ITR/SYS) Contract No. IIS-0112991. Additional support was provided by the Center for e-Business (MIT), DaimlerChrysler AG, Eastman Kodak Company, Honda R&D Co., Ltd., The Eugene McDermott Foundation, and The Whitaker Foundation. I thank the sponsors for continuing to support all kinds of research.

On a personal note, Drew, Emir and Jeff, the rest of the college gang that stayed in Boston, made life much more fun than it would have been otherwise. Dino and Nick have kept life interesting on random occasions in Boston and New York. Bhrigu and Radu, while far away, have been the best friends anyone could wish for. For the second half of graduate school, Jennifer has been my harshest critic and my strongest supporter, and every day I find myself grateful for her. My parents, Ranen and Veena, and my brothers, Saumya and Jishnu, have always been there for me. I cannot imagine a more supportive and loving family.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview

There is no doubt that participants in economic and financial markets are not in fact equivalent to Laplace's demon – that fantastical character with an intellect vast enough that, in Laplace's own words, if it "knew all of the forces that animate nature and the mutual positions of the beings that compose it, ... could condense into a single formula the movement of the greatest bodies of the universe and that of the lightest atom; for such an intellect nothing could be uncertain and the future just like the past would be present before its eyes." Yet, those who traditionally study markets in academia, economists and theorists of finance, business and management, have tried to make their models parsimonious enough that the Laplacian demon for any given domain is not hard to imagine.[1] These models yield elegant solutions, beautiful equations that help us to immediately grasp how the system changes when one parameter moves around. Nevertheless, we might have reached a point of severely diminishing returns in building such models. Where can we go from here?

This thesis argues that we can gain fresh insight into problems involving the interaction of agents in structured environments by building richer models, in which agents have to solve harder problems in order to perform as well as they can. This calls for the expertise of computer science in solving agent decision-problems, in addition to the traditional expertise of mathematics. To quote Bernard Chazelle (2006), in a recent interview, "computer

[1] Glimcher (2003) might be the first person to make the comparison between agents in economic theory and the Laplacian demon.

science ... is a new way of thinking, a new way of looking at things. For example, mathematics can't come near to describing the complexity of human endeavors in the way that computer science can. To make a literary analogy, mathematics produces the equivalent of one-liners – equations that are pithy, insightful, brilliant. Computer science is more like a novel by Tolstoy: it is messy and infuriatingly complex. But that is exactly what makes it unique and appealing – computer algorithms are infinitely more capable of capturing nuances of complex reality in a way that pure mathematics cannot."

The approach suggested here should appeal to classical economics and finance because the agents of this thesis are not irrational – they try to solve problems as well as possible given their informational and computational constraints. At the same time, the very act of building complex models presents new algorithmic problems for agent designers to solve. In addition to the scientific value of modeling, this kind of research can also contribute to the engineering of autonomous agents that can successfully participate in markets, an application that should become increasingly important as electronic markets and digital, autonomous personal assistant agents become more pervasive. This thesis defines a principled approach to studying learning in rich models of market environments, and provides a proof of concept by demonstrating the applicability of this approach in modeling settings drawn from two different broad domains, financial market microstructure and search theory.

The differences between the approach I suggest here and the more traditional approaches to modeling from the literatures of economics, finance, and management are predicated on the existence of complicated learning and decision problems for agents to solve, problems that will usually turn out to be difficult to solve in a fashion that is guaranteed to be optimal. The existence of such problems implies that the models within which agents operate are significantly more complicated than traditional models, which value parsimony above almost all else. While building simplified, stylized models has obviously been a tremendously valuable enterprise in many disciplines, I contend that there are times when this approach risks throwing the baby out with the bathwater, and we should not be averse to building more complex models, even if we lose some analytical tractability, and the comfort associated with the existence of well-defined, unique optimal agent decision processes or algorithms. The question of which algorithm to use is not simple. Some of the learning and decision problems that arise in market contexts are intrinsically hard. They can involve the agent's uncertainty about the underlying model of the world, high degrees of nonstationarity, and dependence on the behavior of other agents.

I will argue in this thesis that we must solve the engineering problem of designing successful algorithms[2] in complex market domains as a first step in the kind of analysis I propose. The problems of study are then inverse problems at the agent and societal levels.

- What kinds of learning and decision-making algorithms are successful in which types of markets or social settings? Can we create algorithms that allow an agent to be successful across a broad range of models and parameters?
- What is the impact on societal dynamics of different learning and decision-making procedures used by agents?

In addition to providing a principled means of studying equilibrium outcomes in markets, this methodology also implicitly allows for analysis of market dynamics, instead of just static outcomes. Thus the "market outcomes" I discuss cover the whole range of market dynamics, not just steady state behavior. Of course, the idea of studying learning problems in market domains is in itself not novel. The novelty is in a willingness to consider models in which agent decision problems are complex enough that they cannot necessarily be solved optimally, but we can still design good algorithms for them, algorithms which are better than any others that are known for the same problems. The second chapter will make this approach and description much more precise and place it in the context of the existing literatures in computer science, economics, finance, and cognitive science.

The third and fourth chapters look in detail at extensions of two canonical models of market microstructure, a model of market-making due to Glosten and Milgrom (1985), and Kyle's (1985) model of insider trading. The third chapter describes an algorithm that can be used by a market-maker for setting prices in a dealer market. A market-maker serves as a liquidity provider in such markets and must quote bid and ask prices at which she is willing to buy and sell stocks at all times. Glosten and Milgrom framed the problem abstractly, but this chapter provides a real implementation of the price-setting equations that can be used to set dollar-and-cent prices and study market properties. The fourth chapter studies the problem of how to trade optimally when you possess superior information about the value of a stock. Kyle solved the insider's problem in a stylized model, and I

[2] The question of what "successful" means is addressed in detail in the second chapter.

extend his model to consider a harder problem that arises when the insider is not aware of some important environmental parameters. The insider can still learn the equilibrium strategy, but may be better off using an approximate strategy when the horizon is limited, and this strategy may have different implications for markets.

Chapters 5 and 6 are focused on a different problem domain, namely search (in the decision-theoretic and economic sense of the word). Chapter 5 studies a search problem from the perspective of a single decision-maker who expects offers to appear probabilistically in a fixed time frame and must decide on each offer as soon as it appears. In particular, this chapter examines how well an optimal decision-maker performs in the search process as a function of the level of information available to her about what her options may be. This chapter actually departs a little bit from the major theme of the thesis in studying a somewhat more circumscribed problem where optimal behavior can at least be computed (although closed-form solutions are rarely available) based on the agent's beliefs. However, this chapter touches on the applicability of this problem to understanding systems of two-sided search which are not nearly as tractable. Chapter 6 actually studies a model of two-sided search in the context of a "dating market" in which men and women repeatedly go out on dates with each other. The decision problems for agents become very hard and it is difficult to define "optimal" behavior, but this chapter does allow us to gain some insights into the importance of the matching mechanism from the two-sided perspective, to complement the one-sided perspective from Chapter 5.

Throughout the thesis, the focus is on computer science problems of algorithms for online learning and sequential decision-making as well as on the social dynamics and outcomes of these algorithms. The algorithms developed in this thesis relate closely to work in online learning, reinforcement learning and dynamic programming.

1.2 Motivation

An understanding of when market institutions fail to achieve efficient outcomes is one of the most important goals of economics and, more generally, of social science. While the study of markets has traditionally been the domain of economics and finance theory, disciplines like computer science are becoming increasingly important because they provide unorthodox tools for studying dynamic, learning and evolutionary behavior. The fields of economics and finance are defined in part by the study of equilibria derived by assuming rationality among agents who participate in the markets. In recent years there have been two movements that depart from the assumptions of these models. One of these shifts has been to move away from models of perfectly rational agents to more realistic models such as behavioral models, which attempt to replicate the kinds of behavior observed in humans, and models of bounded rationality, which impose computational and other constraints on the decision-making routines of market participants. The other important shift has been a renewed emphasis on nonequilibrium dynamics and the explicit modeling of the process of trading and trading frictions. Computational modeling has become a central tool in studying nonequilibrium dynamics and the process of trading. Exact and closed-form solutions to the equations that define market dynamics are often difficult to find except in the simplest models. Approximation and simulation become the methods of choice for characterizing market properties in such situations.

One of the goals of this thesis is to contribute to the understanding of market imperfections and market dynamics through explicit models of learning agents that participate in the markets. This would not only solve outstanding problems, it would also establish the success of the methodology of modeling bounded rationality through the study of learning behavior and add to the growing literature on agent-based simulation of market dynamics. The best way to model markets through learning agents is to develop algorithms that perform successfully in the environments they are created for. Thus, another goal of this thesis is to develop algorithms that can be used by market participants in real and artificial markets.

1.3 Bounded Rationality

The methodology used by economists defines the field as distinctly as the problems it studies. One of the core underlying assumptions that allows analytical derivation of equilibrium in many problems is that of unbounded rationality and optimizing behavior on the part of agents. This is a problematic assumption in humans, and even in machines when problems reach a certain level of complexity.

John Conlisk (1996) surveys the evidence for bounds on rationality at two levels — rationality tests on single individuals that show humans are subject to biases and the use of heuristics, and "confounded" evidence from markets in which conventional economic theory does not explain market anomalies and theories of bounded rationality provide possible reconciliations. Brian Arthur (1994) identifies an important set of situations in which perfect deductive rationality breaks down easily, namely multi-agent situations, in which agents may not be able to rely on other agents to behave perfectly rationally, and are thus forced to guess the behavior of the other agents. He defines "complexity economics" as the study of questions of how "actions, strategies or expectations might react in general to — might endogenously change with — the aggregate patterns these create" (Arthur 1999). What kinds of models should one build to augment or replace the standard models of markets built on the assumptions of perfect rationality and individual optimization?

The local and global levels of evidence of bounds on rationality indicate at least two directions one can follow in modeling systems of interacting economic agents. One is to directly model the system as a whole. Approaches in this category are typically based on statistical mechanics. The growing field of econophysics falls in this paradigm, changing the focus from the decision problems of agents to the modeling of aggregate behavior of systems in which not much intelligence is required or assumed of the individual components (Mantegna and Stanley 2000, Johnson et al. 2003, *inter alia*). According to Jenny Hogan (2005), writing in the *New Scientist*, "While economists' models traditionally regard humans as rational beings who always make intelligent decisions, econophysicists argue that in large systems the behaviour of each individual is influenced by so many factors that the net result is random, so it makes sense to treat people like atoms in a gas."

Vince Darley (1999) contrasts this "macroscopic" approach with the "microscopic" approach in which the agent is the unit of analysis and the model is built by examining each interaction and behavior at all levels. At the agent level there is no unifying theory of bounded rationality (Gigerenzer and Selten 2001, Conlisk 1996, Arthur 1994, *inter alia*). Gigerenzer and Selten (2001) point out that researchers in psychology, economics, artificial intelligence and animal biology, among other fields, have all studied the problem and proposed "solutions" of various forms. They also note that the term "bounded rationality" itself has come to mean different things to different people. These meanings include constrained optimization, where the costs to perfect optimization outweigh the benefits; irrationality, mostly in the sense of fallacious reasoning about probabilities and expectations; and the use of heuristics. For Gigerenzer and Selten, the first two are not truly about bounded rationality because they treat optimization as the norm and study deviations from it as abnormalities or approximations. In this thesis I will use the term "bounded rationality" in the most inclusive sense, to discuss any departure from full calculative rationality.

Many of the models of economics fall somewhere between the two extremes of micro- and macro-scopic modeling of systems. For example, the two "gold standard" models of financial market microstructure, the Glosten-Milgrom model of market-making (Glosten and Milgrom 1985) and Kyle's model of insider trading (Kyle 1985) use sophisticated game theoretic and optimizing models of market-makers and traders with monopolistic insider information respectively, but model the rest of the order flow with simple stochastic models. On the other hand, econophysics models typically use low-intelligence models of agents and focus more on the dynamics of the collective behavior of these agents, trying to show how complex phenomena can arise from simple agents and simple interactions, using the tools of statistical mechanics.

This thesis presents models of markets with sophisticated economic agents rather than econophysics models of low-intelligence market participants. However, in the argument over the "right" level of sophistication to use in modeling market participants, I will relax the stringent requirements of economic theory and argue that agents that are *engineered to adapt to their environments and perform "successfully," if not optimally* are the next step to take in trying to understand market outcomes. On a completely speculative note, this approach and its relation to the econophysics movement might be thought of as parallel to the discussion of Highly-Optimized Tolerance (HOT) vs. complex adaptive systems (CAS) by Carlson and Doyle (2002). HOT emphasizes a view of system complexity based on engineering and design, as opposed to complexity emerging "between order and disorder." Similarly, my approach views market dynamics as arising from the complex interactions of intelligent agents, rather than emerging from the inherent complexities of interactions between random agents.

1.4 Market Dynamics

Economics in the conventional sense is the study of patterns of behavior in equilibrium. Arthur (1999) claims that "conventional economic theory chooses not to study the unfolding of patterns its agents create, but rather to simplify its questions in order to seek analytical solutions." The questions that sometimes get short shrift concern the path dynamics of the system. Is it possible that the system will remain out-of-equilibrium? If there are multiple equilibria, under what conditions will one get selected rather than the others? For example, suppose new positive information about a stock is relayed to all the participants in a stock market. We know that the traded price should go up to reflect the new information. However, what process will it follow in its rise? Will the increase be orderly and in small increments or will there be a sudden jump? How will the price process be affected by different possible market structures? Computational modeling is an ideal tool for studying the problems which arise very naturally when one thinks about the process rather than just the outcome. In each of the particular markets I look at in this thesis, I analyze the dynamics of market behavior in detail. The agent-based simulation approach is a natural methodology that enables such analysis, because we can look at path dynamics of systems as a natural part of the research.

1.5 Online Learning and Sequential Decision-Making

The algorithmic questions that arise in the solution of the kinds of problems agents must solve in complex, uncertain market environments fall under the rubric of online learning and sequential decision-making, possibly in multi-agent environments. Research in these areas has been widespread in many different fields, including reinforcement learning (Sutton and Barto 1998, Kaelbling et al. 1996), stochastic control and neuro-dynamic programming (Bertsekas and Tsitsiklis 1996), learning theory (Freeman and Saad 1997, Hu and Wellman 1998), multi-agent learning (Stone and Veloso 2000), and the theory of learning in games (Fudenberg and Levine 1998).[3]

Reinforcement learning focuses on the problem faced by an agent who must learn through trial-and-error interaction with the environment it is placed in while its actions in turn affect the environment. The only feedback an agent receives is through the reward it gets after selecting each action. This reward could conceivably only be received after a whole sequence of actions. Some of the major concerns of the literature which are important to this thesis include the exploration-exploitation tradeoff, which asks when an agent

[3] The citations here are to textbooks and important surveys, with the exception of the online learning citations, for which I am not aware of a classic survey.

should forgo immediate reward to learn more about its environment so that it can make better decisions in the future, learning from delayed rewards, and learning with continuous state and action spaces.

In real markets, agents have to learn “online” without having access to either a complete model of the environment or an offline simulator that they can use to collect simulated experience. Kakade (2003) calls this the most realistic and challenging reinforcement learning environment, and notes that it is much harder to create successful algorithms in this setting. Explicitly, the problem becomes one of maximizing the (possibly discounted) sum of rewards up to some future horizon (possibly infinite). Neuro-dynamic programming often deals with this problem in the context of stochastic control and value function approximation. An important issue in maximizing total reward is solving the exploration-exploitation dilemma – when is it right to take a myopically suboptimal action because of the expected value of learning more about another action? Online learning can also become important in contexts where a learning algorithm forms part of a decision-making system and it is impractical to memorize each interaction with the environment and then run a batch learning algorithm to learn the best model at each time step. Instead, it becomes critical to efficiently use each example interaction with the environment to quickly and robustly update agent beliefs.

The theory of learning in games has achieved theoretical credibility for its ability to forecast the global outcomes of simple individual learning rules in multi-agent settings. The multi-agent learning community often focuses on simple general algorithms with guarantees for restricted classes of games (Stone and Veloso 2000, Conitzer and Sandholm 2003). While the theory is powerful in many ways, the algorithms necessary for good performance in the kinds of games and models I examine in this thesis are often too complicated for the present state of the art in providing theoretical guarantees. Even if we could provide theoretical guarantees, they might prove useless in real markets – proving that something is no worse than one-sixth as good as the optimal in terms of expected utility is not nearly as useful as empirically demonstrating performance within a few percentage points.

1.6 Contributions

I hope that readers of this thesis will be convinced that the kind of modeling and analysis it advocates is, in fact, the right approach to thinking about many problems in market environments. I attempt to present computational and simulation models of markets that make novel predictions while at the same time staying well-grounded in basic economic and decision-theoretic notions. Throughout this work I try to maintain an underlying focus on market *structures* and how they impact market outcomes. Therefore, in terms of applications, this thesis should be able to more realistically suggest the impact of structural changes to market institutions. The important specific technical contributions of the thesis in different domains are as follows. The next section of this chapter provides more detailed descriptions of the individual chapters.

- Chapter 3 introduces the first method for explicitly solving the Glosten-Milgrom equations for real prices, and produces predictions about the behavior of prices in securities markets. It makes both the engineering contribution of a first step towards a practical market-making algorithm in the academic literature, and various predictions about market properties that could not have been made without the existence of such an algorithm. In some cases, these results confirm economic intuitions in a significantly more complex setting (like the existence of a local profit maximum for a monopolistic market-maker) and in others they can be used to provide quantitative guesses for variables such as *rates of convergence* to efficient market conditions following price jumps that provide insider information.

- Chapter 4 makes a connection between a classic line of literature on insider trading in market microstructure and artificial intelligence problems of reinforcement learning by relaxing some assumptions from the finance models. It shows the learnability of equilibrium behavior under certain conditions in a standard model of insider trading first introduced by Kyle, but it also shows that the time scale of convergence to the equilibrium behavior may be impractical, and agents with limited time horizons may be better off using approximate algorithms that do not converge to equilibrium behavior. This result may have implications for both the design of trading agents and for analysis of data on insider trading.

- Chapter 5 studies a class of search problems in which there is a search "season" prior to hiring or matching, like academic job markets. It introduces different models and solves for expected values in many cases, and studies the difference between a "high information" process where applicants are immediately told when they have been rejected and a "low information" process where employers do not send any signal when they reject an applicant. The most important intuition to emerge from the analysis is that the relative benefit of the high information process is much greater when applicants do not know their own "attractiveness," which implies that search markets might be able to eliminate inefficiencies effectively by providing good information, and we do not always have to think about redesigning markets as a whole.
- Chapter 6 studies two-sided search explicitly and introduces a new class of multi-agent learning problems, *two-sided bandit problems*, that capture the learning and decision problems of agents in matching markets in which agents must learn their preferences. It also empirically studies outcomes under different periodwise matching mechanisms and shows that some basic intuitions are preserved in the model.

1.7 Thesis Overview

Chapter 2 seeks to connect the literatures from artificial intelligence, economics, and cognitive science to make the case that not only is the notion of bounded optimality from the AI literature the right goal for agent design, it can also serve as a principled means for modeling boundedly rational agents in complex systems like economic markets. While appealing, this goal leaves open two critical questions. First, bounded optimality is defined over an expected set of problems the agent might face and it is not obvious what criterion to use for expected problems. Second, it will typically be impossible to design a provably boundedly optimal agent even given a set of expected problems, because the agent design problem itself is intractable. These problems become particularly important when agents must learn from their environments. In order to deal with these questions we may need to abandon the formalism of mathematics and instead look towards the process of science and engineering. I argue that it is critical to evaluate agents in terms of the expected set of problems they would face if they were deployed in the real world, either in software or in hardware, and the agent programs we use for modeling should be the best known program for any given problem subject to the broader expected set of problems the algorithm might be expected to solve – the algorithm we would choose to use if we had to hand over control of our own behavior in that domain to an artificial agent.

Chapter 3 develops a model of a learning market-maker by extending the Glosten-Milgrom model of dealer markets. The market-maker tracks the changing true value of a stock in settings with informed traders (with noisy signals) and liquidity traders, and sets bid and ask prices based on its estimate of the true value. The performance of the market-maker in markets with different parameter values is evaluated empirically to demonstrate the effectiveness of the algorithm, and the algorithm is then used to derive properties of price processes in simulated markets. When the true value is governed by a jump process, there is a two regime behavior marked by significant heterogeneity of information and large spreads immediately following a price jump, which is quickly resolved by the market-maker, leading to a rapid return to homogeneity of information and small spreads. I also discuss the similarities and differences between this model and real stock market data in terms of distributional and time series properties of returns.

The fourth chapter introduces algorithms for learning how to trade using insider (superior) information in Kyle's model of financial markets. Prior results in finance theory relied on the insider having perfect knowledge of the structure and parameters of the market. I show in this chapter that it is possible to learn the equilibrium trading strategy when its form is known even without knowledge of the parameters governing trading in the model. However, the rate of convergence to equilibrium is slow, and an approximate algorithm that does not converge to the equilibrium strategy achieves better utility when the horizon is limited. I analyze this approximate algorithm from the perspective of reinforcement learning and discuss the importance of domain knowledge in designing a successful learning algorithm.

Chapter 5 examines a problem that often arises in the process of searching for a job or for a candidate to fill a position. Applicants do not know if they will receive an offer from any given firm with which they interview, and, conversely, firms do not know whether applicants will definitely take positions they are offered. In this chapter, we model the search process as an optimal stopping problem with probabilistic appearance of offers from the perspective of a single decision-maker who wants to maximize the realized value of the offer she accepts. Our main results quantify the *value of information* in the following sense: how much better off is the decision-maker if she knows each time whether an offer appeared or not, versus the case where she is only informed when offers actually appear? We show that for some common distributions of offer values she can expect to receive very close to her optimal value even in the lower information case as long as she knows the probability that any given offer will appear. However, her expected value in the low information case (as compared to the high information case) can fall dramatically when she does not know the appearance probability *ex ante* but must infer it from data. This suggests that hiring and job-search mechanisms may not suffer from serious losses in efficiency or stability from participants hiding information about their decisions unless agents are uncertain of their own attractiveness as employees or employers.

While Chapter 5 makes inferences about two-sided search processes based on results from considering the more tractable one-sided process, Chapter 6 uses simulation techniques to study a two-sided problem directly. Specifically, we study the decision problems facing agents in repeated matching environments with learning, or *two-sided bandit problems*, and examine the dating market, in which men and women repeatedly go out on dates and learn about each other, as an example. We consider three natural matching mechanisms and empirically examine properties of these mechanisms, focusing on the asymptotic stability of the resulting matchings when the agents use a simple learning rule coupled with an epsilon-greedy exploration policy. Matchings tend to be more stable when agents are patient in two different ways — if they are more likely to explore early or if they are more optimistic. However, the two forms of patience do not interact well in terms of increasing the probability of stable outcomes. We also define a notion of regret for the two-sided problem and study the distribution of regrets under the different matching mechanisms.

Finally, Chapter 7 summarizes the main results presented in this thesis. While each chapter contains ideas for future research in its specific area, Chapter 7 casts a wider net and suggests some important broad directions for future research.

Chapter 2

On Learning, Bounded Rationality, and Modeling

2.1 Introduction

Human beings regularly make extremely complex decisions in a world filled with uncertainty. For example, we decide which gas station to refuel at, and whether to put money into a bank account or a retirement fund, with surprisingly little effort. Designing algorithms for artificial agents in similar situations, however, has proven extremely difficult. Historically, research in artificial intelligence has focused on designing algorithms with the ability to solve these problems in principle, given infinite computational power (Russell 1997). Many problems that arise in everyday decision-making are likely to be impossible to solve perfectly given computational constraints, so this kind of *calculative rationality*, as Russell calls it, is not a particularly interesting practical goal. In order to progress towards designing an intelligent system, we need a better theory of decision-making and learning when rationality is bounded by computational resource considerations. Not only is such a theory critical to our understanding of the nature of intelligence, it could also be applied to study interaction between agents in increasingly realistic models of social and economic systems by serving as a replacement for the classical economic theory of unbounded rationality. Russell (1997) discusses two possible theories, *metalevel rationality* and *bounded optimality*. Russell and Norvig (2003) also consider a more general notion of bounded rationality in the tradition of Herbert Simon's (1955) satisficing. Bounded optimality is in many ways the most appealing of these theories, since it strives to replace agents that always make rational *decisions* with rational *agent programs*. Bounded optimality could also be a more plausible model of how humans work. We know that human beings do not make rational decisions, but it is possible that our brains are an optimal or near-optimal decision-making system for the environments we inhabit.

One of the great attractions of bounded optimality is that it can serve both as a definition of intelligence meeting the needs of research in AI and as a formal model of decision-making to replace unbounded rationality. Unfortunately, while bounded optimality might be the right goal to strive for in agent design, achieving this goal, or even knowing whether it can be achieved, is difficult when agents are uncertain about the world. This issue becomes particularly important in the context of agents that are not fully informed about the environment they are placed in, and who must learn how to act in a successful manner. The question of what is meant by boundedly rational learning and how it can be analyzed is a difficult one that I discuss in some detail below.

The broad plan of this chapter is as follows.

- I start by considering various perspectives on bounded rationality. While Russell (1997) and Parkes (1999) provide plenty of detail on most of the perspectives related to agent design, I will consider in more detail the tradition of bounded rationality research started by Herbert Simon with the notion of satisficing, and continued in the program of fast and frugal heuristics by Gerd Gigerenzer and others (Gigerenzer and Goldstein 1996, Gigerenzer and Selten 2001, *inter alia*).

- After that, I move on to considering how the field of machine learning, and in particular, reinforcement learning, needs to adapt in order to move towards the goal of designing boundedly optimal agents for increasingly complex domains.

- In the last major section, I discuss how a theory of boundedly optimal agents may provide a compelling replacement for the unboundedly rational agents of economic theory. The notion of bounded optimality can serve as a principled approach to modeling complex systems and address many of the criticisms of bounded rationality research found in the economics literature.

2.2 Bounded Rationality and Agent Design

The agent-based approach to AI involves designing agents that "do the right thing" (Russell 1997). The "right thing" in this context means that agents should take actions that maximize their expected utility (or probability of achieving their goals). Ideally, an agent should be perfectly rational. Unfortunately, there is really no such thing as a perfectly rational agent in the world. As Russell (1997, pp. 6) says, "physical mechanisms take time to process information and select actions, hence the behavior of real agents cannot immediately reflect changes in the environment and will generally be suboptimal." *Calculative rationality*, the ability to compute the perfectly rational action in principle, given sufficient time and computational resources, is not a useful notion, because agents that act in the world have physical constraints on when they need to choose their actions. We are left to contemplate other options for agent design.

The heuristics and biases program made famous by Kahneman and Tversky studies actual human behavior and how it deviates from the norms of rational choice. This program is not in any way prescriptive, as it mainly focuses on cataloging deviations from the presumed normative laws of classical decision theory. Thus, this program does not provide any suitable *definitions* for intelligence that we can work towards, although understanding human deviations from decision-theoretic norms might prove informative in the design of good algorithms, as I shall argue later.

Another approach from the literature of cognitive science is the use of "satisficing" heuristics in the tradition of Simon (1955), who introduced the notion that human decision-makers do not exhaustively search over the space of outcomes to choose the best decision, but instead stop as soon as they see an outcome that is above some satisfactory threshold "aspiration level." Conlisk (1996) cites various papers in the economics literature that start from Simon's notion of bounded rationality, and claims that, within economics, "the spirit of the idea is pervasive." In cognitive science and psychology, Gigerenzer and others have recently popularized the use of "fast and frugal" heuristics and algorithms as the natural successor to satisficing. Gigerenzer and Goldstein (1996) state their view of heuristics as being "ecologically rational" (capable of exploiting structures of information present in the environment) while nevertheless violating classical norms of rationality. They have a program to design computational models of such heuristics, which are "fast, frugal and simple enough to operate effectively when time, knowledge, and computational might are limited" while making it quite clear that they do not agree with the view of heuristics as "imperfect versions of optimal statistical procedures too complicated for ordinary minds to carry out" (Goldstein and Gigerenzer 2002). They reject the Kahneman-Tversky program because it maintains the normative nature of classical decision theory.

It is interesting that in their 1997 paper, Gigerenzer and Goldstein held a simulated contest between a satisficing algorithm and "rational" inference procedures, and found that the satisficing procedure matched or outperformed more sophisticated statistical algorithms. The fact that a simple algorithm performs very well on a possibly complex task is not surprising in itself, but what is very clear is that if it can be expected to perform *better* than a sophisticated statistical algorithm, it must either be a better inference procedure or have superior prior information encoded within it *in the context of the environment in which it is tested*. As agent designers, if we had to design an agent that solves a given range of problems, and had access to the information that a satisficing heuristic was the best known algorithm for that range of problems, it would be silly not to use that algorithm in the agent, all else being equal. While I will return to this issue later in this section, the problem with fast and frugal heuristics as a program for agent design is the loose definition of what constitutes a satisfactory outcome, or of what kinds of decision-making methods are "ecologically rational" in the language of Goldstein and Gigerenzer. How do we know that one heuristic is better than another? What if our agent has to perform many different tasks?

Russell proposes two other options for a goal for agent design – *metalevel rationality* and *bounded optimality*. Metalevel rationality involves reasoning about the costs of reasoning. An agent that is rational at the metalevel "selects computations according to their expected utility" (Russell 1997). This is what Conlisk (1996) refers to as *deliberation cost*, and both Russell and Conlisk make the explicit connection to the analogous *value of information*. Conlisk argues strongly for incorporating deliberation cost into optimization problems that arise in economics, saying that human computation is a scarce resource, and economics is by definition the study of the allocation of scarce resources. He suggests that instead of optimally solving an optimization problem P, a decision-maker should solve an augmented problem $F(P)$ in which the cost of deliberation is taken into account. The problem, as he realizes, is that it is also costly to reason about $F(P)$, and, therefore, theoretically at least, one should reason about $F(F(P)), F(F(F(P))), \ldots$. This infinite regress is almost always ignored in the literature that does take deliberation cost into account, assuming that $F(P)$ is, in some sense, a "good enough" approximation.

Economics is not alone in this myopic consideration of deliberation cost. In the AI literature, the tradition of studying deliberation cost (or the essentially equivalent concept which Russell calls the *value of computation*) as part of solving a decision problem dates back to at least the work of Eric Horvitz (1987), and almost all the algorithms that have been developed have used myopically optimal metareasoning at the first level, or shown bounds in very particular instances. The history of metalevel "rationality" in the AI literature is more that of a useful tool for solving certain kinds of problems (especially in the development of anytime algorithms) than as a formal specification for intelligent agents. This is mostly because of the infinite regress problem described above – as (Russell 1997, page 10) writes (of the first metalevel, or the problem $F(P)$ in Conlisk's notation), "perfect rationality at the metalevel is unattainable and calculative rationality at the metalevel is useless."

This leaves us with Russell's last, and most appealing, candidate – *bounded optimality*, first defined by Horvitz (1987) as "the optimization of [utility] given a set of assumptions about expected problems and constraints on resources." Russell (1997) says that bounded optimality involves stepping "outside the agent" and specifying that the agent *program* be rational, rather than every single agent decision. An agent's decision procedure is bounded optimal if the expected utility of an action selected by the procedure is at least as high as that of the action selected by any decision procedure subject to the same resource bounds in the same environment. Of course, we are now requiring a lot of the agent designer, but that seems to make more sense as a one-time optimization problem than requiring the same of the agent for every optimization problem it faces. The problem with bounded optimality is that, for any reasonably complex problem, it will prove incredibly hard to design a boundedly optimal agent, and it might also be hard to prove the optimality of the agent program. We have shifted the burden of rationality to the designer, but the burden still exists.

Nevertheless, bounded optimality seems to be the right goal. If one agent algorithm can be shown to perform better than another given the same beliefs about the set of problems the agent may face and the same computational resources, the first algorithm should be the one used. Of course, this can be problematic because it is not clear *where* it must perform better. What if the prior beliefs of the agents are completely wrong? Again, the only solution is to analyze this from the perspective of the agent designer. Rodney Brooks (1991) argues that the real world is its own best model. While I do not endorse the notion that the task of modeling by simplifying is therefore useless, I do think we should hold boundedly optimal algorithms to the standard of reality. The performance of algorithms should be tested on problems that are as realistic as possible.

Let me use this for a quick foray into the broader question of what we should be doing when we design algorithms intended to be boundedly optimal. Let me use this for a quick foray into the broader question of what we should be doing *as artificial intelligence researchers* when we design algorithms intended to be boundedly optimal. The two points that (Brooks 1991, page 140) makes about the direction of AI research are (quoting directly):

- We must incrementally build up the capabilities of intelligent systems, having complete systems at each step of the way and thus automatically ensure that the pieces and their interfaces are valid.
- At each step we should build complete intelligent systems that we let loose in the real world with real sensing and real action. Anything less provides a candidate with which we can delude ourselves.

Brooks carries this forward to argue for robotics and immersion in the real *physical* world as the most important research program in AI. Oren Etzioni (1993) argues against this view, arguing that the creation of so-called softbots, agents embedded in software, is perhaps a more useful focus for AI research, since it allows us as researcher to tackle high-level issues more immediately without getting sidetracked. The concept of designing boundedly optimal agents is not necessarily tied to either of these research directions. While my later focus on economic modeling falls more in line with Etzioni's arguments than Brooks', the basic goal of agent design is the same in both areas, and I believe that both are useful programs of research in AI. This goal is to solve the engineering problem of creating an agent that will be optimally successful in the environments in which we as agent designers expect it to be placed given the resources available to the agent. The agent could be a software agent attempting to trade optimally in an online market, or it could be a robot attempting to navigate many different kinds of terrain, or almost anything else one can think of.

One critical caveat needs to be mentioned – eventually, we would like to create agents that are generally intelligent across a range of domains and problems (holding a conversation, ordering food from a restaurant, buying a plane ticket, and so on). These agents will have limited computational resources, and therefore, they might not be able to spend a lot of these resources on any one task. They will need to be boundedly optimal in the context of all the problems they have to solve, and therefore, they might have to be designed so that they would not be boundedly optimal for any one of those problems given the same computational resources. This caveat will also apply to modeling economic and social systems. We cannot assume too much in the way of computational resources or exclusivity of access to these resources when we model using the methodology I propose. In fact, suppose the human brain itself is boundedly optimal for the environment it inhabits with respect to some kind of evolutionary survival utility function (or fitness function). Then the kinds of deviations we see from decision-theoretic norms could be explained by the fact that even if an algorithm were not optimal for a particular class of problems, if the agent could perhaps face a significantly larger class of problems, or a better inference procedure were significantly more costly, it might be boundedly optimal to program the agent with that algorithm since this would save valuable resources that could be devoted to other problems. In fact, some of the algorithms described as "fast and frugal heuristics" (Gigerenzer and Goldstein 1996, Goldstein and Gigerenzer 2002) may well be bounded optimal in the larger context of the human brain.

2.3 Bounded Optimality and Learning

In the tradition of economics and decision theory, rational learning is understood to mean Bayesian updating of beliefs based on observations. This definition finesses two major problems we encounter in agent design.

- It shifts responsibility onto the designer's beliefs about the set of problems the agent might face. It could be very hard to design an appropriate prior for an agent that must make many different kinds of decisions in the world.
- Even if the designer's prior beliefs are correct, full Bayesian learning is computationally infeasible in all but the simplest cases. There may be no single algorithm that

provably always outperforms others in selecting optimal actions within the specified computational limits, or finding such an algorithm may be hard.

First, let us consider a simple illustrative problem that is an extension of the classical supervised learning framework to a situation where a predictive agent receives utility from making correct predictions while her actions have no influence on the environment. Later we will turn to the even more difficult case in which the agent's actions impact its environment.

Learning to Predict: Consider a situation in which an agent, call her Mary, receives as input a real-valued vector X at each time step, and has to predict Y, where Y is known to be a (probabilistic or noisy) function of X. The two standard cases are regression, where Y is real valued, or classification, where Y takes on one of a few specific different values. For simplicity, consider the *binary classification* case where $Y \in \{0, 1\}$. Suppose Mary's task is to predict whether Y will be 0 or 1. Immediately after she makes her prediction, she is informed of the true value of Y and receives utility 1 for making the correct prediction, and 0 for making the wrong prediction. Suppose Mary knows that she will see 100 such examples, and her goal is to maximize the utility she receives over the course of this game. How should she play? The theory of unbounded rationality and "rational learning" as used by economics would say that Mary starts with a prior $\Pr(Y|X)$, makes her decision based on the particular instantiation $X = x$ that she sees (predicting $Y = 1$ if $\Pr(Y = 1|X = x) > \Pr(Y = 0|X = x)$ and $Y = 0$ otherwise), and then updates her estimate $\Pr(Y|X)$ using Bayes' rule after the true value of Y is revealed to her. Unfortunately, even ignoring the issue of how to specify a good prior, performing the full Bayesian updates at each step is computationally prohibitive. This brings us into the sphere of bounded optimality. What method would achieve as high a utility as possible for Mary given reasonable computational resources? This question does not have a definite answer, even if we clearly specify the exact resource constraints on the agent. Many different algorithms for the supervised learning problem, both online and offline, could be applied in this situation. Mary could memorize all the examples she sees and use a support vector machine to learn a classifier after each step (even then, she would need to make choices about kernels and parameters), or she could use an ensemble classifier like boosted decision trees, or an online algorithm like the Widrow-Hoff procedure, but there is no one algorithm that is clearly better than the others in terms of expected utility across domains.

2.3.1 Learning How to Act

An agent that acts in the world gains utility from the actions it takes and also learns about the world through the effects of its actions. It must balance exploration (taking myopically suboptimal actions in order to learn more) with exploitation of what it has learned. Perfectly rational Bayesian learning is only possible for certain families of prior beliefs and extremely simple problems. The discovery of an optimal algorithm that balances exploration and exploitation for even the "simple" case of the multi-armed bandit problem was hailed as almost miraculous when it was first published.

The multi-armed bandit problem (Berry and Fristedt 1985, Gittins and Jones 1974, *inter alia*) is often considered the paradigmatic exploration-exploitation problem, because the tradeoff can be expressed simply. A single agent must choose which arm of a slot machine to pull at each time step, knowing that the arms may have different reward distributions. The goal of the agent is to maximize the discounted sum of payoffs. It turns out there is actually an optimal way to play the multi-armed bandit under the assumptions of stationary reward distributions and geometric discounting (Gittins and Jones 1974). The optimal action at any time is to play the arm with the highest Gittins index, a single number associated with each arm that is based solely on the history of rewards associated with that arm. The Gittins index is reasonably easy to compute for certain families of reward distributions, but can be difficult in other circumstances.

The multi-armed bandit is an easy problem, though, compared to the kinds of problems faced by agents in realistic environments. For example, it is hard to define "optimality" in a nonstochastic bandit problem, where the reward for each arm at each time period may have been picked by an adversary (Auer et al. 2002), or a nonstationary case, where the reward distribution for each arm may change over time. Or consider the case where there are multiple people playing the bandit and the arms of the bandit themselves have agency and must try to maximize their own reward, which is dependent on who pulls them in each period (the "two-sided" bandit problem (Das and Kamenica 2005)). Even solving problems with a Markovian structure can be extremely hard, especially when they are not fully observable.

2.3.2 Evaluating Learning Algorithms

While a whole line of literature stresses the importance of resource bounds in understanding computational limitations on rationality in many decision-making environments, the problem becomes particularly severe when the agent does not have a perfect model of its environment and must learn this model through experience. The scenarios examined above raise two fundamental questions for the field of machine learning which parallel those in the introductory discussion. First, if we are seeking to design a boundedly optimal learning algorithm, what should constitute the expected set of problems against which it is evaluated? Second, even given such a set of problems, what should we do if we cannot find a boundedly optimal algorithm, or prove its optimality? I will defer discussion of the second question to the next section, because it will be important for understanding how we can use the notion of bounded optimality in modeling complex systems.

Of course it is impossible to definitively answer these questions, but it is important to keep them in mind as researchers. I believe that it is critical to evaluate algorithms from the perspective of performance in the real world, given the expected set of problems an agent would face if it were deployed in the world, either physically or as a software agent. While there is of course value to the traditional computer science program of proving worst case bounds and evaluating algorithms on arbitrary problem spaces, at some stage (not necessarily at the very initial stage of development) algorithms that are designed to be parts of intelligent agents must face the discipline of the real world.

The problem of learning how to act when an agent gets rewards from its interactions with the environment has been studied extensively (especially in the Markovian framework) in the reinforcement learning and neurodynamic programming literatures. However, analysis has typically focused on the ability of algorithms to eventually learn the true underlying model, and hence the asymptotically optimal decision procedure, and not on the expected utilities achieved by algorithms in potentially limited interactions with the environment. This expected utility viewpoint becomes especially important in the true agent design problem, because our goal must be to design agents that can take the actual costs of exploration and learning into account. An agent does not have access to an offline model of its environment so that it can improve its performance before acting in the world. The agent must be able to make tradeoffs in an online manner, where failure or poor performance immediately impacts the agent negatively. It is therefore critical from the perspective of AI research to attack problems of learning how to act from the perspective of expected utility received in the world, especially keeping in mind the real costs of exploration. It will probably prove natural to adopt a Bayesian perspective in analyzing this issue. To quote (Berry and Fristedt 1985, pp. 4) , in their discussion of bandit problems, "[it] is not that researchers in bandit problems tend to be 'Bayesians'; rather, Bayes's theorem provides a convenient mathematical formalism that allows for adaptive learning, and so is an ideal tool in sequential decision problems."

2.4 Bounded Optimality and Modeling

Historically, research on the outcomes of interaction between self-interested optimizing agents has been the domain of economic theory. Economists place a high value on analytical tractability and model parsimony. They tend to simplify models until agent behavior and interactions can be reduced to a set of equations that provide intuition to the person analyzing the system. In the words of Brian Arthur (1999), "conventional economic theory chooses not to study the unfolding of the patterns its agents create, but rather to simplify its questions in order to seek analytical solutions." These simplified questions have no need of the notion of bounded optimality, because the decision (and learning) problems faced by agents are "easy" in the sense that they can be solved efficiently without using excessive computational resources. If we move to more complicated models, we will necessarily have to examine more difficult agent decision problems, and we need to think about agent decision-making differently. In a seminal paper, Herbert Simon (1955) identified the problem:

> Broadly stated, the task is to replace the global rationality of economic man with a kind of rational behavior that is compatible with the access to information and the computational capacities that are actually possessed by organisms, including man, in the kinds of environments in which such organisms exist. One is tempted to turn to the literature of psychology for the answer. Psychologists have certainly been concerned with rational behavior, particularly in their interest in learning phenomena. But the distance is so great between our present psychological knowledge of the learning and choice processes and

> the kinds of knowledge needed for economic and administrative theory that a marking stone placed halfway between might help travelers from both directions to keep to their courses.

How about bounded optimality as a milestone? With the increasing availability of huge amounts of computational power on every researcher's desktop, we can now analyze models using computational tools, and so it is no longer absolutely critical to simplify models to the extreme. This means we can study models in which agent decision problems are no longer easy and computational resource constraints must be taken into account. This is particularly important, as noted above, in circumstances where agents are not perfectly informed of the structure or the parameters of the environments in which they are placed. How should this kind of modeling proceed so as not to fall into the traps that have made bounded rationality research anathema to many mainstream economists?

John Conlisk (1996) both raises and answers many of the criticisms of bounded rationality research. Perhaps the most important and frequent such criticism is that the decision procedures modeled by those conducting the research are ad hoc. Conlisk summarizes this argument as follows "Without the discipline of optimizing models, economic theory would degenerate into a hodge podge of ad hoc hypotheses which cover every fact but which lack overall cohesion and scientific refutability" [pp. 685]. He goes on to say that discipline comes from good scientific practice, not strict adherence to a particular approach, and suggests that modeling bounded rationality based on deliberation cost would enforce a certain discipline. Conlisk's preferred approach is the equivalent of metalevel rationality, but I would venture to propose that bounded optimality might be both a better model for the purpose of enforcing discipline (given the infinite regress problem) and a more satisfying model of the processes of actual decision-makers in the world. Maybe humans and economic firms *do* take the best actions available, given their capacities for reasoning about these actions and the knowledge of their environments available to them.

However, from the discussion above, it is clear that very often we will not know if an agent algorithm is in fact boundedly optimal or not. Does this invalidate the idea that bounded optimality can serve as a principled replacement for unbounded rationality in economic models? I would argue that this is not the case, but we have to turn to good practice in science and engineering rather than relying on the formalism of mathematics. We should strive to engineer good algorithms for complicated problems, attempting to be rational ourselves in the design of these algorithms. The choice of how to model an agent that is part of a complex system should be made by using the same agent that we would use if we had to engineer an agent for maximally successful performance in that system given our beliefs about the system. Hopefully we could eventually reach more and more realistic models, benefiting both algorithm development and hence the eventual goal of building an intelligent agent as well as our understanding of complex social and economic systems. Let me finish by presenting a case study.

2.4.1 Learning in Economic Models

Many economic models already account for learning in an explicit manner. There is nothing particularly novel about suggesting that agents learn from the environment around them. In an interview with Thomas Sargent, Evans and Honkapohja (2005) explore many of the issues related to how the program called "learning theory" originated in the macroeconomics literature. Let me briefly summarize Sargent's overview of the development of this literature in order to draw a parallel to the overall argument of this chapter.

Learning theory in macroeconomics originated in some ways as a response to rational expectations economics. In the rational expectations literature, there exists what Sargent calls a "communism of models" in which all the agents share the same model of the world, and this is the true model, or "God's model".[1] There is no place for different beliefs in rational expectations theory. Margaret Bray and David Kreps started a research program to show what would happen if you endowed agents with different beliefs, learning algorithms, and data on what had happened in the past. Agents should continue to update their models and then optimize based on their current beliefs. The interesting outcome was that in many cases, the only possible outcomes of the system were close to rational expectations equilibria. In some cases, the learning models helped to eliminate possible rational expectations equilibria because they could not be reached through the learning dynamic. Further research along these lines has started to examine what happens to equilibria when agents can have model uncertainty, not just parameter uncertainty. Along with equilibrium selection, the theory has also contributed much by helping to understand the rates of reaching equilibrium in different problems and in characterizing the situations

[1] Sargent uses the term "model" to mean probability distributions over all the inputs and outputs of the larger economic model.

where the system dynamics show serious deviations from expected equilibrium behavior. Similar research has now become an integral part of the game theory literature as well (Fudenberg and Levine 1998).

How does this relate to bounded rationality? Learning behavior is not necessarily "irrational," so why would we have to think of it any differently? Sargent echoes this manner of thinking in talking about "robust control" (in which an agent explicitly has doubts about her model of the world and must take these into account in decision-making) when he says that it is not a type of bounded rationality because "[the agent's] fear of model misspecification is out in the open" which makes her smarter than a rational expectations agent (Evans and Honkapohja 2005). The problem that arises with this belief is that it is *impossible* to do provably optimal learning in a model in which there is any kind of complexity to the agents' beliefs. We must take computational resource constraints into account, and agents cannot be unboundedly rational. It becomes very hard to even define rationality meaningfully in most of these situations. To quote Horvitz (1987), "[Constraints in resources] can transform a non-normative technique into the 'preferred choice' of devout Bayesians, and can convert the strictest formalists into admirers of heuristics." We have to think about what constitutes a "good" learning algorithm and whether it makes sense (both from the agent-design and the modeling perspectives) to endow the agent with that algorithm. The learning literature in economics typically focuses on very simple methods of least-squares learning which might not be the choice we would make as agent designers if we had to write an algorithm to participate in the world we are modeling.

2.5 Conclusion

This chapter attempts to link the literatures of artificial intelligence, economics, and cognitive science so that those familiar with any of these disciplines will be able to see the parallels and connections easily. I have argued for a particular methodology for both designing artificial agents and for modeling agents that are participants in a complex system like an economic market. This methodology is to start by assuming bounded optimality, or the rationality of the agent program, as the goal in both cases. Since this goal needs to be defined in terms of the expected set of problems an agent will face, we should design agents that would perform successfully in the real world, and expect that the set of problems the agent will face is the set of problems it would encounter in the world. Finally, we will not necessarily ever know or be able to show that an agent we have designed is boundedly optimal. We might have to replace our desire for proving that an agent *is* boundedly optimal with a more scientific or engineering based approach, in which we try to design the best algorithm *so far developed* for a problem given the computational and other constraints on the algorithm.

To conclude with an example, suppose we were to design an agent that took care of our finances. We would want it to successfully trade stocks and bonds, perhaps even foreign exchange, while at the same time taking care of more mundane tasks like maintaining bank accounts and paying mortgages. It is not far-fetched to think that we will in the near future be able to design an agent that is close to boundedly optimal for this problem. Eventually we might be able to use insights from the design of this financial agent to build a truly intelligent agent, but in the meanwhile, if we are happy deploying the agent to take care of our finances, we should use it as our model of an economic decision-making agent when we model financial markets.

Chapter 3

A Learning Market-Maker

3.1 Introduction

The detailed study of equity markets necessarily involves examination of the processes and outcomes of asset exchange in markets with explicit trading rules. Price formation in markets occurs through the process of trading. The field of market microstructure is concerned with the specific mechanisms and rules under which trades take place in a market and how these mechanisms impact price formation and the trading process (O'Hara 1995, Madhavan 2000).

The first market I examine in detail is a stylized version of a dealer market — a market in which two-sided prices are set by a market-maker, (bid and ask prices, at which the market-maker is willing to buy and sell shares respectively) and traders can buy or sell stocks to the market-maker at these quoted prices. The problem I analyze here is the dealer's decision problem under conditions of asymmetric information. Suppose that the dealer knows that certain traders ("informed traders") have better information than she does about the true underlying value of the stock. She does not know if any given trader is better informed than she is, but she does know the distribution of informed and uninformed traders in the market. How should she set bid and ask prices in this model?

The canonical work on this problem in the market microstructure literature is that of Glosten and Milgrom (1985). However, Glosten and Milgrom only solve the price-setting equations they derive in extremely simplistic cases (for example, the stock can have two underlying values, "high" and "low") which remain analytically tractable, and it is very hard to derive any general predictions about market properties from their results.

What if the market-maker had to actually compute prices in a more realistic situation? The real world imposes certain bounds on what kinds of actions a market-maker can take and what kinds of reasoning capacities a market-maker has. In particular, quoted prices must be in integral units, and the market-maker does not have unlimited memory of everything that has happened. Further, the market-maker must make rapid online revisions of its beliefs about the true price of a stock. What constitutes optimal behavior in this situation? I will describe a method that a market-maker can use to actually set prices in a more realistic framework than that of Glosten and Milgrom, and consider what this implies for price processes in real markets. Thus, in this chapter, I present an algorithm for explicitly computing approximate solutions to the expected-value equations for setting prices in an extension of the Glosten-Milgrom model with probabilistic shocks to the underlying true price and noisy informed traders. I validate the algorithm by showing that it produces reasonable market-maker behavior across a range of simulations, and use the algorithm to study the time series and distributional properties of returns and compare them to real stock market data.

The model can also be used to study the impact of different parameters on market properties, and a particularly interesting result that emerges is that there is a two regime behavior in which extreme heterogeneity of information immediately following a jump in the true value (characterized by high spreads and volatility) is quickly resolved and the market returns to a state of homogeneous information characterized by low spreads and volatility.

In this model, price-taking informed and uninformed traders interact through a price-setting market-maker. Informed traders receive a (potentially noisy) signal indicating the true underlying value of the stock and make buy and sell decisions based on the market-maker's quotes and the signal they receive. The true value receives periodic shocks drawn from a Gaussian distribution. Market-makers receive no information about the true value of the stock and must base their estimates solely on the order flow they observe.

Glosten and Milgrom derive the market-maker's price setting equations under asymmetric information to be such that the bid quote is the expectation of the true value given that a sell order is received and the ask quote is the expectation of the true value given that a buy order is received. These expectations cannot be computed (except in "toy" instances) without maintaining a probability density estimate over the true value of the stock, especially when the true value itself may change. The major technical contribution of this chapter is the introduction of a nonparametric density estimation technique for maintaining a probability distribution over the true value that the market-maker can use to set prices. I also present a method to approximately solve the price setting equations in a realistic situation with dollars-and-cents quotes and prices. Market-makers using this algorithm in simulations can successfully achieve low spreads without incurring losses. Market-making agents are also often constrained by inventory control considerations brought about by risk aversion (Amihud and Mendelson 1980), so I study the effects of using an inventory control function that is added as an extra module to the market-making algorithm. The inventory control module greatly reduces the variance in market-maker profits.

The simulations yield interesting market properties in different situations. Bid-ask spreads are higher in more volatile markets, market-makers increase the spread in response to uncertainty about the true price, the distribution of returns is leptokurtic, and the autocorrelation of raw returns decays rapidly.

3.2 Related Work

This approach to microstructure problems in dealer markets falls between the traditional theoretical models, such as those of Garman (1976), Glosten and Milgrom (1985) and Kyle (1985) and the agent-based or artificial markets approach adopted by Darley et al. (2000) and Raberto et al. (2001) among others. This extension of the theoretical model of Glosten and Milgrom into a more realistic setting is nevertheless much simpler than most agent-based models. The study of price properties in simulated markets is related to the burgeoning econophysics literature that studies statistical properties of price movements in real markets and their departures from the efficient market hypothesis (Gabaix et al. 2003, Farmer and Lillo 2004, Bouchaud et al. 2004, *inter alia*). Much of the econophysics work attempts to work backward from real stock market data and model price impact functions and thus derive properties of price processes. The approach taken here is complementary – it makes the traditional theoretical economic description richer and derives properties from the theory.

3.3 The Market Model and Market-Making Algorithm

3.3.1 Market Model

The market I analyze is a discrete time dealer market with only one stock. The market-maker sets bid and ask prices (P_b and P_a respectively) at which it is willing to buy or sell one unit of the stock at each time period (when necessary the bid and ask prices at time period i are denoted as P_b^i and P_a^i). If there are multiple market-makers, the market bid and ask prices are the maximum over each dealer's bid price and the minimum over each dealer's ask price. All transactions occur with the market-maker taking one side of the trade and a member of the trading crowd (henceforth a "trader") taking the other side.

The stock has a true underlying value (or fundamental value) V^i at time period i. All market makers are informed of V^0 at the beginning of a simulation, but do not receive any direct information about V after that[1]. At time period i, a single trader is selected from the trading crowd and allowed to place either a (market) buy or (market) sell order for one unit of the stock. There are two types of traders in the market, uninformed traders and informed traders. An uninformed trader will place a buy or sell order for one unit with equal probability, or no order with some probability if selected to trade. An informed trader who is selected to trade knows V^i and will place a buy order if $V^i > P_a^i$, a sell order if $V^i < P_b^i$ and no order if $P_b^i \leq V^i \leq P_a^i$.

In addition to perfectly informed traders, the model also allows for the presence of noisy informed traders. A noisy informed trader receives a signal of the true price $W^i = V^i + \tilde{\eta}(0, \sigma_W)$ where $\tilde{\eta}(0, \sigma_W)$ represents a sample from a normal distribution with mean 0 and variance σ_W^2. The noisy informed trader believes this is the true value of the stock, and places a buy order if $W^i > P_a^i$, a sell order if $W^i < P_b^i$ and no order if $P_b^i \leq W^i \leq P_a^i$.

The true underlying value of the stock evolves according to a jump process. At time $i + 1$, with probability p, a jump in the true value occurs[2]. It is also possible to fix the periodicity of these jumps to model, for example, daily releases of information. When a jump occurs, the value changes according to the equation $V^{i+1} = V^i + \tilde{\omega}(0, \sigma)$ where $\tilde{\omega}(0, \sigma)$ represents a sample from a normal distribution with mean 0 and variance σ^2. Market-makers are informed of when a jump has occurred, but not of the size or direction of the

[1] That is, the only signals a market-maker receives about the true value of the stock are through the buy and sell orders placed by the trading crowd.

[2] p is typically small, of the order of 1 in 1000 in most simulations

jump.

This model of the evolution of the true value corresponds to the notion of the true value evolving as a result of occasional news items. The periods immediately following jumps are the periods in which informed traders can trade most profitably, because the information they have on the true value has not been disseminated to the market yet, and the market maker is not informed of changes in the true value and must estimate these through orders placed by the trading crowd. The market-maker will not update prices to the neighborhood of the new true value for some period of time immediately following a jump in the true value, and informed traders can exploit the information asymmetry.

3.3.2 The Market-Making Algorithm

The market-maker attempts to track the true value over time by maintaining a probability distribution over possible true values and updating the distribution when it receives signals from the orders that traders place. The true value and the market-maker's prices together induce a probability distribution on the orders that arrive in the market. The market-maker must maintain an online probabilistic estimate of the true value.

Glosten and Milgrom (1985) analyze the setting of bid and ask prices so that the market maker enforces a zero profit condition. The zero profit condition corresponds to the Nash equilibrium in a setting with competitive market-makers. Glosten and Milgrom suggest that the market maker should set $P_b = E[V|\text{Sell}]$ and $P_a = E[V|\text{Buy}]$. The market-making algorithm computes these expectations using the probability density function being estimated.

Various layers of complexity can be added on top of the basic algorithm. For example, minimum and maximum conditions can be imposed on the spread, and an inventory control mechanism could form another layer after the zero-profit condition prices are decided. I will describe the density estimation technique in detail before addressing other possible factors that market-makers can take into account in deciding how to set prices. For simplicity of presentation, I neglect noisy informed traders in the initial derivation, and present the updated equations for taking them into account later.

Derivation of Bid and Ask Price Equations

Let α be the proportion of informed traders in the trading crowd, and let η be the probability that an uninformed trader places a buy (or sell) order. Then the probability that an uninformed trader places no order is $1 - 2\eta$.

In order to estimate the expectation of the underlying value, it is necessary to compute the conditional probability that $V = x$ given that a particular type of order is received. Taking market sell orders as an example:

$$E[V|\text{Sell}] = \int_0^\infty x \Pr(V = x|\text{Sell})\, dx$$

To explicitly (approximately) compute these values, discretize the X-axis into intervals, with each interval representing one cent. Then we get:

$$E[V|\text{Sell}] = \sum_{V_i=V_{\min}}^{V_i=V_{\max}} V_i \Pr(V = V_i|\text{Sell})$$

Applying Bayes' rule and simplifying:

$$E[V|\text{Sell}] = \sum_{V_i=V_{\min}}^{V_i=V_{\max}} \frac{V_i \Pr(\text{Sell}|V = V_i) \Pr(V = V_i)}{\Pr(\text{Sell})}$$

The *a priori* probability of a sell order (denoted by P_{Sell}) can be computed by taking advantage of the fact that informed traders will always sell if $V < P_b$ and never sell otherwise, while uninformed traders will sell with a constant probability:

$$P_{\text{Sell}} = \sum_{V_i=V_{\min}}^{V_i=V_{\max}} \Pr(\text{Sell}|V = V_i) \Pr(V = V_i)$$

$$= \sum_{V_i=V_{\min}}^{V_i=P_b-1} [(\alpha + (1-\alpha)\eta) \Pr(V = V_i)] + \sum_{V_i=P_b}^{V_i=V_{\max}} [((1-\alpha)\eta) \Pr(V = V_i)] \tag{3.1}$$

Since P_b is set by the market maker to $E[V|\text{Sell}]$:

$$P_b = \frac{1}{P_{\text{Sell}}} \sum_{V_i=V_{\min}}^{V_i=V_{\max}} V_i \Pr(\text{Sell}|V = V_i) \Pr(V = V_i)$$

Since $V_{\min} < P_b < V_{\max}$,

$$P_b = \frac{1}{P_{\text{Sell}}} \sum_{V_i=V_{\min}}^{V_i=P_b-1} V_i \Pr(\text{Sell}|V = V_i) \Pr(V = V_i) + \\ \frac{1}{P_{\text{Sell}}} \sum_{V_i=P_b}^{V_i=V_{\max}} V_i \Pr(\text{Sell}|V = V_i) \Pr(V = V_i) \tag{3.2}$$

The term $\Pr(\text{Sell}|V = V_i)$ is constant within each sum, because of the influence of informed traders. An uninformed trader is equally likely to sell whatever the market maker's bid price. On the other hand, an informed trader will never sell if $V > P_b$. Therefore, $\Pr(\text{Sell}|V < P_b) = (1-\alpha)\eta + \alpha$ and $\Pr(\text{Sell}|V \geq P_b) = (1-\alpha)\eta$. The above equation reduces to:

$$P_b = \frac{1}{P_{\text{Sell}}} \left(\sum_{V_i=V_{\min}}^{V_i=P_b-1} ((1-\alpha)\eta + \alpha) V_i \Pr(V = V_i) + \sum_{V_i=P_b}^{V_i=V_{\max}} ((1-\alpha)\eta) V_i \Pr(V = V_i) \right) \tag{3.3}$$

Using a precisely parallel argument, we can derive the expression for the market-maker's ask price. First, note that the prior probability of a buy order, P_{Buy} is:

$$P_{\text{Buy}} = \sum_{V_i=V_{\min}}^{V_i=P_a} [((1-\alpha)\eta) \Pr(V = V_i)] + \sum_{V_i=P_a+1}^{V_i=V_{\max}} [(\alpha + (1-\alpha)\eta) \Pr(V = V_i)] \tag{3.4}$$

Then P_a is the solution to the equation:

$$P_a = \frac{1}{P_{\text{Buy}}} \left(\sum_{V_i=V_{\min}}^{V_i=P_a} ((1-\alpha)\eta) V_i \Pr(V = V_i) + \sum_{V_i=P_a+1}^{V_i=V_{\max}} ((1-\alpha)\eta + \alpha) V_i \Pr(V = V_i) \right) \tag{3.5}$$

Accounting for Noisy Informed Traders

An interesting feature of the probabilistic estimate of the true value is that the probability of buying or selling is the same conditional on V being smaller than or greater than a certain amount. For example, $\Pr(\text{Sell}|V = V_i, V_i \leq P_b)$ is a constant, independent of V. If we assume that all informed traders receive noisy signals, with the noise normally distributed with mean 0 and variance σ_W^2, and, as before, α is the proportion of informed traders in the trading crowd, then equation 3.2 still applies. Now the probabilities $\Pr(\text{Sell}|V = V_i)$ are no longer determined solely by whether $V_i < P_b$ or $V_i \geq P_b$. Instead, the new equations are:

$$\Pr(\text{Sell}|V = V_i, V_i \le P_b) = (1-\alpha)\eta + \alpha \Pr(\tilde{\eta}(0, \sigma_W^2) < (P_b - V_i)) \tag{3.6}$$

$$\Pr(\text{Sell}|V = V_i, V_i > P_b) = (1-\alpha)\eta + \alpha \Pr(\tilde{\eta}(0, \sigma_W^2) > (V_i - P_b)) \tag{3.7}$$

The second term in the first equation reflects the probability that an informed trader would sell if the fundamental value were less than the market-maker's bid price. This will occur as long as $W = V + \tilde{\eta}(0, \sigma_W^2) < P_b$. The second term in the second equation reflects the same probability under the assumption that $V \ge P_b$.

We can compute the conditional probabilities for buy orders equivalently:

$$\Pr(\text{Buy}|V = V_i, V_i \le P_a) = (1-\alpha)\eta + \alpha \Pr(\tilde{\eta}(0, \sigma_W^2) > (P_a - V_i)) \tag{3.8}$$

$$\Pr(\text{Buy}|V = V_i, V_i > P_a) = (1-\alpha)\eta + \alpha \Pr(\tilde{\eta}(0, \sigma_W^2) < (V_i - P_a)) \tag{3.9}$$

Now, we have the new buy and sell priors:

$$\begin{aligned} P_{\text{Sell}} = & \sum_{V_i=V_{\min}}^{V_i=P_b-1} \left[\alpha \Pr(\tilde{\eta}(0, \sigma_W^2) < (P_b - V_i)) + (1-\alpha)\eta\right] \Pr(V = V_i) + \\ & \sum_{V_i=P_b}^{V_i=V_{\max}} \left[\alpha \Pr(\tilde{\eta}(0, \sigma_W^2) > (V_i - P_b)) + (1-\alpha)\eta\right] \Pr(V = V_i) \end{aligned} \tag{3.10}$$

$$\begin{aligned} P_{\text{Buy}} = & \sum_{V_i=V_{\min}}^{V_i=P_a} \left[\alpha \Pr(\tilde{\eta}(0, \sigma_W^2) > (P_a - V_i)) + (1-\alpha)\eta\right] \Pr(V = V_i) + \\ & \sum_{V_i=P_a+1}^{V_i=V_{\max}} \left[(\alpha \Pr(\tilde{\eta}(0, \sigma_W^2) < (V_i - P_a) + (1-\alpha)\eta)\right] \Pr(V = V_i) \end{aligned} \tag{3.11}$$

Substituting these conditional probabilities back into the fixed point equations and the density update rule used by the market-maker by combining equations 3.2, 3.6 and 3.7, and using the sell prior from equation 3.10

$$P_b = \frac{1}{P_{\text{Sell}}} \sum_{V_i=V_{\min}}^{V_i=P_b} \left[((1-\alpha)\eta + \alpha \Pr(\tilde{\eta}(0,\sigma_W^2) < (P_b - V_i)))V_i \Pr(V = V_i)\right] +$$
$$\frac{1}{P_{\text{Sell}}} \sum_{V_i=P_b+1}^{V_i=V_{\max}} \left[((1-\alpha)\eta + \alpha \Pr(\tilde{\eta}(0,\sigma_W^2) > (V_i - P_b)))V_i \Pr(V = V_i)\right] \tag{3.12}$$

Similarly, for the ask price, using the buy prior from equation 3.11:

$$P_a = \frac{1}{P_{\text{Buy}}} \sum_{V_i=V_{\min}}^{V_i=P_a} \left[((1-\alpha)\eta + \alpha \Pr(\tilde{\eta}(0,\sigma_W^2) > (P_a - V_i)))V_i \Pr(V = V_i)\right] +$$
$$\frac{1}{P_{\text{Buy}}} \sum_{V_i=P_a+1}^{V_i=V_{\max}} \left[((1-\alpha)\eta + \alpha \Pr(\tilde{\eta}(0,\sigma_W^2) < (V_i - P_a)))V_i \Pr(V = V_i)\right] \tag{3.13}$$

Approximately Solving the Equations

A number of problems arise with the analytical solution of these discrete equations for setting the bid and ask prices. Most importantly, we have not yet specified the probability distribution for priors on V, and any reasonably complex solution leads to a form that makes analytical solution infeasible. Secondly, the values of $V_{\min}$ and $V_{\max}$ are undetermined. And finally, actual solution of these fixed point equations must be approximated in discrete spaces. Each of these problems must be solved to construct an approximate solution to the problem.

The algorithm assumes that the market-making agent is aware of the true value at time 0, and from then onwards the true value infrequently receives random shocks (or jumps) drawn from a normal distribution (the variance of which is known to the agent). The market-maker constructs a vector of prior probabilities on various possible values of V as follows.

If the initial true value is V_0 (when rounded to an integral value in cents), then the agent constructs a vector going from $V_0 - 4\sigma$ to $V_0 + 4\sigma - 1$ to contain the prior value probabilities. The probability that $V = V_0 - 4\sigma + i$ is given by the ith value in this vector[3]. The vector is initialized by setting the ith value in the vector to $\int_{-4\sigma+i}^{-4\sigma+i+1} \mathcal{N}(0,\sigma)\,dx$ where $\mathcal{N}$ is the normal density function in x with specified mean and variance. The vector is maintained

[3] The true value can be a real number, but for all practical purposes it ends up getting truncated to the floor of that number.

in a normalized state at all times so that the entire probability mass for V lies within it.

The fixed point equations 3.12 and 3.13 are approximately solved by using the result from Glosten and Milgrom that $P_b \leq E[V] \leq P_a$ and then, to find the bid price, for example, cycling from E[V] downwards until the difference between the left and right hand sides of the equation stops decreasing. The fixed point real-valued solution must then be closest to the integral value at which the distance between the two sides of the equation is minimized.

Updating the Density Estimate

The market-maker receives probabilistic signals about the true value. With perfectly informed traders, each signal says that with a certain probability, the true value is lower (higher) than the bid (ask) price. With noisy informed traders, the signal differentiates between different possible true values depending on the market-maker's bid and ask quotes. Each time that the market-maker receives a signal about the true value by receiving a market buy or sell order, it updates the posterior on the value of V by scaling the distributions based on the type of order. The Bayesian updates are easily derived. For example, for $V_i \leq P_a$ and market buy orders:

$$\Pr(V = V_i|\text{Buy}) = \frac{\Pr(\text{Buy}|V = V_i)\Pr(V = V_i)}{\Pr(\text{Buy})}$$

The prior probability $V = V_i$ is known from the density estimate, the prior probability of a buy order is known from equation 3.11, and $\Pr(\text{Buy}|V = V_i, V_i \leq P_a)$ can be computed from equation 3.8. We can compute the posterior similarly for each of the cases. One case that is instructive to look at since it is not derived above is the case when no order is received.

$$\Pr(V = V_i|\text{No order}) = \frac{\Pr(\text{No order}|V = V_i)\Pr(V = V_i)}{\Pr(\text{No order})}$$

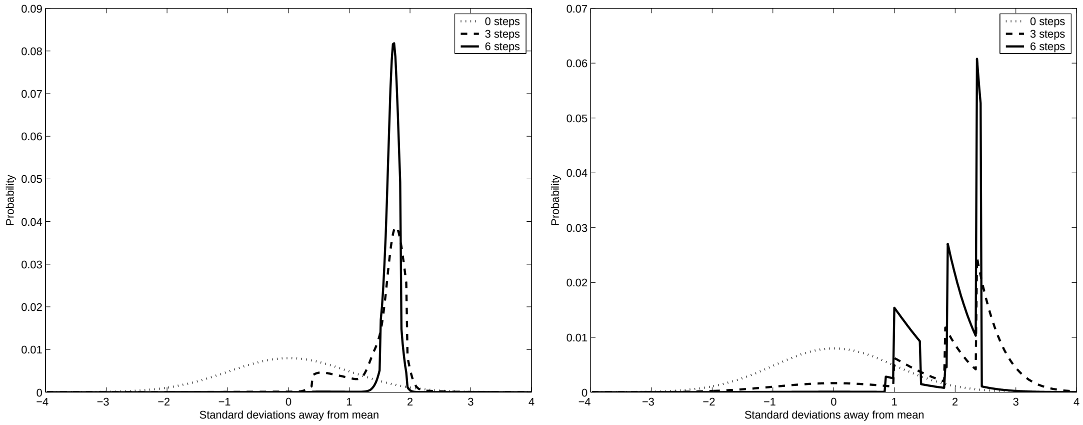


Figure 3-1: The evolution of the market-maker's probability density estimate with noisy informed traders (left) and perfectly informed traders (right)

Now,

$$\begin{aligned}
\Pr(\text{No order}|V = V_i, V_i < P_b) &= (1-\alpha)(1-2\eta) + \alpha \Pr(\tilde{\eta}(0,\sigma_W^2) > (P_b - V_i)) \\
\Pr(\text{No order}|V = V_i, P_b \leq V_i \leq P_a) &= (1-\alpha)(1-2\eta) + \alpha[\Pr(P_b - V_i < \tilde{\eta}(0,\sigma_W^2)) + \\
&\qquad \Pr(V_i - P_a < \tilde{\eta}(0,\sigma_W^2))] \\
\Pr(\text{No order}|V = V_i, V_i > P_a) &= (1-\alpha)(1-2\eta) + \alpha \Pr(V_i - P_a < \tilde{\eta}(0,\sigma_W^2))
\end{aligned}$$

which allows us to compute the prior as well as all the terms in the numerator.

In the case of perfectly informed traders, the signal only specifies that the true value is higher or lower than some price, and not how much higher or lower. In that case, the update equations are as follows. If a market buy order is received, this is a signal that with probability $(1-\alpha)\eta + \alpha, V > P_a$. Similarly, if a market sell order is received, the signal indicates that with probability $(1-\alpha)\eta + \alpha, V < P_b$.

In the former case, all probabilities for $V = V_i, V_i > P_a$ are multiplied by $(1-\alpha)\eta + \alpha$, while all the other discrete probabilities are multiplied by $1 - \alpha - (1-\alpha)\eta$. Similarly, when a sell order is received, all probabilities for $V = V_i, V_i < P_b$ are multiplied by $(1-\alpha)\eta + \alpha$, and all the remaining discrete probabilities are multiplied by $1 - \alpha - (1-\alpha)\eta$ before renormalizing.

These updates lead to less smooth density estimates than the updates for noisy informed traders, as can be seen from figure 3-1, which shows the density functions 0, 3 and 6 steps after a jump in the underlying value of the stock. The update equations that consider noisy informed traders smoothly transform the probability distribution around the last transaction price by a mixture of a Gaussian and a uniform density, whereas the update equations for perfectly informed traders discretely shift all probabilities to one side of the transaction price in one direction and on the other side of the transaction price in the other direction. The estimates for perfectly informed traders are more susceptible to noise, as they do not restrict most of the mass of the probability density function to as small an area as the estimates for noisy informed traders.

3.4 Experimental Evaluation

3.4.1 Experimental Framework

Unless specified otherwise, it can be assumed that all simulations take place in a market populated by noisy informed traders and uninformed traders. The noisy informed traders receive a noisy signal of the true value of the stock with the noise term being drawn from a Gaussian distribution with mean 0 and standard deviation 5 cents. The standard deviation of the jump process for the stock is 50 cents, and the probability of a jump occurring at any time step is 0.001. The probability of an uninformed buy or sell order is 0.5. The market-maker is informed of when a jump occurs, but not of the size or direction of the jump. The market-maker may use an inventory control function (defined below) and can increase the spread by lowering the bid price and raising the ask price by a fixed amount (this is done to ensure profitability and is also explained below). I report average results from 200 simulations, each lasting 50,000 time steps.

3.4.2 Prices Near a Jump

Figure 3-2 shows that the market-maker successfully tracks the true value over the course of an entire simulation. These results are from a simulation with half the traders being perfectly informed and the other half uninformed. The bid-ask spread reflects the market-maker's uncertainty about the true value — for example, it is much higher immediately after the true value has jumped.

Figure 3-2 also demonstrates that the asymmetry of information immediately following a price jump gets resolved very quickly. To investigate this further, we can track the

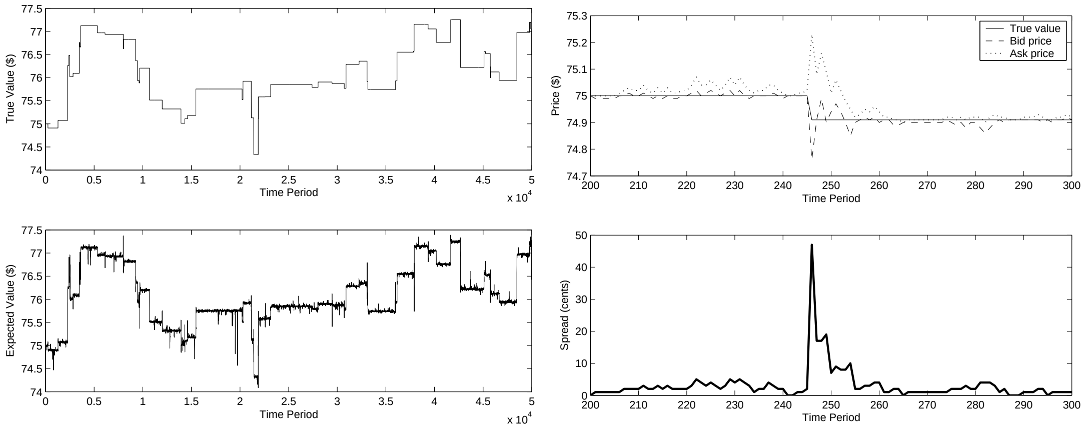


Figure 3-2: The market-maker's tracking of the true price over the course of the simulation (left) and immediately before and after a price jump (right)

average spread immediately following a price jump in a similar market environment (except with noisy informed traders instead of perfectly informed ones). The results of this experiment are shown in figure 3-3. It is clear that the informational asymmetry gets resolved very quickly (within thirty trades) independently of the standard deviation of the jump process.

3.4.3 Profit Motive

The zero-profit condition of Glosten and Milgrom is expected from game theoretic considerations when multiple competitive dealers are making markets in the same stock. However, since this method is an approximation scheme, the zero profit method is unlikely to truly be zero-profit. Further, the market-maker is not always in a perfectly competitive scenario where it needs to restrict the spread as much as possible.

The simplest solution to the problem of making profit is to increase the spread by pushing the bid and ask prices apart after the zero-profit bid and ask prices have been computed using the density estimate obtained by the market-making algorithm. Figure 3-4 shows the profit obtained by a single monopolistic market-maker in markets with different percentages of informed traders. The numbers on the X axis show the amount (in cents) that is subtracted from (added to) the zero-profit bid (ask) price in order to push the spread apart (I will refer to this number as the shift factor).

With lower spreads, most of the market-maker's profits come from the noise factor of

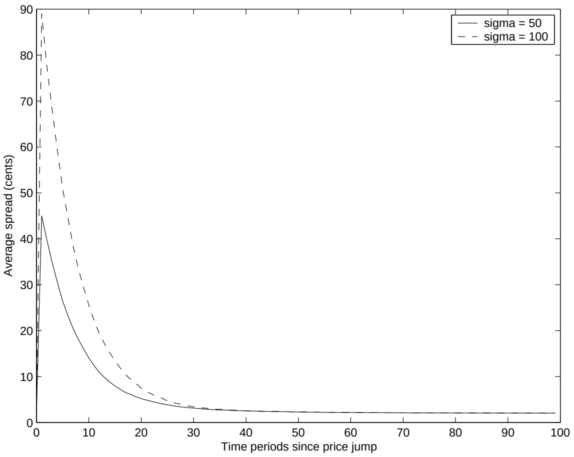


Figure 3-3: Average spread following a price jump for two different values of the standard deviation of the jump process

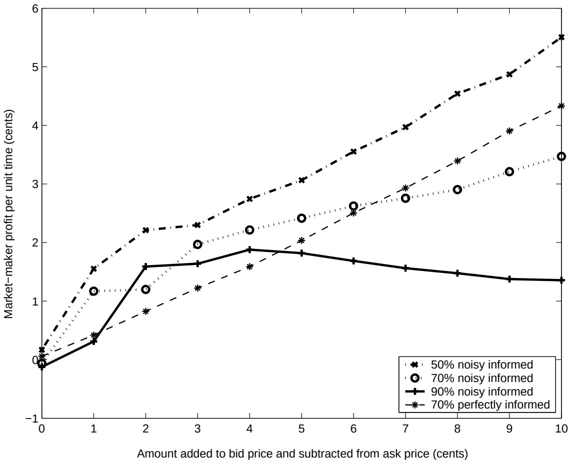


Figure 3-4: Market-maker profits as a function of increasing the spread

the informed traders, whereas with a higher spread, most of the market-maker's profits come from the trades of uninformed traders. Different percentages of informed traders lead to differently shaped profit curves. For example, there is a sharper jump in the transition from a shift factor of 0 to a shift factor of 1 with fewer noisy informed traders (50% or 70%) whereas with 90% noisy informed traders the sharper jump is in going from a shift factor of 1 to a shift factor of 2. With only 50% of the traders being informed, the market-maker's profit keeps increasing with the size of the spread.

However, increasing the spread beyond a point is counterproductive if there are enough noisy informed traders in the markets, because then the market-maker's prices are far enough away from the true value that even the noise factor cannot influence the informed traders to make trades. With 90% of the traders being informed, a global maximum (at least for reasonable spreads) is attained with a low spread. This is where the tradeoff between not increasing the spread too much in order to profit from the noise in the informed traders' signals and increasing the spread more to profit more from uninformed traders is optimized. On the opposite end of the spectrum, the market-maker's profits increase smoothly and uniformly with the spread when there are only perfectly informed traders in the market in addition to the uninformed traders, since all the market-maker's profits are from the uninformed traders.

It is important to note that market-makers can make reasonable profits with low average spreads. For a market with 70% of the trading crowd consisting of noisy informed traders and the remaining 30% consisting of uninformed traders, our algorithm, using a shift factor of 1, achieves an average profit of 1.17 cents per time period with an average spread of 2.28 cents. Using a shift factor of 0, the average profit is -0.06 cents with an average spread of 0.35 cents. These parameter settings in this environment yield a market-maker that is close to a Nash equilibrium player, and it is unlikely that any algorithm would be able to outperform this one in direct competition in such an environment given the low spread. An interesting avenue to explore is the possibility of adaptively changing the shift factor depending on the level of competition in the market. Clearly, in a monopolistic setting, a market-maker is better off using a high shift factor, whereas in a competitive setting it is likely to be more successful using a smaller one. An algorithm for changing the shift factor based on the history of other market-makers' quotes would be useful.

3.4.4 Inventory Control

Stoll (1978) analyzes dealer costs in conducting transactions and divides them into three categories. These three categories are portfolio risk, transaction costs and the cost of asymmetric information. In the model presented so far, following Glosten and Milgrom (1985), I have assumed that transactions have zero execution cost and developed a pricing mechanism that explicitly attempts to set the spread to account for the cost of asymmetric information. A realistic model for market-making necessitates taking portfolio risk into account and controlling inventory in setting bid and ask prices. In the absence of consideration of trade size and failure conditions, portfolio risk should affect the placement of the bid and ask prices, but not the size of the spread,[4] unless the market-maker is passing the costs of holding inventory along to the traders through the spread (Amihud and Mendelson 1980, Stoll 1978, Grossman and Miller 1988). If the market-maker has a long position in the stock, minimizing portfolio risk is achieved by lowering both bid and ask prices, and if the market-maker has a short position, inventory is controlled by raising both bid and ask prices.

Inventory control can be incorporated into the architecture of the market-making algorithm by using it as an adjustment parameter applied after bid and ask prices have been determined by equations 3.12 and 3.13. A simple inventory control technique investigated here is to raise or lower the bid and ask prices by a linear function of the inventory holdings of the market-maker. The amount added to the bid and ask prices is $-\gamma I$ where I is the amount of inventory held by the market-maker (negative for short positions) and γ is a risk-aversion coefficient.

Table 3.1 shows statistics indicating the effectiveness of the inventory control module for minimizing market-maker risk and the effects of using different values of γ. The figures use the absolute value of the difference between last true value and initial true value as a proxy for market volatility. 200 simulations were run for each experiment, and 70% of the traders were noisy informed traders, while the rest were uninformed. The market-maker used a shift factor of 1 for increasing / decreasing the ask / bid prices respectively.

γ	0	0.1	1
Average (absolute) inventory holdings	1387.2	9.74	1.66
Profit (cents per period)	1.169	0.757	0.434
Standard Deviation of profit	9.3813	0.0742	0.0178

Table 3.1: Average absolute value of MM inventory at the end of a simulation, average profit achieved and standard deviation of per-simulation profit for market-makers with different levels of inventory control

Shift	σ	p	Spread	Profit
0	50	.001	0.3479	-0.0701
0	50	.005	1.6189	-0.1295
0	100	.001	0.6422	-0.0694
0	100	.005	3.0657	-0.2412
1	50	.001	2.3111	0.7738
1	50	.005	3.5503	0.6373
1	100	.001	2.6142	0.7629
1	100	.005	4.9979	0.6340

Table 3.2: Market-maker average spreads (in cents) and profits (in cents per time period) as a function of the shift (amount added to ask price and subtracted from bid price), standard deviation of the jump process (σ) and the probability of a jump occurring at any point in time (p)

3.4.5 The Effects of Volatility

Volatility of the underlying true value process is affected by two parameters. One is the standard deviation of the jump process, which affects the variability in the amount of each jump. The other is the probability with which a jump occurs. Table 3.2 shows the result of changing the standard deviation σ of the jump process and the probability p of a jump occurring at any point in time. As expected, the spread increases with increased volatility, both in terms of σ and p. The precise dependence of the expected profit and the average spread on the values of σ and p is interesting. For example, increasing p for $\sigma = 100$ has a more significant percentage impact on the spread than the same increase when $\sigma = 50$. This is probably because the mean reflects the relative importance of the symmetric and asymmetric information regimes, which is affected by p.

[4] One would expect spread to increase with the trade size. The size of the spread is, of course, affected by the adverse selection arising due to the presence of informed traders.

3.4.6 Accounting for Jumps

The great advantage of this algorithm for density estimation and price setting is that it quickly restricts most of the probability mass to a relatively small region of values/prices, which allows the market-maker to quote a small spread and still break even or make profit. The other side of this equation is that once the probability mass is concentrated in one area, the probability density function on other points in the space becomes small. In some cases, it is not possible to seamlessly update the estimate through the same process if a price jump occurs. Another problem is that a sequence of jumps could lead to the value leaving the $[-4\sigma, 4\sigma]$ window used by the density estimation technique.

The discussion above assumes that the market-maker is explicitly informed of when a price jump has occurred, although not of the size or direction of the jump. The problem can be solved by recentering the distribution around the current expected value and reinitializing in the same way in which the prior distribution on the value is initially set up. The "unknown jump" case is more complicated. An interesting avenue for future work, especially if trade sizes are incorporated into the model, is to devise a formal mathematical method for deciding when to recenter the distribution. An example of such a method would be to learn a classifier that is good at predicting when a price jump has occurred. Perhaps there are particular types of trades that commonly occur following price jumps, especially when limit orders and differing trade sizes are permitted. Sequences of such trades may form patterns that predict the occurrence of jumps in the underlying value.

An example of a very simple rule that demonstrates the feasibility of such an idea is to recenter based on some notion of order imbalance. Such a rule could recenter when there have been k more buy orders than sell orders (or vice versa) in the last n time steps. Table 3.3 shows the results obtained using $n = 10$ and $k = 5$ values with the market-maker increasing (decreasing) the ask (bid) price by 1 cent beyond the zero profit case, and using linear inventory control with $\gamma = 0.1$. The loss of the expectation is defined as the average of the absolute value of the difference between the true value and the market-maker's expectation of the true value at each point in time. While this rule makes a loss, the spread is reasonable and the expectation is not too far away on average from the true value. This demonstrates that it is reasonable to assume that there are endogenous ways for market-makers to know when jumps have occurred.

Case	Profit	Loss of expectation	Average spread
Known	0.7693	0.7546	2.3263
Unknown	-0.6633	4.5616	4.3708

Table 3.3: Average profit (in cents per time period), loss of expectation and average spread (cents) with jumps known and unknown

3.5 Time Series and Distributional Properties of Returns

We can utilize the market and price-setting models developed so far in order to derive price properties in the simulated market and compare these properties to what is seen in real markets by analyzing return data for ten stocks from the TAQ database. Obviously the simplicity of the model simplicity means that it will not capture many of the features of real data. This discussion is intended to highlight where the model agrees and disagrees with real data and to hypothesize why these differences occur and whether they can be accommodated by additions to the model.

The discussion in this section uses standardized log returns, with fifty discrete time periods as the length of time for simulation returns and five minutes for stock returns. The simulation data is averaged over 100 runs of 50,000 time steps each with 70% informed traders and the market-maker using inventory control with $\gamma = 1$ and increasing the ask price and decreasing the bid price by one cent beyond the zero-profit computation in order to ensure profitability. The probability of a jump in the true value at any time, $p = 0.005$ and the standard deviation of the jump process $\sigma = 50$ (cents). The stock data from the TAQ database is for ten randomly selected component stocks of the S&P 500 index for the year 2002.[5]

Liu et al. (1999) present a detailed analysis of the time series properties of returns in a real equity market (they focus on the S&P 500 and component stocks). Their major findings are that return distributions are leptokurtic and fat-tailed, with power-law decay in the tails, volatility clustering occurs and the autocorrelation of absolute values of returns is persistent over large time scales (again with power-law decay), as opposed to the autocorrelation of raw returns, which disappears rapidly[6]. The recent econophysics literature

[5] The symbols for the ten stocks are CA, UNP, AMAT, GENZ, GLK, TNB, PMTC RX, UIS and VC. Of these the first four are considered large cap (with market capitalizations in excess of 6 billion dollars) and the other six are small cap.

[6] Liu et al. (1999) are not the first to discover these properties of financial time series. However, they summarize much of the work in an appropriate fashion and provide detailed references, and they present novel

has seen a growing debate about the origin of power laws in such data, for example, the theory of Gabaix et al. Gabaix et al. (2003), Plerou et al. (2004) and the alternative analysis of Farmer and Lillo (2004).

Simulation results show rapid decay of autocorrelation of raw returns (the coefficient is already at noise levels at lag 1). Bouchaud et al. (2004) discuss how prices are a random walk because of a critical balance between liquidity takers who place market orders and create temporal correlations in the sign of trades because they do not wish to place huge orders that move the market immediately, and liquidity providers who attempt to mean-revert the price. In this model, all the traders except the market maker are liquidity takers, and they have an even harsher restriction on the trade size they can place. Explicitly modeling the price-setting process of the liquidity provider shows that the autocorrelation of raw returns decays rapidly and arbitrage opportunities do not arise.

Looking at the real data, there is a negative serial correlation of raw returns at one lag for the small cap stocks (Figure 3-5). This may be because of less trading in these stocks. We do not see this spike in the model presented here if we look at price changes over fifty periods, but if we look at them over fewer discrete time periods (say one or two) instead of fifty, we can see a statistically significant negative autocorrelation at one lag as well.

In terms of absolute returns, the real data shows a pronounced daily trend, with the autocorrelation coefficient spiking at the lag corresponding to one day (figure 3-6). This one day periodicity probably corresponds to opening and closing procedures (which also cause the spread to widen). Part of this phenomenon can be replicated in this model by fixing the periodicity of shocks to the true value. This part corresponds to our intuition of major information shocks coming at the beginning of trading days, and induces a concept of the beginning of the trading day among agents in the market. However, it is hard to model the fact that the autocorrelation coefficient is higher for lags corresponding to, say three-quarters of a day. This is because the agents do not have a notion of the market closing, which may be what drives up the coefficient for these lags in real data. Perhaps a model in which the market-maker is sure that a price jump has occurred at the beginning of trading days, but also assumes the possibility of unknown jumps later in the day could explain these facts.

Simulations using the extended model presented here yield return distributions with

results on the power law distribution of volatility correlation.

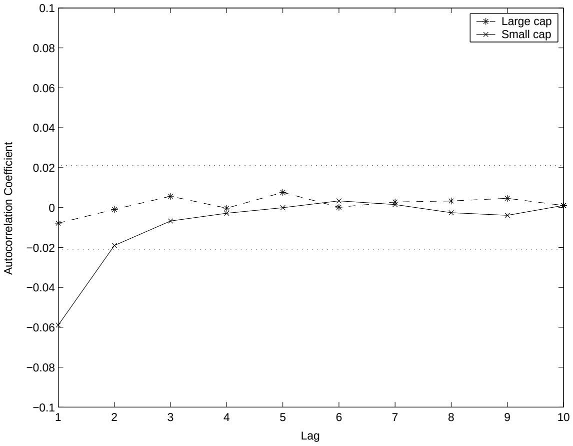


Figure 3-5: Autocorrelation of raw returns for small and large cap stocks

Note: The dotted lines represent noise levels computed as $\pm 3/\sqrt{N}$ where N is the length of the time series.

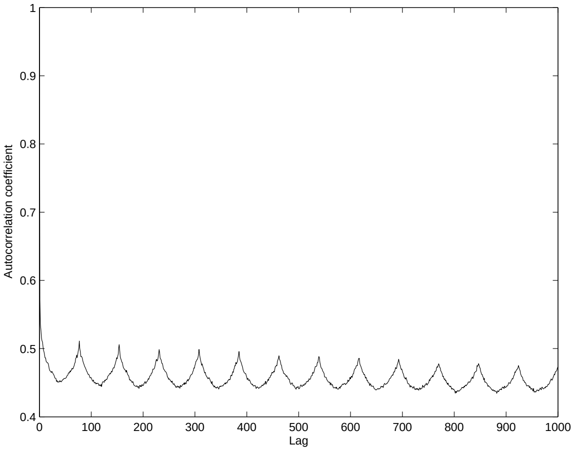


Figure 3-6: Autocorrelation of absolute returns for stock data

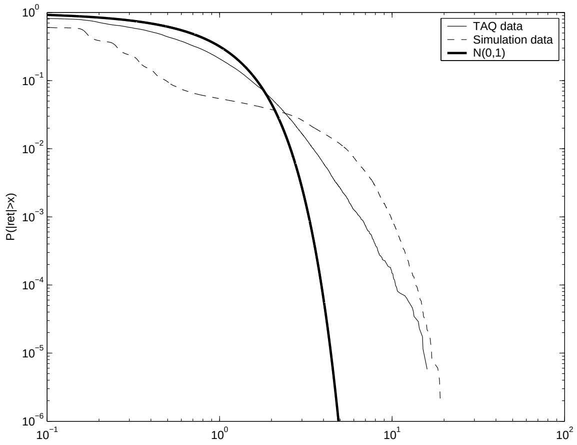


Figure 3-7: Distribution of absolute returns for simulation data and stock data, along with the cumulative distribution of the absolute value of a random variable drawn from a standard normal distribution

some similarities to stock market data, as can be seen from Figure 3-7. The distribution of returns is leptokurtic, although it does not decay with a power law tail, suggesting that the model needs further extensions before contributing to the debate on the origin of power law tails. A huge proportion of the returns are very small, and virtually all of these occur in the symmetric information regime, and there are very few large returns, most of which occur in the asymmetric information regime immediately following a price jump.

The sample kurtosis for the simulation return data is 49.49 (by way of comparison, the sample kurtosis for the large cap stocks is 19.49 and that for the small cap stocks is 13.18). The exact shape of the distribution is affected by parameters like artificial inflation of the spread and inventory control. If the market-maker were to dynamically change the spread during the course of a simulation based on factors like competition or the need to maintain market quality, perhaps that would yield power law tails.

3.6 Discussion

This chapter extends the Glosten-Milgrom model of dealer markets by describing an algorithm for maintaining a probability density estimate of the true value of a stock in a dynamic market with regular shocks to the value and using this estimate to explicitly set prices in a somewhat realistic framework. The new model explicitly incorporates noise into the specification of informed trading, allowing for a rich range of market behavior. A careful empirical evaluation of characteristics of the market-making algorithm in simulation yields helpful insights for the problem of designing a market-making agent. Further, this framework allows the development of an agent-based model of a dealer market for studying time series, distributional and other properties of prices, and interesting interactions between different parameters.

There are two regimes in the simulated markets. Immediately following a price jump, information is very heterogeneous, spreads are high, and the market is volatile. This informational asymmetry gets resolved rapidly, and the market settles into a regime of homogeneous information with small spreads and low volatility. Analyzing time series and distributional properties of returns in the model shows some similarities and some differences from real data. The differences, in particular, could serve as a starting point for further extensions of this model to computationally study the effects of information and explicit modeling of the true value process in price formation.

Chapter 4

Learning to Trade With "Insider" Information

4.1 Introduction

In financial markets, information is revealed by trading. Once private information is fully disseminated to the public, prices reflect all available information and reach market equilibrium. Before prices reach equilibrium, agents with superior information have opportunities to gain profits by trading. This chapter focuses on the design of a general algorithm that allows an agent to learn how to exploit superior or "insider" information.[1] Suppose a trading agent receives a signal of what price a stock will trade at n trading periods from now. What is the best way to exploit this information in terms of placing trades in each of the intermediate periods? The agent has to make a tradeoff between the profit made from an immediate trade and the amount of information that trade reveals to the market. If the stock is undervalued it makes sense to buy some stock, but buying too much may reveal the insider's information too early and drive the price up, relatively disadvantaging the insider.

This problem has been studied extensively in the finance literature, initially in the context of a trader with monopolistic insider information (Kyle 1985), and later in the context of competing insiders with homogeneous (Holden and Subrahmanyam 1992) and hetero-

[1] The term "insider" information has negative connotations in popular belief. I use the term solely to refer to superior information, however it may be obtained (for example, paying for an analyst's report on a firm can be viewed as a way of obtaining insider information about a stock).

geneous (Foster and Viswanathan 1996) information.[2] All these models derive equilibria under the assumption that traders are perfectly informed about the structure and parameters of the world in which they trade. For example, in Kyle's model, the informed trader knows two important distributions — the ex ante distribution of the liquidation value and the distribution of other ("noise") trades that occur in each period.

In this chapter, I start from Kyle's (1985) original model, in which the trading process is structured as a sequential auction at the end of which the stock is liquidated. An informed trader or "insider" is told the liquidation value some number of periods before the liquidation date, and must decide how to allocate trades in each of the intervening periods. There is also some amount of uninformed trading (modeled as white noise) at each period. The clearing price at each auction is set by a market-maker who sees only the combined order flow (from both the insider and the noise traders) and seeks to set a zero-profit price. In the next section I discuss the importance of this problem from the perspectives of research both in finance and in reinforcement learning. In sections 4.3 and 4.4 I introduce the market model and two learning algorithms, and in Section 4.5 I present experimental results. Finally, Section 4.6 concludes and discusses future research directions.

4.2 Motivation: Bounded Rationality and Reinforcement Learning

One of the arguments for the standard economic model of a decision-making agent as an unboundedly rational optimizer is the argument from learning. In a survey of the bounded rationality literature, John Conlisk (1996) lists this as the second among eight arguments typically used to make the case for unbounded rationality. To paraphrase his description of the argument, it is all right to assume unbounded rationality because agents learn optima through practice. Commenting on this argument, Conlisk says "learning is promoted by favorable conditions such as rewards, repeated opportunities for practice, small deliberation cost at each repetition, good feedback, unchanging circumstances, and a simple context." The learning process must be analyzed in terms of these issues to see if it will indeed lead to agent behavior that is optimal and to see how differences in the environ-

[2] My discussion of finance models in this chapter draws directly from these original papers and from the survey by O'Hara (1995).

ment can affect the learning process. The design of a successful learning algorithm for agents who are not necessarily aware of who else has inside information or what the price formation process is could elucidate the conditions that are necessary for agents to arrive at equilibrium, and could potentially lead to characterizations of alternative equilibria in these models.

One way of approaching the problem of learning how to trade in the framework developed here is to apply a standard reinforcement learning algorithm with function approximation. Fundamentally, the problem posed here has infinite (continuous) state and action spaces (prices and quantities are treated as real numbers), which pose hard challenges for reinforcement learning algorithms. However, reinforcement learning has worked in various complex domains, perhaps most famously in backgammon Tesauro (1995) (see Sutton and Barto (1998) for a summary of some of the work on value function approximation). There are two key differences between these successes and the problem studied here that make it difficult for the standard methodology to be successful without properly tailoring the learning algorithm to incorporate important domain knowledge.

First, successful applications of reinforcement learning with continuous state and action spaces usually require the presence of an offline simulator that can give the algorithm access to many examples in a costless manner. The environment envisioned here is intrinsically online — the agent interacts with the environment by making potentially costly trading decisions which actually affect the payoff it receives. In addition to this, the agent wants to minimize exploration cost because it is an active participant in the economic environment. Achieving a high utility from early on in the learning process is important to agents in such environments. Second, the sequential nature of the auctions complicates the learning problem. If we were to try and model the process in terms of a Markov decision problem (MDP), each state would have to be characterized not just by traditional state variables (in this case, for example, last traded price and liquidation value of a stock) but by how many auctions in total there are, and which of these auctions is the current one. The optimal behavior of a trader at the fourth auction out of five is different from the optimal behavior at the second auction out of ten, or even the ninth auction out of ten. While including the current auction and total number of auctions as part of the state would allow us to represent the problem as an MDP, it would not be particularly helpful because the generalization ability from one state to another would be poor. This problem might be mitigated in circumstances where the optimal behavior does not change much from auction to auction, and characterizing these circumstances is important. In fact, I describe an algorithm below that uses a representation where the current auction and the total number of auctions do not factor into the decision. This approach is very similar to model based reinforcement learning with value function approximation, but the main reason why it works very well in this case is that we understand the form of the optimal strategy, so the representations of the value function, state space, and transition model can be tailored so that the algorithm performs close to optimally. I discuss this in more detail in Section 4.5.

An alternative approach to the standard reinforcement learning methodology is to use explicit knowledge of the domain and learn separate functions for each auction. The learning process receives feedback in terms of actual profits received for each auction from the current one onwards, so this is a form of direct utility estimation (Widrow and Hoff 1960). While this approach is related to the direct-reinforcement learning method of Moody and Saffell (2001), the problem studied here involves more consideration of delayed rewards, so it is necessary to learn something equivalent to a value function in order to optimize the total reward.

The important domain facts that help in the development of a learning algorithm are based on Kyle's results. Kyle proves that in equilibrium, the expected future profits from auction i onwards are a linear function of the square difference between the liquidation value and the last traded price (the actual linear function is different for each i). He also proves that the next traded price is a linear function of the amount traded. These two results are the key to the learning algorithm. I will show in later sections that the algorithm can learn from a small amount of randomized training data and then select the optimal actions according to the trader's beliefs at every time period. With a small number of auctions, the learning rule enables the trader to converge to the optimal strategy. With a larger number of auctions the number of episodes required to reach the optimal strategy becomes impractical and an approximate mechanism achieves better results. In all cases the trader continues to receive a high flow utility from early episodes onwards.

4.3 Market Model

The model is based on Kyle's (1985) original model. There is a single security which is traded in N sequential auctions. The liquidation value v of the security is realized after the Nth auction, and all holdings are liquidated at that time. v is drawn from a Gaussian distribution with mean p_0 and variance Σ_0, which are common knowledge. Here we assume that the N auctions are identical and distributed evenly in time. An informed trader or insider observes v in advance and chooses an amount to trade Δx_i at each auction $i \in \{1, \ldots, N\}$. There is also an uninformed order flow amount Δu_i at each period, sampled from a Gaussian distribution with mean 0 and variance $\sigma_u^2 \Delta t_i$ where $\Delta t_i = 1/N$ for our purposes (more generally, it represents the time interval between two auctions).[3] The trading process is mediated by a market-maker who absorbs the order flow while earning zero expected profits. The market-maker only sees the combined order flow $\Delta x_i + \Delta u_i$ at each auction and sets the clearing price p_i. The zero expected profit condition can be expected to arise from competition between market-makers.

Equilibrium in the monopolistic insider case is defined by a profit maximization condition on the insider which says that the insider optimizes overall profit given available information, and a market efficiency condition on the (zero-profit) market-maker saying that the market-maker sets the price at each auction to the expected liquidation value of the stock given the combined order flow.

Formally, let π_i denote the profits made by the insider on positions acquired from the ith auction onwards. Then $\pi_i = \sum_{k=i}^{N} (v - p_k) \Delta x_k$. Suppose that X is the insider's trading strategy and is a function of all information available to her, and P is the market-maker's pricing rule and is again a function of available information. X_i is a mapping from $(p_1, p_2, \ldots, p_{i-1}, v)$ to x_i where x_i represents the insider's total holdings after auction i (from which Δx_i can be calculated). P_i is a mapping from $(x_1 + u_1, \ldots, x_i + u_i)$ to p_i. X and P consist of all the component X_i and P_i. Kyle defines the sequential auction equilibrium as a pair X and P such that the following two conditions hold:

[3] The motivation for this formulation is to allow the representative uninformed trader's holdings over time to be a Brownian motion with instantaneous variance σ_u^2. The amount traded represents the change in holdings over the interval.

1. *Profit maximization*: For all $i = 1, \dots, N$ and all X':

$$E[\pi_i(X, P)|p_1, \dots, p_{i-1}, v] \geq E[\pi_i(X', P)|p_1, \dots, p_{i-1}, v]$$

2. *Market efficiency*: For all $i = 1, \dots, N$, $p_i = E[v|x_1 + u_1, \dots, x_i + u_i]$

The first condition ensures that the insider's strategy is optimal, while the second ensures that the market-maker plays the competitive equilibrium (zero-profit) strategy. Kyle (1985) also shows that there is a unique linear equilibrium.

Theorem 1 (Kyle, 1985). *There exists a unique linear (recursive) equilibrium in which there are constants $\beta_n, \lambda_n, \alpha_n, \delta_n, \Sigma_n$ such that for:*

$$\begin{aligned} \Delta x_n &= \beta_n(v - p_{n-1})\Delta t_n \\ \Delta p_n &= \lambda_n(\Delta x_n + \Delta u_n) \\ \Sigma_n &= \mathit{var}(v|\Delta x_1 + \Delta u_1, \dots, \Delta x_n + \Delta u_n) \\ E[\pi_n|p_1, \dots, p_{n-1}, v] &= \alpha_{n-1}(v - p_{n-1})^2 + \delta_{n-1} \end{aligned}$$

Given Σ_0 the constants $\beta_n, \lambda_n, \alpha_n, \delta_n, \Sigma_n$ are the unique solution to the difference equation system:

$$\begin{aligned} \alpha_{n-1} &= \frac{1}{4\lambda_n(1 - \alpha_n\lambda_n)} \\ \delta_{n-1} &= \delta_n + \alpha_n\lambda_n^2\sigma_u^2\Delta t_n \\ \beta_n\Delta t_n &= \frac{1 - 2\alpha_n\lambda_n}{2\lambda_n(1 - \alpha_n\lambda_n)} \\ \lambda_n &= \beta_n\Sigma_n/\sigma_u^2 \\ \Sigma_n &= (1 - \beta_n\lambda_n\Delta t_n)\Sigma_{n-1} \end{aligned}$$

subject to $\alpha_N = \delta_N = 0$ and the second order condition $\lambda_n(1 - \alpha_n\lambda_n) = 0$.[4]

The two facts about the linear equilibrium that will be especially important for learning

[4] The second order condition rules out a situation in which the insider can make unbounded profits by first destabilizing prices with unprofitable trades.

are that there exist constants $\lambda_i, \alpha_i, \delta_i$ such that:

$$\Delta p_i = \lambda_i(\Delta x_i + \Delta u_i) \tag{4.1}$$

$$E[\pi_i | p_1, \ldots, p_{i-1}, v] = \alpha_{i-1}(v - p_{i-1})^2 + \delta_{i-1} \tag{4.2}$$

Perhaps the most important result of Kyle's characterization of equilibrium is that the insider's information is incorporated into prices gradually, and the optimal action for the informed trader is not to trade particularly aggressively at earlier dates, but instead to hold on to some of the information. In the limit as $N \to \infty$ the rate of revelation of information actually becomes constant. Also note that the market-maker imputes a strategy to the informed trader without actually observing her behavior, only the order flow.

4.4 A Learning Model

4.4.1 The Learning Problem

I am interested in examining a scenario in which the informed trader knows very little about the structure of the world, but must learn how to trade using the superior information she possesses. I assume that the price-setting market-maker follows the strategy defined by the Kyle equilibrium. This is justifiable because the market-maker (as a specialist in the New York Stock Exchange sense (Schwartz 1991)) is typically in an institutionally privileged situation with respect to the market and has also observed the order-flow over a long period of time. It is reasonable to conclude that the market-maker will have developed a good domain theory over time.

The problem faced by the insider is similar to the standard reinforcement learning model (Kaelbling et al. 1996, Bertsekas and Tsitsiklis 1996, Sutton and Barto 1998) in which an agent does not have complete domain knowledge, but is instead placed in an environment in which it must interact by taking actions in order to gain reinforcement. In this model the actions an agent takes are the trades it places, and the reinforcement corresponds to the profits it receives. The informed trader makes no assumptions about the market-maker's pricing function or the distribution of noise trading, but instead tries to maximize profit over the course of each sequential auction while also learning the appropriate functions.

4.4.2 A Learning Algorithm

At each auction i the goal of the insider is to maximize

$$\pi_i = \Delta x_i(v - p_i) + \pi_{i+1} \tag{4.3}$$

The insider must learn both p_i and π_{i+1} as functions of the available information. We know that in equilibrium p_i is a linear function of p_{i-1} and Δx_i, while π_{i+1} is a linear function of $(v - p_i)^2$. This suggests that an insider could learn a good representation of next price and future profit based on these parameters. In this model, the insider tries to learn parameters a_1, a_2, b_1, b_2, b_3 such that:

$$p_i = b_1 p_{i-1} + b_2 \Delta x_i + b_3 \tag{4.4}$$

$$\pi_{i+1} = a_1(v - p_i)^2 + a_2 \tag{4.5}$$

These equations are applicable for all periods except the last, since p_{N+1} is undefined, but we know that $\pi_{N+1} = 0$. From this we get:

$$\pi_i = \Delta x_i(v - b_1 p_{i-1} - b_2 \Delta x_i - b_3) + a_1(v - b_1 p_{i-1} - b_2 \Delta x_i - b_3)^2 + a_2 \tag{4.6}$$

The profit is maximized when the partial derivative with respect to the amount traded is 0. Setting $\frac{\partial \pi_i}{\partial(\Delta x_i)} = 0$:

$$\Delta x_i = \frac{-v + b_1 p_{i-1} + b_3 + 2a_1 b_2(v - b_1 p_{i-1} - b_3)}{2a_1 b_2^2 - 2b_2} \tag{4.7}$$

Now consider a repeated sequential auction game where each *episode* consists of N auctions. Initially the trader trades randomly for a particular number of episodes, gathering data as she does so, and then performs a linear regression on the stored data to estimate the five parameters above *for each auction*. The trader then updates the parameters periodically by considering all the observed data (see Algorithm 1 for pseudocode). The trader trades optimally according to her beliefs at each point in time, and any trade provides information on the parameters, since the price change is a noisy linear function of the amount traded. There may be benefits to sometimes not trading optimally in order to learn more.

This becomes a problem of both active learning (choosing a good Δx to learn more, and a problem of balancing exploration and exploitation.

```
Data: $T$: total number of episodes, $N$: number of auctions, $K$: number of initialization
      episodes, $D[i][j]$: data from episode $i$, auction $j$, $F_j$: estimated parameters for auction $j$
for $i = 1 : K$ do
    for $j = 1 : N$ do
        Choose random trading amount, save data in $D[i][j]$
    end
end
for $j = 1 : N$ do
    Estimate $F_j$ by regressing on $D[1][j] \dots D[K][j]$
end
for $i = K + 1 : T$ do
    for $j = 1 : N$ do
        Choose trading amount based on $F_j$, save data in $D[i][j]$
    end
    if $i \mod 5 = 0$ then
        for $j = 1 : N$ do
            Estimate $F_j$ by regressing on $D[1][j] \dots D[i][j]$
        end
    end
end
```

Algorithm 1: The equilibrium learning algorithm

4.4.3 An Approximate Algorithm

An alternative algorithm would be to use the same parameters for each auction, instead of estimating separate a's and b's for each auction (see Algorithm 2). Essentially, this algorithm is a learning algorithm which characterizes the state entirely by the last traded price and the liquidation price, irrespective of the particular auction number or even the total number of auctions. The value function of a state is given by the expected profit, which we know from equation 4.6. We can solve for the optimal action based on our knowledge of the system. In the last auction before liquidation, the insider trades knowing that this is the last auction, and does not take future expected profit into account, simply maximizing the expected value of that trade.

Stating this more explicitly in terms of standard reinforcement learning terminology, the insider assumes that the world is characterized by the following.

- A continuous state space where the state is $v - p$, where p is the last traded price.
- A continuous action space where actions are given by Δx, the amount the insider chooses to trade.

- A stochastic transition model mapping p and Δx to p' (v is assumed constant during an episode). The model is that p' is a (noisy) linear function of Δx and p.

- A (linear) value function mapping $(v-p)^2$ to π, the expected profit.

In addition, the agent knows at the last auction of an episode that the expected future profit from the next stage onwards is 0.

Of course, the world does not really conform exactly to the agent's model. One important problem that arises because of this is that the agent does not take into account the difference between the optimal way of trading at different auctions. The great advantage is that the agent should be able to learn with considerably less data and perhaps do a better job of maximizing finite-horizon utility. Further, if the parameters are not very different from auction to auction this algorithm should be able to find a good approximation of the optimal strategy. Even if the parameters are considerably different for some auctions, if the expected difference between the liquidation value and the last traded price is not high at those auctions, the algorithm might learn a close-to-optimal strategy. The next section discusses the performance of these algorithms, and analyzes the conditions for their success. I will refer to the first algorithm as the equilibrium learning algorithm and to the second as the approximate learning algorithm in what follows.

```
Data: $T$: total number of episodes, $N$: number of auctions, $K$: number of initialization
      episodes, $D[i][j]$: data from episode $i$, auction $j$, $F$: estimated parameters
for $i = 1 : K$ do
    for $j = 1 : N$ do
        Choose random trading amount, save data in $D[i][j]$
    end
end
Estimate $F$ by regressing on $D[1][] \dots D[K][]$ for $i = K + 1 : T$ do
    for $j = 1 : N$ do
        Choose trading amount based on $F$, save data in $D[i][j]$
    end
    if $i \mod 5 = 0$ then
        Estimate $F$ by regressing on $D[1][] \dots D[i][]$
    end
end
```

Algorithm 2: The approximate learning algorithm

4.5 Experimental Results

4.5.1 Experimental Setup

To determine the behavior of the two learning algorithms, it is important to compare their behavior with the behavior of the optimal strategy under perfect information. In order to elucidate the general properties of these algorithms, this section reports experimental results when there are 4 auctions per episode. For the equilibrium learning algorithm the insider trades randomly for 50 episodes, while for the approximate algorithm the insider trades randomly for 10 episodes, since it needs less data to form a somewhat reasonable initial estimate of the parameters.[5] In both cases, the amount traded at auction i is randomly sampled from a Gaussian distribution with mean 0 and variance $100/N$ (where N is the number of auctions per episode). Each simulation trial runs for 40,000 episodes in total, and all reported experiments are averaged over 100 trials. The actual parameter values, unless otherwise specified, are $p_0 = 75, \Sigma_0 = 25, \sigma_u^2 = 25$ (the units are arbitrary). The market-maker and the optimal insider (used for comparison purposes) are assumed to know these values and solve the Kyle difference equation system to find out the parameter values they use in making price-setting and trading decisions respectively.

4.5.2 Main Results

Figure 4-1 shows the average absolute value of the quantity traded by an insider as a function of the number of episodes that have passed. The graphs show that a learning agent using the equilibrium learning algorithm appears to be slowly converging to the equilibrium strategy in the game with four auctions per episode, while the approximate learning algorithm converges quickly to a strategy that is not the optimal strategy. Figure 4-2 shows two important facts. First, the graph on the left shows that the average profit made rises much more sharply for the approximate algorithm, which makes better use of available data. Second, the graph on the right shows that the average total utility being received is higher from episode 20,000 onwards for the equilibrium learner (all differences between the algorithms in this graph are statistically significant at a 95% level). Were the simulations to run long enough, the equilibrium learner would outperform the approximate

[5] This setting does not affect the long term outcome significantly unless the agent starts off with terrible initial estimates.

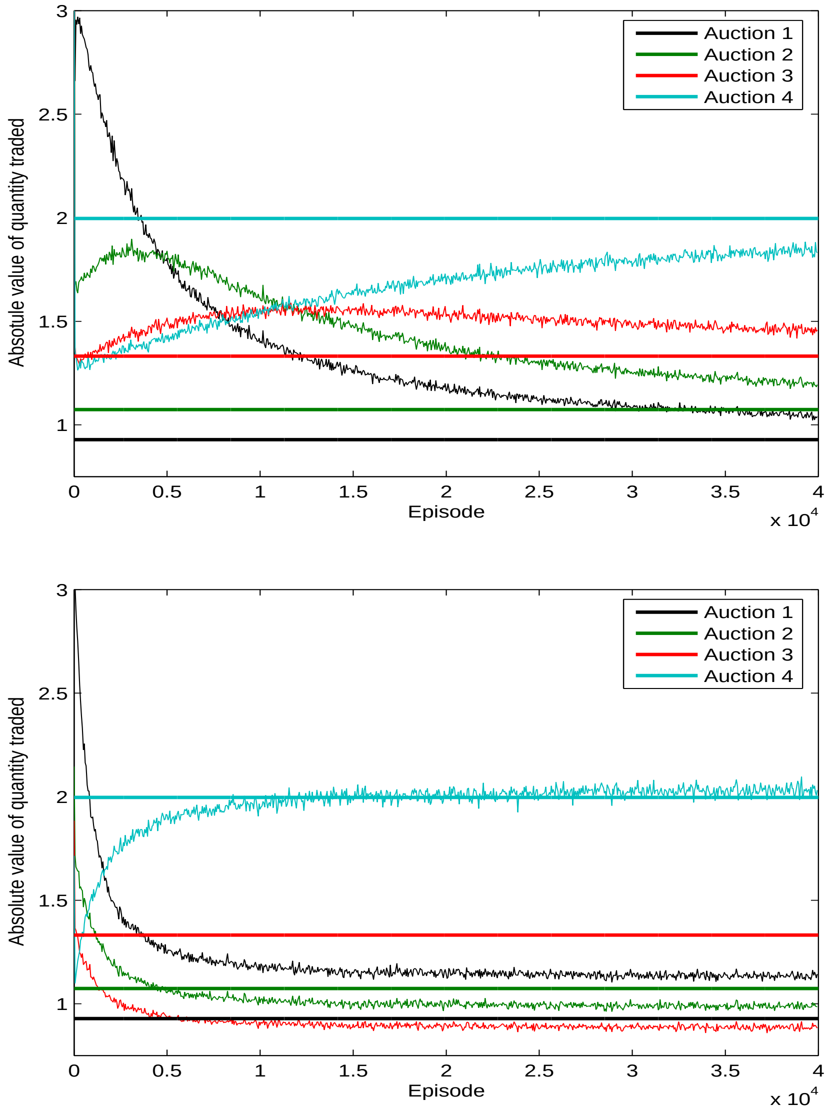
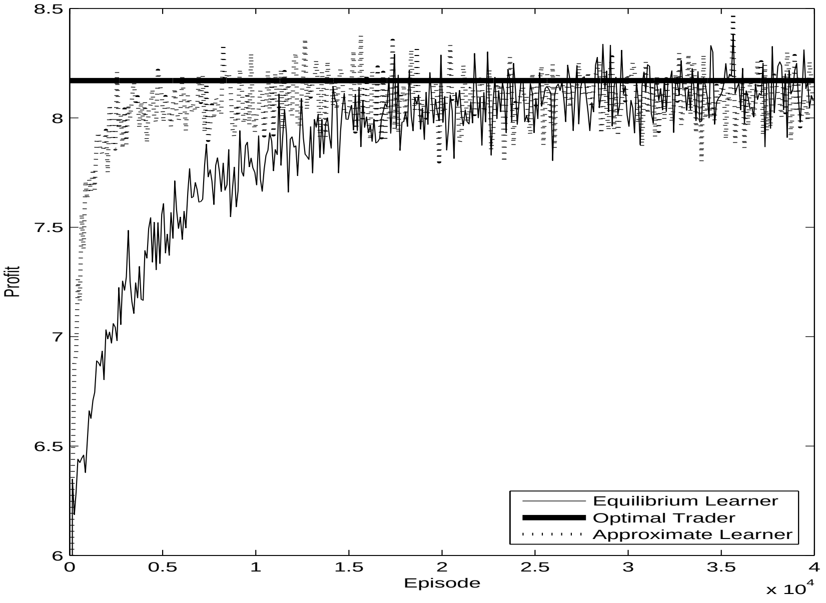


Figure 4-1: Average absolute value of quantities traded at each auction by a trader using the equilibrium learning algorithm (above) and a trader using the approximate learning algorithm (below) as the number of episodes increases.
Note: The thick lines parallel to the x-axis represent the average absolute value of the quantity that an optimal insider with full information would trade.



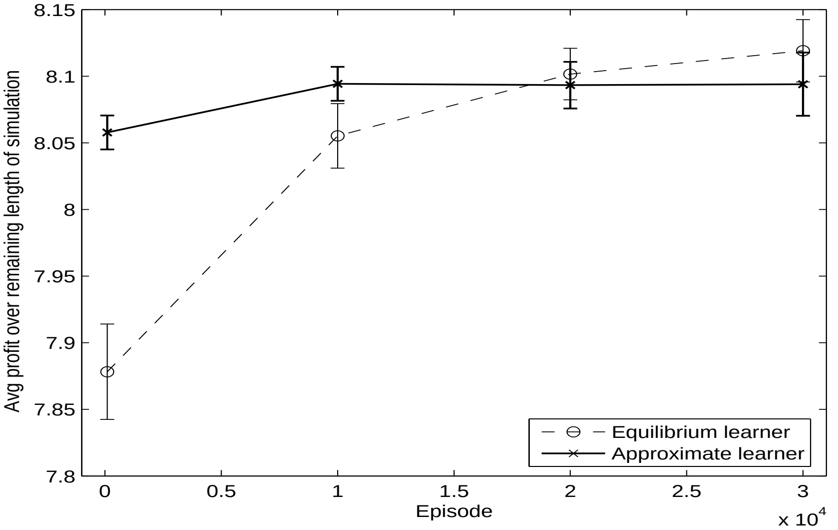


Figure 4-2: Above: Average flow profit received by traders using the two learning algorithms (each point is an aggregate of 50 episodes over all 100 trials) as the number of episodes increases. Below: Average profit received until the end of the simulation measured as a function of the episode from which measurement begins (for episodes 100, 10,000, 20,000 and 30,000).

learner in terms of total utility received, but this would require a huge number of episodes per trial.

Clearly, there is a tradeoff between achieving a higher flow utility and learning a representation that allows the agent to trade optimally in the limit. This problem is exacerbated as the number of auctions increases. With 10 auctions per episode, an agent using the equilibrium learning algorithm actually does not learn to trade more heavily in auction 10 than she did in early episodes even after 40,000 total episodes, leading to a comparatively poor average profit over the course of the simulation. This is due to the dynamics of learning in this setting. The opportunity to make profits by trading heavily in the last auction are highly dependent on not having traded heavily earlier, and so an agent cannot learn a policy that allows her to trade heavily at the last auction until she learns to trade less heavily earlier. This takes more time when there are more auctions. It is also worth noting that assuming that agents have a large amount of time to learn in real markets is unrealistic.

The graphs in Figures 4-1 and 4-2 reveal some interesting dynamics of the learning process. First, with the equilibrium learning algorithm, the average profit made by the agent slowly increases in a fairly smooth manner with the number of episodes, showing that the agent's policy is constantly improving as she learns more. An agent using the approximate learning algorithm shows much quicker learning, but learns a policy that is not asymptotically optimal. The second interesting point is about the dynamics of trader behavior — under both algorithms, an insider initially trades far more heavily in the first period than would be considered optimal, but slowly learns to hide her information like an optimal trader would. For the equilibrium learning algorithm, there is a spike in the amount traded in the second period early on in the learning process. This is also a small spike in the amount traded in the third period before the agent starts converging to the optimal strategy.

4.5.3 Analysis of the Approximate Algorithm

The behavior of the trader using the approximate algorithm is interesting in a variety of ways. First, let us consider the pattern of trades in Figure 4-1. As mentioned above, the trader trades more aggressively in period 1 than in period 2, and more aggressively in period 2 than in period 3. Let us analyze why this is the case. The agent is learning a strategy that makes the same decisions independent of the particular auction number (except for the last auction). At any auction other than the last, the agent is trying to choose Δx to maximize:

$$\Delta x(v - p') + W[S_{v,p'}]$$

where p' is the next price (also a function of Δx, and also taken to be independent of the particular auction) and $W[S_{v,p'}]$ is the value of being in the state characterized by the liquidation value v and (last) price p'. The agent also believes that the price p' is a linear function of p and Δx. There are two possibilities for the kind of behavior the agent might exhibit, given that she knows that her action will move the stock price in the direction of her trade (if she buys, the price will go up, and if she sells the price will go down). She could try to trade *against* her signal, because the model she has learned suggests that the potential for future profit gained by pushing the price away from the direction of the true liquidation value is higher than the loss from the one trade.[6] The other possibility is that she trades *with* her signal. In this case, the similarity of auctions in the representation ensures that she trades with an intensity proportional to her signal. Since she is trading in the correct direction, the price will move (in expectation) towards the liquidation value with each trade, and the average amount traded will go down with each successive auction. The difference in the last period, of course, is that the trader is solely trying to maximize $\Delta x(v - p')$ because she knows that it is her last opportunity to trade.

The success of the algorithm when there are as few as four auctions demonstrates that learning an approximate representation of the underlying model can be very successful in this setting as long as the trader behaves differently at the last auction. Another important question is that of how parameter choice affects the profit-making performance of the approximate algorithm as compared to the equilibrium learning algorithm. In order to study this question, I conducted experiments that measured the average profit received when measurement starts at various different points for a few different parameter settings (this is the same as the second experiment in Figure 4-2). The results are shown in Table 4.1. These results demonstrate especially that the profit-making behavior of the equilibrium learning algorithm is somewhat variable across parameter settings while the behavior of

[6] This is not really learnable using linear representations for everything unless there is a different function that takes over at some point (such as the last auction), because otherwise the trader would keep trading in the wrong direction and never receive positive reinforcement.

From episode	$\Sigma_0 = 5, \sigma_u^2 = 25$		$\Sigma_0 = 5, \sigma_u^2 = 50$		$\Sigma_0 = 10, \sigma_u^2 = 25$	
	Approx	Equil	Approx	Equil	Approx	Equil
100	0.986	0.964	0.986	0.983	0.986	0.964
10,000	0.991	0.986	0.990	0.997	0.990	0.986
20,000	0.991	0.992	0.990	0.999	0.989	0.992
30,000	0.991	0.994	0.989	1.000	0.989	0.994

Table 4.1: Proportion of optimal profit received by traders using the approximate and the equilibrium learning algorithm in domains with different parameter settings.

Note: The leftmost column indicates the episode from which measurement starts, running through the end of the simulation (40,000 periods).

the approximate algorithm is remarkably consistent. The advantage of using the approximate algorithms will obviously be greater in settings where the equilibrium learner takes a longer time to start making near-optimal profits. From these results, it seems that the equilibrium learning algorithm learns more quickly in settings with higher liquidity in the market.

4.6 Discussion

This chapter presents two algorithms that allow an agent to learn how to exploit monopolistic insider information in securities markets when agents do not possess full knowledge of the parameters characterizing the environment, and compares the behavior of these algorithms to the behavior of the optimal algorithm with full information. The results presented here demonstrate how domain knowledge can be very useful in the design of algorithms that learn from experience in an intrinsically online setting in which standard reinforcement learning techniques are hard to apply.

It would be interesting to examine the behavior of the approximate learning algorithm in market environments that are not necessarily generated by an underlying linear mechanism. For example, if many traders are trading in a double auction type market, would it still make sense for a trader to use an algorithm like the approximate one presented here in order to maximize profits from insider information?

I would also like to investigate what differences in market properties are predicted by the learning model as opposed to Kyle's model. Another direction for future research is the use of an online learning algorithm. Batch regression can become prohibitively expensive as the total number of episodes increases. While one alternative is to use a fixed window of past experience, hence forgetting the past, another plausible alternative is to use an online algorithm that updates the agent's beliefs at each time step, throwing away the example after the update. Under what conditions do online algorithms converge to the equilibrium? Are there practical benefits to the use of these methods?

Perhaps the most interesting direction for future research is the multi-agent learning problem. First, what if there is more than one insider and they are all learning?[7] Insiders could potentially enter or leave the market at different times, but we are no longer guaranteed that everyone other than one agent is playing the equilibrium strategy. What are the learning dynamics? What does this imply for the system as a whole? Another point is that the presence of suboptimal insiders ought to create incentives for market-makers to deviate from the complete-information equilibrium strategy in order to make profits. What can we say about the learning process when both market-makers and insiders may be learning?

[7] Theoretical results show that equilibrium behavior with complete information is of the same linear form as in the monopolistic case (Holden and Subrahmanyam 1992, Foster and Viswanathan 1996).

Chapter 5

A Search Problem with Probabilistic Appearance of Offers

5.1 Introduction.

Many job markets are structured in a manner where potential employees submit their applications to a number of employing firms simultaneously, and then wait to hear back from these firms. Firms themselves often make exploding offers that employees have to decide on in a short time-frame. Sometimes the firms will tell potential employees as soon as they are no longer under consideration, and in other cases they wait until the end of the search process to provide this information to applicants. The central question that we address in this chapter is this: *How much better off is an applicant if she is told every time she has been rejected by a firm, as opposed to only knowing when she receives offers?*

In order to study this problem, we construct a stylized model in which the decision problem faced by agents is a version of the problem variously referred to in the literature as the Cayley-Moser problem, the (job) search problem, the house hunting problem and the problem of selling an asset (Ferguson (1989)). In the original problem, a job applicant knows that there will be exactly n job opportunities, which will be presented to her sequentially. At the time each job is presented, she observes the utility she would receive from taking that job offer (one can think of it purely in terms of wages), and must decide immediately whether to accept the job offer or not. If she declines the offer, she may not go back to it. If she accepts it, she may not pick any of the subsequent offers. What is the strategy that maximizes her expected utility? This problem has been addressed for various distributions of offer values, and much of that work is summarized by Gilbert and Mosteller (1966).

The problem we consider is a variant of the above problem in which the total number of possible offers is known, but each offer appears only with a certain probability. This problem is motivated in part by models of two-sided matching markets like labor markets or dating markets. In particular, a problem considered by Das and Kamenica (2005) is one in which men are asked out on dates by women, and must respond immediately, but, while they have priors on the values of going out with particular women, they do not know the order in which women are going to appear, so they are not aware of whether or not a better option might come along in the future. This is because a better woman than the one currently asking a man out might either have already appeared in the ordering and not asked him out, or might appear later and not ask him out, or might appear later and ask him out. A similar problem can arise in faculty hiring processes for universities and colleges. Universities may not know whether applicants will definitely take positions that are offered, and, conversely, applicants do not know if they will receive an offer from any given university with which they interview. This chapter only looks at one side of this process without considering the dynamics involved when multiple agents interact, potentially strategically. Another motivation comes from thinking of the offers as investment opportunities (Gilbert and Mosteller (1966)). In particular, the continuous-time variant we discuss can be interpreted in terms of investment opportunities that arrive as a Poisson process where the decision-maker wants to choose the best one. To simplify the analysis, we assume that the probability that a particular offer appears, p, is the same across all offers and is independent of the actual value of the offer. The value of p may or may not be known to the applicant, and can be thought of as a measure of the “attractiveness” of the applicant or decision-maker.

Most of the previous research on search models focuses on solving an agent’s infinite horizon optimal stopping problem when there is either a cost to generating the next offer, or a discount factor associated with future utility (the book by DeGroot (1970) provides an account of much of this line of research). The problem we study here is a finite-horizon search problem with no cost to seeing more offers and no search frictions. The basic questions we pose and attempt to answer relate to how much the expected utility of the decision-maker changes between different information sets and different mechanisms. The question with regard to information sets can be thought of as follows. Suppose you interview with n firms that might want to hire you. Then the companies get ordered randomly and come along in that order and decide whether or not to make you an offer. How much would you pay to go from a situation in which you only saw which companies made you an offer (the *low information* variant) to a situation in which you saw, for each company, whether or not they chose to make you an offer (the *high information* variant)? Generalizing the two informational cases to continuous time provides good approximations for large n and insight into the value of information in these cases. It also allows us to make an interesting connection to a closely related problem called the *secretary problem*. We will also discuss the difference in expected utility between two different mechanisms. The exploding offer mechanism can lead to a substantial decline in the expected utility of a job-seeker compared to a mechanism in which she sees all the offers she will receive simultaneously and can choose from among them. What if you could pay to see the entire set of offers you would get simultaneously so that you could pick among them? How much should you be willing to pay? We will explicitly compare the expected loss in value in going from this *simultaneous choice* mechanism to the *sequential choice* mechanism that generates the stopping problem.

5.1.1 Related Work.

In the classical secretary problem (CSP), a decision-maker has to hire one applicant out of a pool of n applicants who will appear sequentially. Again, the decision-maker must decide immediately upon seeing an applicant whether to hire her or not. The key difference between secretary problems and search problems, as Ferguson (1989) notes, is that in secretary problems “the payoff depends on the observations only through their relative ranks and not otherwise on their actual values.” The most studied types of secretary problems are games with 0-1 payoffs, with the payoff of 1 being received if and only if the decision-maker hires the best applicant. The decision-maker’s optimal policy is thus one that maximizes the probability of selecting the best applicant.

A historical review of the early literature on secretary problems, including important references, can be found in the chapter by Gilbert and Mosteller (1966), as can solutions to many extensions of the basic problem, including the search problem (with finite and known n and no search costs) for various different distributions over the values of applicants. Many interesting variants of the original problem, mostly focusing on maximizing the probability of hiring the best applicant, have appeared in intervening decades. For instance, Cowan and Zabczyk (1978) introduce a continuous-time version of the problem with applicants arriving according to a Poisson process, which is closely related to the continuous-time problem we describe in Section 5.4. Their work has been extended by Bruss (1987) and by Kurushima and Ano (2003). Stewart (1981) studies a secretary problem with an unknown number of applicants which is also related to the problem we consider, but differs in the sense that he assumes n to be a random variable and the arrival times of offers to be i.i.d. exponential random variables, so that the decision-maker must maintain a belief distribution on n in order to optimize.

There has been considerable interest in explicitly modeling two-sided search and matching problems in the economics community. In particular, Burdett and Wright (1998) study two-sided search with nontransferable utility, which is relevant to our model because we assume exogenous offer values, implying that an employer cannot make her offer more attractive by, for example, offering a higher salary. While these issues are considered in greater detail in the next chapter, the book by Roth and Sotomayor (1990) and the chapter by Mortensen and Pissarides (1999) both provide excellent background on this line of literature in economics.

5.1.2 Contributions.

This chapter introduces a model of search processes where offers appear probabilistically and sequentially without explicit costs to sampling more offers, but with a limited number of possibilities that cannot be recalled. This is a good model for various job search and hiring processes where offers are "exploding" and search takes place during a fixed hiring season. Our main contributions can be summarized as follows:

a) We introduce two possible search processes, a "high information" process in which agents find out whether an offer appears or does not appear (this can also be thought of as agents being accepted or rejected) at each point in time, and a "low information" process in which agents only receive signals when an offer appears, so they do not know how many times they might have been rejected already.

b) We solve for the expected values of the low and high information processes for uniform and exponentially distributed offer values when agents know the underlying probability of offer appearance. We show that the expected utility in the low information process comes very close to the expected utility in the high information process, and that the gap is widest in a critical range of expected number of offers between four and six.

c) We show that when agents do not know the true probability of offer appearance the expected utility in the low information process declines substantially relative to the high information process. This shows that the most important informational value of rejections lies in helping decision makers estimate their own "attractiveness," when this attractiveness is measured in terms of the probability of offer appearance.

d) We introduce continuous time versions of the search processes, characterized by Poisson appearance of offers, and obtain closed form solutions for expected values of the high information processes. The solutions have a surprisingly simple form, which helps us gain insight into the dependence of the expected value on the offer arrival rate.

e) We evaluate the "competitive ratio" (in the sense used in computer science (Borodin and El-Yaniv 1998, e.g.)), which quantifies the relative reduction in the expected value, compared to the case where all offers are received simultaneously. We compare the competitive ratios of expected values in the stopping problem and the "simultaneous choice" problem to the ratios of expected values in the high and low information cases.

5.2 The Model.

We consider a search process in which a decision-maker (job-seeker) has to choose among n potential total offers, which appear sequentially. At each point in time, an offer either appears (with probability p), in which case its value w is revealed to the applicant, or does not appear (with probability $1-p$). If an offer does not appear, the applicant may or may not be told this fact. For the purposes of this chapter, we assume that all offers have an identical probability of appearance p, and that the values w are independently and identically distributed. We will consider two cases for the distribution of w, namely uniform and exponential. The job-seeker must decide immediately upon seeing an offer whether to accept it or not. If she accepts the offer, she receives utility w, and if she rejects it she may not recall that offer in the future.

We consider a number of variants of this process for the two distributions mentioned above. The two axes along which we parameterize the process are (a) whether or not the decision-maker knows the probability p of getting an offer; and (b) whether or not the decision-maker receives a signal when an offer does *not* appear. In the first case, the question is whether or not the decision-maker has to learn p. The second case essentially embodies two informational variants of the decision problem. In the high information variant, the decision-maker is told at each of the n stages whether an offer appeared or not. Therefore, she always knows the exact total number of possible offers that may yet appear. In the low information variant, the decision-maker is only informed when an offer appears — if the offer does not appear the decision-maker is not informed of this event. Thus, the decision-maker does not know how many offers are potentially left out of the n total offers. We will begin by showing results about the informational variants assuming that the decision-maker knows p. In each case we will consider two distributions over the offers w_i, one a uniform $[0, 1]$ and the other an exponential distribution with rate parameter α. For calibration, when we report numerical results, we assume $\alpha = 2$ so that the expected values of draws from both distributions are the same (0.5).

5.2.1 An Example Where $n = 2$.

As a motivating example, let us consider the case where $n = 2$, offer values are uniformly distributed in $[0, 1]$, and offers arrive with probability p. Later we will derive the expected values for general n. We can compute the expected value for an agent participating in the search process in the high and low information cases. In general, we will denote the expected value of the high information search process with n possible offers as H_n and the value of the low information process with n possible offers as L_n.

First, in the high information case, the agent knows that there are two time periods t in total, and she knows which time period she is in. At $t = 1$ the reservation value of an agent is her expected value if she declines the offer, which is just her expected value in the one period process. In the one period process, the agent should always accept any offer she receives, so the expected value is just the product of the probability that an offer appears and the expected value of that offer, or $0.5p$. Therefore, at $t = 1$, the agent should accept an offer only if it is greater than $0.5p$. Since offer values are distributed uniformly in $[0, 1]$, the probability that this is the case is $1 - 0.5p$. The expected value of the offer given that she does accept it is $(1 + 0.5p)/2$. The expected continuation value of the process if she rejects the offer is $0.5p$. Given that an offer arrives at $t = 1$ with probability p, the expected value of the search process is:

$$\begin{aligned} H_2 &= p\left((1-0.5p)\frac{1+0.5p}{2} + 0.5p(0.5p)\right) + (1-p)(0.5p) \\ &= \frac{1}{8}p^3 - \frac{1}{2}p^2 + p \end{aligned}$$

The low information case is somewhat more complicated. The major difference from the high information case is that the decision-maker's threshold for stopping at the first offer to appear changes. When the decision-maker sees the first offer (assuming she ever sees an offer and has to make a decision), she does not know if the offer is first in the ordering or if the offer is second in the ordering and the first offer did not appear. The probability that she will see another offer is then the probability that a second offer will appear given that one has appeared. Suppose we denote realized appearance/non-appearance outcomes by vectors of zeros and ones where the zeros indicate non-appearance and the ones indicate appearance. The total space of outcomes is $\{[0\ 0], [0\ 1], [1\ 0], [1\ 1]\}$. The appearance of one offer reduces the possible space of outcomes to $\{[0\ 1], [1\ 0], [1\ 1]\}$. The probability that a second offer appears given that a first has appeared is then $p^2/((1-p)p + p(1-p) + p^2) = p/(2-p)$. Therefore the threshold for the decision-maker to stop at the first offer to appear is $p/(4-2p)$.

The four possible cases for the low information process when $n = 2$ can be analyzed as follows (where, as in Section 5.2.1 0 denotes non-appearance of an offer and 1 denotes appearance):

1. $[0\ 0]$: Occurs with probability $(1-p)^2$ and has value 0.

2. $[0\ 1]$: Occurs with probability $(1-p)p$. The offer which appears is accepted with probability $1 - \frac{1}{2}\frac{p}{2-p}$, and if rejected, the utility received is 0. Therefore, the expected

value is:

$$\begin{aligned}
&\Pr\left(w > \frac{p}{2(2-p)}\right) E[w|w > \frac{p}{2(2-p)}] \\
&= \left(1 - \frac{p}{2(2-p)}\right)\left(\frac{p}{2(2-p)} + \frac{1}{2}(1 - \frac{p}{2(2-p)})\right) \\
&= \frac{3p^2 - 16p + 16}{8(2-p)^2}
\end{aligned}$$

3. $[1\ 0]$: Precisely the same argument as the previous case, with the same probability and expected value.

4. $[1\ 1]$: Occurs with probability p^2. In this case, if the first offer to appear is rejected, the second offer is automatically going to be selected. Therefore the expected value will be the sum of the above expected value and the expected value of the second given that the first is rejected (weighted by the probability of the first being rejected). The additional term is then:

$$\begin{aligned}
&(1 - \Pr(w > \frac{1}{2}\frac{p}{2-p}))(1/2) \\
&= \frac{p}{4(2-p)}
\end{aligned}$$

   Adding this to the expected value for the previous case and simplifying gives:

$$\frac{p^2 - 12p + 16}{8(2-p)^2}$$

Then the total expected value is:

$$\begin{aligned}
L_2 &= p(1-p)\frac{3p^2 - 16p + 16}{8(2-p)^2} + p^2\frac{p^2 - 12p + 16}{8(2-p)^2} \\
&= \frac{-5p^4 + 26p^3 - 48p^2 + 32p}{8(2-p)^2}
\end{aligned}$$

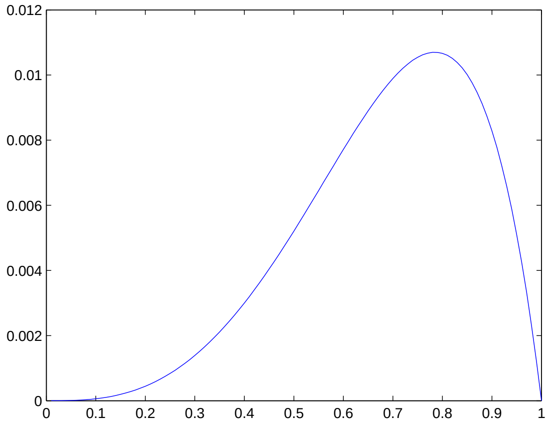


Figure 5-1: Expected value of the difference between the high and low information cases as a function of p for $n = 2$ and values independently drawn from a uniform $[0, 1]$ distribution.

The Value of Information

Simplifying the difference in expected values between the high and low information processes for $n = 2$, $D_2 = H_2 - L_2$, we find that:

$$D_2 = \frac{(p-1)p^3}{8(p-2)}$$

By setting the derivative to 0, we find that the difference is largest for $p = 0.7847$. Figure 5-1 shows the values of D_2 for p between 0 and 1.

5.3 The Search Process for General n.

This section provides the recursive solutions for the expected values of participating in the high and low information search processes. Solving for the expected value of the high information case is trivial, but it will serve as a point of comparison for the low information cases and will allow us to generalize to an interesting continuous time variant.

5.3.1 The High Information Case

This section derives the equations for computing the expected value of participating in the high information search process for general n and arbitrary p. In both cases, the base case is the expected value when $n = 1$, which is given by the product of the probability of an offer appearing (p) and the expected value of the offer given that it does appear (0.5 when offers are distributed uniformly in $[0, 1]$ and $1/\alpha$ when offers are distributed exponentially with rate parameter α). Also, in all cases when there are n possible offers remaining, the threshold for accepting an offer should be the expected value of the search process with $n - 1$ possible offers. Let w denote the value of the offer:

$$H_n = p\left[\Pr(w > H_{n-1})E(w|w > H_{n-1}) + (1 - \Pr(w > H_{n-1}))H_{n-1}\right] + (1 - p)H_{n-1}$$

Uniform $[0, 1]$ Distribution

In this case,

$$\begin{aligned} \Pr(w > H_{n-1}) &= 1 - H_{n-1} \\ E(w|w > H_{n-1}) &= H_{n-1} + \frac{1 - H_{n-1}}{2} = \frac{1 + H_{n-1}}{2} \end{aligned}$$

This gives us:

$$\begin{aligned} H_n &= p((1 - H_{n-1})\frac{1 + H_{n-1}}{2} + H_{n-1}^2 + (1 - p)H_{n-1} \\ &= p\frac{1 + H_{n-1}^2}{2} + (1 - p)H_{n-1} \qquad (5.1) \end{aligned}$$

and we know that, in the base case $H_1 = 0.5p$.

Exponential Distribution with Rate Parameter α

In this case,

$$\begin{aligned} \Pr(w > H_{n-1}) &= \int_{H_{n-1}}^{\infty} \alpha e^{-\alpha x}\,dx \\ &= e^{-\alpha H_{n-1}} \\ E(w|w > H_{n-1}) &= \int_{0}^{\infty} \alpha e^{-\alpha x}(x + H_{n-1})\,dx \quad \text{(Using the memorylessness property)} \\ &= \frac{1}{\alpha} + H_{n-1} \end{aligned}$$

Therefore,

$$
\begin{aligned}
H_n &= p[e^{-\alpha H_{n-1}}(\frac{1}{\alpha} + H_{n-1}) + (1 - e^{-\alpha H_{n-1}})H_{n-1}] + (1-p)H_{n-1} \\
&= p[\frac{1}{\alpha}e^{-\alpha H_{n-1}} + H_{n-1}] + (1-p)H_{n-1} \\
&= p\frac{1}{\alpha}e^{-\alpha H_{n-1}} + H_{n-1} \qquad (5.2)
\end{aligned}
$$

and we know that in the base case, $H_1 = p\frac{1}{\alpha}$.

5.3.2 The Low Information Case

In the low information process with n total possible offers, any state at which the decision-maker has to take a decision can be completely characterized by n and by the number of offers that have appeared thus far, denoted by k. The expected value of not stopping at offer k (state (n, k)) is given by the product of the probability that state $[n, k+1]$ will be reached (if not, the decision-maker sees no more offers and gets utility 0) and the expected value $L(n, k+1)$.

The probability that state $(n, k+1)$ is reached given that (n, k) was reached is:

$$
q_k = \frac{\sum_{i=k+1}^{n} \binom{n}{i} p^i (1-p)^{n-i}}{\sum_{i=k}^{n} \binom{n}{i} p^i (1-p)^{n-i}}
$$

The continuation value of the process (the expected value of not stopping) is $q_k L(n, k+1)$. We know that $L(n, n) = 0.5$ for offers distributed uniformly in $[0, 1]$ and $L(n, n) = 1/\alpha$ for offers distributed exponentially with rate parameter α, so we can compute the expected value recursively. Let $z_k = q_k L(n, k+1)$ and w be the value of the kth offer to appear. Then, for the case where offers are distributed uniformly on $[0, 1]$:

$$
\begin{aligned}
L(n, k) &= \Pr(w > z_k)E[w|w > z_k] + \Pr(w < z_k)z_k \\
&= (1 - z_k)\frac{1 + z_k}{2} + z_k^2 \\
&= \frac{1}{2}(1 + z_k^2)
\end{aligned}
$$

Similarly, for the case where offers are distributed exponentially with rate parameter

α:

$$L(n,k) = \Pr(w > z_k)E[w|w > z_k] + \Pr(w < z_k)z_k = \frac{1}{\alpha}e^{-\alpha z_k} + z_k$$

The expected value of the n offer low information process is then $L_n = L(n,0)$.

5.3.3 The Value of Information

Figure 5-2 shows the value of information for various different n and for the two distributions we consider. We can see that the critical region where the value of information is highest is reached at lower p for higher n – this happens when the expected value of the process is in an intermediate range. A rule of thumb is that the value of information is highest when the expected number of offers, np, is in the range of 4 to 6. We formalize this in a continuous time setting in the next section. The most important observation is that the information does not appear to be critical to making a good decision. Even in the worst of all the cases in Figure 5-2, the loss from participating in the low information process is only about 3%. Therefore, it seems clear that participants do not suffer great declines in expected utility from not being told when they are rejected, as long as they know the true probability p of offers appearing. In Section 5.5 we consider the case where p is unknown and show that the loss can be significantly higher.

5.4 Continuous Time Variants.

The natural continuous time limits of the process introduced in Section 2 involves Poisson arrivals of offers over a limited time horizon. We assume that offers arrive according to a Poisson process with arrival rate λ in the time interval $[0, 1]$. Again, the offer payoffs are sampled from either a uniform $[0, 1]$ distribution or an exponential distribution with rate parameter α, and the decision-maker has to decide upon seeing each offer whether to stop and accept that offer or continue searching. These continuous time variants allow us to abstract away from the particular number of possible offers and think in terms of the expected number of offers. We show that the high information processes have closed form solutions for the expected value at any point in time that allow us to gain insight into the dependence of the expected value on the expected number of offers. In this section we

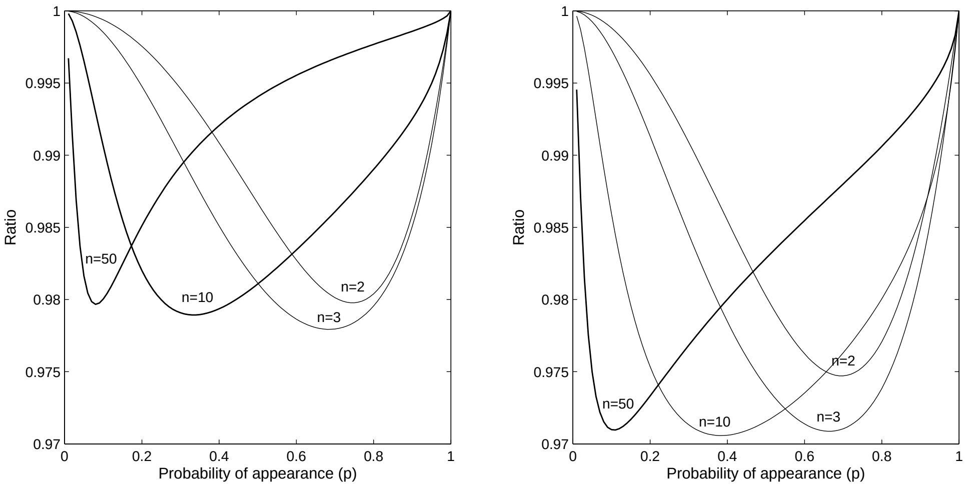


Figure 5-2: The ratio of the expected values of the low and high information processes for different values of n and p, for offer values drawn from the uniform $[0, 1]$ distribution (left) and the exponential distribution with rate parameter 2 (right).

study and solve for the expected values of a decision-maker in the high and low information continuous time search processes, and discuss the relation between these processes and the discrete variants discussed above.

5.4.1 The High Information Variant

In the high information variant, each time an offer appears, the decision-maker gets to see both the value of the offer, say w, and the precise time of appearance, t. The decision-maker should stop if w is greater than the continuation value $v(t)$. At any time t, to derive the continuation value we need to consider when the next offer will be received. At time t, the probability density function of the time of the next offer arrival (if any) is $\lambda e^{-\lambda(x-t)}$ for $x \leq 1$ (any density after 1 effectively "gets lost"). The value of receiving an offer at time x can be derived as in Section 3.

Uniform Distribution

Let w be the random value of an offer received at time x. The value of receiving such an offer is:

$$\begin{aligned}
&\Pr(w > v(x))E[w|w > v(x)] + \Pr(w < v(x))v(x) \\
&= (1 - v(x))(v(x) + \frac{1 - v(x)}{2}) + v^2(x) \qquad \text{(because } w \sim U[0, 1]\text{)} \\
&= \frac{1}{2}(1 - v^2(x)) + v^2(x) \\
&= \frac{1}{2}(1 + v^2(x))
\end{aligned}$$

The continuation value at time t must satisfy:

$$v(t) = \int_t^1 \lambda e^{-\lambda(x-t)} \frac{1}{2}(1 + v^2(x))\, dx$$

Therefore,

$$e^{-\lambda t} v(t) = \frac{1}{2}\lambda \int_t^1 e^{-\lambda x}(1 + v^2(x))\, dx$$

Differentiating with respect to t,

$$(-\lambda v(t) + v'(t))e^{-\lambda t} = -\frac{1}{2}\lambda e^{-\lambda t}(1 + v^2(t))$$

Since $v(1) = 0$ and $v \in [0, 1]$,

$$v'(t) = -\frac{1}{2}\lambda(v(t) - 1)^2$$

Or

$$\left(-\frac{1}{v(t) - 1}\right)' = -\frac{1}{2}\lambda$$

Integrating from t to 1,

$$\frac{1}{v(1) - 1} - \frac{1}{v(t) - 1} = \frac{1}{2}\lambda(1 - t)$$

Which gives us the solution:

$$v(t) = \frac{1-t}{\frac{2}{\lambda}+1-t} \tag{5.3}$$

Therefore the value of a process with arrival rate λ is $v(0) = \lambda/(\lambda+2)$.

Exponential Distribution

The logic is exactly the same as above, except that with an exponential distribution with rate parameter α the continuation value at time t must satisfy

$$v(t) = \int_t^1 \lambda e^{-\lambda(x-t)}(\frac{1}{\alpha}e^{-\alpha v(x)} + v(x))\,dx$$

Differentiating with respect to t, we get:

$$\Rightarrow v'(t) = -\frac{\lambda}{\alpha}e^{-\alpha v(t)}$$

or

$$v(t) = \frac{1}{\alpha}\log(-\lambda t + c)$$

where c is a constant of integration. Using the boundary condition $v(1) = 0$

$$v(t) = \frac{1}{\alpha}\log(-\lambda t + \lambda + 1) \tag{5.4}$$

Therefore, in this case the value of a process with arrival rate λ is $v(0) = \log(1+\lambda)/\alpha$.

5.4.2 The Low Information Variant

In the low information variant of the continuous time process, the decision-maker knows only the number of offers she has received, not the precise time t at which any of the offers were received. Therefore, any time that a decision has to be made, the state is completely characterized by the number of offers received so far. Let the value of a process in which k offers have been received so far (but the decision-maker has not yet seen the value of the kth offer) be denoted by $v[k]$. Let w be the (unknown) value of the current offer. The continuation value of the process can then be computed in a manner exactly analogous to the discrete time case. Let

$$
\begin{aligned}
q_k &= \Pr(\text{At least one more offer will be received} \mid k \text{ offers were received}) \\
z_k &= q_k v[k+1]
\end{aligned}
$$

Then

$$v[k] = \Pr(w > z_k) E[w | w > z_k] + \Pr(w < z_k) z_k$$

For offers distributed uniformly in $[0, 1]$, we have

$$v[k] = \frac{1}{2}(1 + z_k^2) \tag{5.5}$$

For offers distributed exponentially with rate parameter α, we have

$$v[k] = \frac{1}{\alpha} e^{-\alpha z_k} + z_k \tag{5.6}$$

There are two differences from the discrete case. First, q_k must be computed differently, because we now have Poisson arrivals. Let $f(k)$ be the Poisson probability mass function (the probability of getting exactly k offers) and $F(k)$ be the cumulative distribution function, for a particular value of λ. Then

$$
\begin{aligned}
q_k &= \frac{1 - F(k)}{1 - F(k-1)} \\
&= 1 - \frac{f(k)}{1 - F(k-1)}
\end{aligned}
$$

These are easily computed since we know that $f(k) = e^{-\lambda} \frac{\lambda^k}{k!}$ and $F(k) = \sum_{i=0}^{k} e^{-\lambda} \frac{\lambda^i}{i!}$.

The second difference from the binomial case is that we do not have an obvious base case, such as the case where n offers out of n are received, from which we can start a

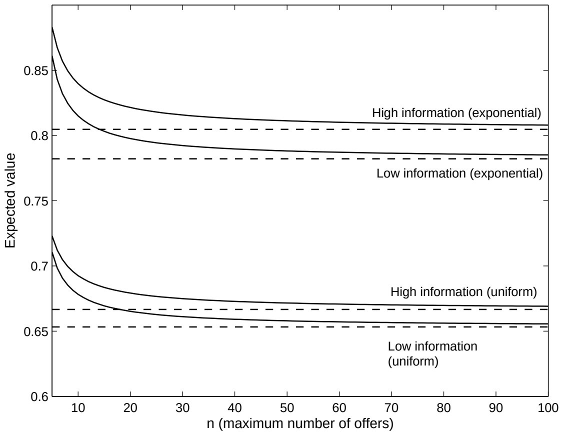


Figure 5-3: Expected values of the high and low information processes in continuous and discrete time holding $\lambda = pn$ constant (at $\lambda = 4$).

Note: Dashed lines represent the values of the continuous time processes and solid lines the values of the discrete time processes

backwards recursion. However, we can show that $\lim_{k\to\infty} q_k = 0$.

$$\begin{aligned}
\lim_{k\to\infty} q_k &= \lim_{k\to\infty} \frac{1 - F(k)}{1 - F(k-1)} \\
&= \lim_{k\to\infty} \frac{-f'(k)}{-f'(k-1)} && \text{(Applying L'Hospital's Rule)} \\
&= \lim_{k\to\infty} \frac{\lambda}{k} = 0
\end{aligned}$$

Therefore, it is reasonable to approximate the actual value by assuming some threshold K such that $q_K = 0$ (the threshold K may depend on the particular value of λ). To convey a sense of the practical value of the threshold K we should note that a threshold such as $K = 200$ enables us to compute the expected values to a high degree of precision for λ as high as 100, since the probability of getting more than 200 offers is completely negligible for $\lambda = 100$. For higher λ values one would need to use higher thresholds.

5.4.3 Relation to the Discrete Time Process

Figure 5-3 shows that the expected values of the discrete time processes converge to the expected values of the continuous time variants as $n \to \infty$, while holding $\lambda = pn$ constant (other values of λ yield similar graphs). We can also show formally that the expected value of the continuous time high information process serves as a lower bound for the expected value of the discrete time high information process when offer values are distributed uniformly in $[0, 1]$.

Theorem 1. *The value of the high information discrete-time process for given p and n is greater than the value of the high information continuous-time process with $\lambda = pn$, when offer values are drawn from a uniform $[0, 1]$ distribution.*

Proof. Let us denote the value of the discrete-time process by $H[i]$, where i is the number of offers that have appeared in the past, and the continuation value of the continuous time process at time t by $v(t)$. We want to show that, when $\lambda = pn$, $H[0] > v(0)$. We shall proceed by induction, showing that, for given p and n,

$$H[i] > v(i/n), \qquad \forall i < n$$

We know that

$$v(t) = \frac{1-t}{2/\lambda + 1 - t} = \frac{\lambda(1-t)}{2 + \lambda(1-t)}$$

For $i = n-1$, we have $H[n-1] = 0.5p$ because the value is sampled from the uniform $[0, 1]$ distribution, and

$$\begin{aligned} v\left(\frac{n-1}{n}\right) &= \frac{\lambda(1 - \frac{n-1}{n})}{2 + \lambda(1 - \frac{n-1}{n})} \\ &= \frac{\frac{\lambda}{n}}{2 + \frac{\lambda}{n}} \\ &= \frac{p}{2+p} && \text{(because } \lambda = np\text{)} \\ &< \frac{1}{2}p && \text{(because } p \in [0,1]\text{)} \\ &= H[n-1] \end{aligned}$$

Now, given that $H[i] > v(i/n)$ we have to show that $H[i-1] > v\left((i-1)/n\right)$ for integral

$i \geq 1$, which will complete the proof. Let $X = v(i/n)$. Then

$$\begin{aligned}
H[i-1] &= p\left(\frac{1}{2}(1 + H[i]^2)\right) + (1-p)H[i] \\
&> \frac{1}{2}p + \frac{1}{2}pX^2 + (1-p)X && \text{(inductive hypothesis)} \\
&= \frac{1}{2}p(1 + X^2 - 2X) + pX + X - pX \\
&= \frac{1}{2}p(1-X)^2 + X
\end{aligned}$$

In order to complete the induction step, it is therefore sufficient to show that

$$\frac{1}{2}p(1-X)^2 > v\left((i-1)/n\right) - X$$

Simplifying the right hand side, we get

$$\begin{aligned}
v\left((i-1)/n\right) - X &= \frac{2\lambda n}{(2n + \lambda n - \lambda i + \lambda)(2n + \lambda n - \lambda i)} \\
&= \frac{2\lambda n}{(2n + \lambda n - \lambda i)^2 + \lambda(2n + \lambda n - \lambda i)} \\
&< \frac{2\lambda n}{(2n + \lambda n - \lambda i)^2} \\
&= \frac{1}{2}p(1-X)^2
\end{aligned}$$

which completes the proof.

□

Conjecture 1. *The value of the high information discrete-time process for specified p and n is greater than the value of the high information continuous-time process with $\lambda = pn$ when offer values are drawn from an exponential distribution.*

We also conjecture that the low information expected values for the continuous time variants may also serve as lower bounds for the discrete time cases. The intuition is that the continuous time versions have a higher variance for the number of offers appearing (np as opposed to $np(1-p)$), which is why they yield lower expected values, especially for high values of p (corresponding to lower n since the product is held constant).

Interestingly, a difficult variant of the secretary problem (with the goal of maximizing

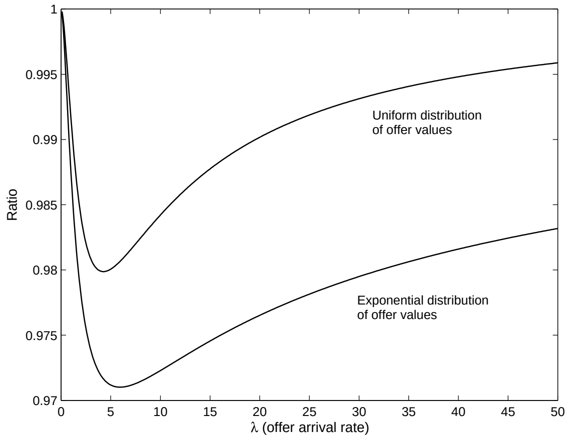


Figure 5-4: Ratio between expected values of the low and high information cases as a function of λ for the continuous time processes.

the probability of selecting the best candidate) has been proposed and solved in continuous time by Cowan and Zabczyk (1978), and generalized by others (Kurushima and Ano (2003), Bruss (1987)). Our problem bears the same relation to this problem as the search problem with non-probabilistic appearance of offers (Gilbert and Mosteller (1966)) (recovered by using $p = 1$ in our case) does to the classical secretary problem.

5.4.4 The Value of Information

As n increases, the continuous time processes become a better approximation to the discrete time cases, and give us an opportunity to study general behavior without worrying about the specific interactions of n and p. Figure 5-4 shows the difference in expected value between the high and low information processes in continuous time expressed as a ratio. We can see that information is most important in a critical range of λ (between around $\lambda = 3$ and $\lambda = 10$, peaking between 4 and 6) for both distributions and the importance of information drops off quickly thereafter. Information is also not particularly important if the expected total number of offers is very small. This confirms our intuitions from the discrete time cases.

5.5 What if p is Unknown?

In some search problems of the kind we have been discussing, the decision-maker may not have a good estimate of the probability p that any given offer will appear. In this case the decision-maker must update her estimate of p while also making decisions as before, with each decision based on her current estimate. This can greatly change the complexion of the problem, and especially of the value of information, because now knowing when an offer will not appear is not only useful for the decision problem, it is also useful for the problem of learning p to help in future decisions.

We will assume that a decision-making agent starts with a prior on p. In the experiments we report here, this prior always starts as a uniform $[0, 1]$ distribution. First, let us consider the high information case and two possible ways of representing and updating the agent's beliefs about p.

5.5.1 The High Information Case

Using a Beta Prior

One possibility is to use a parameterized distribution. The ideal one for this case is the Beta distribution, because the two possible events at each time are success and failure, and the Beta distribution is its own conjugate and is particularly easy to update for this case. If the prior distribution on p before seeing the outcome of a binary event is a $\beta(i, j)$ distribution, then the posterior becomes $\beta(i + 1, j)$ in the event of a success and $\beta(i, j + 1)$ in the event of a failure. The $\beta(1, 1)$ distribution is uniform $[0, 1]$, and so the agent can start with that as the initial prior. Then, in order to compute the expected value of the game at any time after s successes and f failures have been seen, the agent only needs to additionally know the distribution of offer values and the total possible number of offers. However, the dynamic programming recursions are somewhat different than those in earlier sections. An agent who receives an offer and rejects it has a different expected value than an agent who does not receive an offer, due to the informational difference in her next estimate of p.

The value function is parameterized by n, the maximum number of possible offers remaining, s, the number of successes seen so far, and f, the number of failures seen so far.

For offer values distributed uniformly in $[0, 1]$ the expected value of the game is given by:

$$V(n,s,f) = \int_0^1 \eta(x,s{+}1,f{+}1)\left(x\frac{1}{2}\left(1+V^2(n-1,s+1,f)\right)+(1-x)V(n-1,s,f+1)\right)dx$$

where $\eta(x,s+1,f+1)$ represents the density function of the Beta $(s+1,f+1)$ distribution at x, that is the posterior after seeing s successes and f failures when starting with a Beta $(1,1)$ prior.

Similarly, for offer values distributed exponentially with rate parameter α, the expected value is given by:

$$\begin{aligned}V(n,s,f) = \int_0^1 \eta(x,s+1,f+1)\Bigg(x\left(\frac{1}{\alpha}e^{-\alpha V(n,s+1,f)}+V(n,s+1,f)\right)+\\ (1-x)V(n-1,s,f+1)\Bigg)\,dx\end{aligned}$$

To actually compute these values, we can use a discrete approximation to the integral along the probability axis. V can be computed recursively backwards.

Using a Discrete Non-parametric Prior

Another option is to simply use a discrete prior to begin with, and use the appropriate belief vector for subcomputations. The key to making this computation efficient is to note that an agent's beliefs will always be the same when s successes and f failures have been observed, regardless of the path. Therefore, the posterior at this time can be computed as:

$$\Pr(p=x\,|\,s,f) = \frac{\Pr(s\text{ successes out of }s+f|p=x)\Pr(p=x)}{\Pr(s\text{ successes out of }s+f)}$$

Here $\Pr(p=x)$ is the original prior.

5.5.2 The Low Information Case

In the low information case, the only information available to update the decision-maker's beliefs about p is the number of offers made so far. In this case, she must update as follows:

$$\Pr(p=x\,|\,s\text{ offers}) = \frac{\Pr(\text{at least }s\text{ offers}\,|\,p=x)\Pr(p=x)}{\Pr(\text{at least }s\text{ offers})}$$

The probability of getting at least s offers given that $p = x$ can be computed using the cumulative distribution function of the binomial distribution. Also note that the agent's beliefs about p will be the same every time that s successes have been observed.

5.5.3 Evaluating Performance

In order to estimate the expected utility received, we need to specify the form of learning the agent uses, the information available to the agent, and the true probability p of offer appearance. Then for particular values of p and n we can proceed by evaluating the expected value of a Markov chain in which states are characterized by the number of successes and failures seen so far. In either the high or low information cases, the agent will have a certain reservation value at each state that is completely dependent on the number of successes (in both cases) and failures (in the high information case) observed thus far. Then the expected value of being in that state can be computed based just on the agent's reservation value and the true underlying distribution of offer values and probability of offer appearance.

For a given true underlying p and n and a given initial prior we can describe the process as a Markov chain whose state consists of the number of past successes and failures (s and f, respectively). In the high information case, the reservation value of an agent is dependent on s, f, and n, while in the low information case, the reservation value only depends on s and n. Suppressing the dependence on n, denote the reservation value in the high information case by $R_h(s, f)$ and in the low information case by $R_l(s)$. The reservation value at state s is the expected value of the process if the agent does not accept an offer that appears. This is important because the appearance of the offer is itself informative.

Let w be the value of an offer that does appear. Let V_s denote the value of state $(s+1, f)$ and V_f denote the value of state $(s, f+1)$. The value of state (s, f) is 0 when $s + f \geq n$.

Then in the high information case, the value of state (s, f) is:

$$p\big(\Pr(w > R_h(s, f))E[w|w > R_h(s, f)] + \Pr(w < R_h(s, f))V_s\big) + (1 - p)V_f$$

In the low information case, the value of state (s, f) is (the decision-making agent does

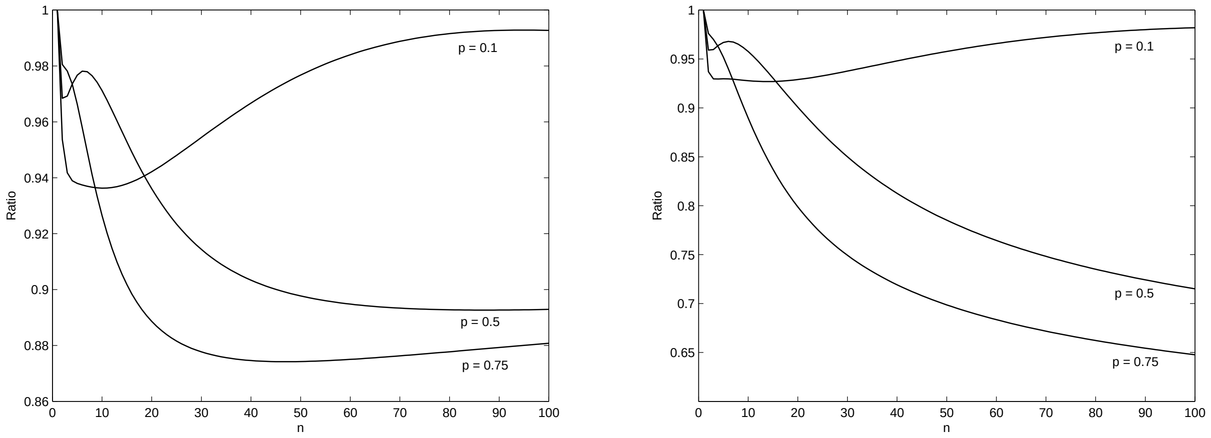


Figure 5-5: Ratio of expected values of the high information and low information search processes when p is unknown, the agent starts with a uniform prior over $[0,1]$ on p, and offers are drawn from a uniform $[0,1]$ distribution (left) or an exponential distribution with rate parameter $\alpha = 2$ (right).
Note that the y-axis is significantly different in the two cases.

not have access to f, but we use it when evaluating the chain):

$$p\left(\Pr(w > R_l(s))E[w|w > R_l(s)] + \Pr(w < R_l(s))V_s\right) + (1-p)V_f$$

The actual reservation values at any given state can be precomputed and stored in a table, since they are completely independent of the value of the state. Then the Markov chain can be evaluated based on this table and the known true probability p.

There is no difference in the expected values for the high information game when using the Beta prior and when using the nonparametric prior, so we report results only from the use of the Beta prior. We first report results when agents start with a uniform prior over $[0,1]$ for p.

Figure 5-5 shows results in terms of the value of information (corresponding to those in Figure 5-2 for the case of known p) for the uniform $[0,1]$ distribution and the exponential distribution with $\alpha = 2$. There are three cases shown in each graph, corresponding to three true underlying probabilities. The first important thing to note is that there are much larger differences in the expected value between the high and low information cases than there were in the case where agents knew p beforehand. For the case of the uniform distribution, in both cases expected values are increasing and are bounded by 1, so the difference does not become as dramatic as for the exponential distribution. The reason why the ratios of expected values are so different is because in the high information case it is “easy” to learn p by updating your estimate based on seeing both when offers appear and they do not. In the low information case, the only information available does not help the agent nearly as much in updating her estimates.

A second interesting effect we see in the graphs is that the ratio declines precipitously for higher true values of p, especially for the exponential distribution. The reason for this huge decline is the tradeoff that an agent must make in her estimate – if there is a larger n then the agent is of course likely to receive more offers, so her threshold should be higher. However, the higher value of n could also “explain away” the appearance of more offers, so that the agent does not realize that the true underlying p is higher. Consider an agent receiving her fourth offer when $n = 50$. Her threshold for accepting the offer cannot depend on p because she does not know p. This leads to decisions that look relatively “better” for different true underlying values. The same rule makes the agent perform better (relative to the high information case) for $p = 0.5$ than for $p = 0.1$ when $n = 5$, but much worse when $n = 20$. When $p = 0.1$ and $n = 5$, the agent is not sufficiently willing to accept offers, because a large part of the mass of her probability beliefs is on $p > 0.1$. However, when $p = 0.5$ and $n = 20$, the agent becomes too conservative and not risky enough in *rejecting* offers, because the appearance of offers does not necessarily tell her that p is higher, it might just be a function of the fact that there are a large number of total possible offers. She thus becomes more likely to take an offer that is not actually of high enough value.

A question that arises in this context is that of what happens when the agent has a less diffuse prior. In many ways this might correspond to a more realistic situation. Suppose she knows that her true probability of receiving offers is definitely between 0.4 and 0.6 when it is actually 0.5. We studied this question by calculating the ratios of expected values of the low and high information processes when the agent starts with a uniform prior on $[0.4, 0.6]$ (modeled using discrete probability masses, and using the nonparametric technique in the high information case as well as the low information case). The results are shown in Figure 5-6. We can see that the ratio actually appears to remain constant (and significantly higher than before) as n increases, showing that the expected value goes down much less as we move to the low information case, as we would expect given that

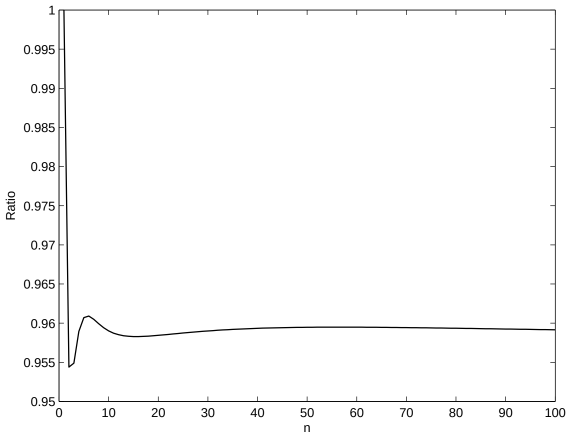


Figure 5-6: Ratio of expected values of the high information and low information search processes when p is unknown, the agent starts with a uniform prior over $[0.4, 0.6]$ on p, and offers are drawn from an exponential distribution with rate parameter $\alpha = 2$.

the case of known p is the limit of concentrating the prior.

5.6 Comparison of Mechanisms: Sequential vs. Simultaneous Choice

So far, we have considered the loss from lack of information within a particular mechanism, a sequential choice mechanism which introduces a stopping problem for the decision maker. In this section we ask a different set of questions – namely, what is the loss from using the sequential choice mechanism itself? This has been an important consideration for previous work on secretary problems and on optimal stopping more generally. We will focus on the difference between the high information case with sequential choice and what we call the simultaneous choice case, in which all offers appear simultaneously, and the decision maker can simply choose the best one. In continuous time, the simultaneous choice case is simply one in which all the appearances are realized, and then at time 1, the decision maker gets to choose the best out of all the realized options. It can also be thought of as allowing the decision-maker to backtrack to previous choices.

First let us consider the continuous time case. What is the expected value of participating in a simultaneous choice process with arrival rate λ? It is the sum over all k of the probability that exactly k offers appear and the expected value given that exactly k offers appear. For all continuous time models, offers arrive as a Poisson process, and the probability of exactly k offers is given by $\frac{e^{-\lambda}\lambda^k}{k!}$.

For offer values distributed uniformly in $[0,1]$, if k choices are available, the expected value is $\frac{k}{k+1}$ (from the order statistic of the uniform distribution). Then the expected value of the process is:

$$\begin{aligned}\sum_{i=0}^{\infty} \Pr(i \text{ successes})\frac{i}{i+1} &= \sum_{i=0}^{\infty} \frac{e^{-\lambda}\lambda^i i}{i!(i+1)} \\ &= \frac{e^{-\lambda}}{\lambda}\sum_{i=0}^{\infty}\left[\frac{(i+1)\lambda^{i+1}}{(i+1)!} - \frac{\lambda^{i+1}}{(i+1)!}\right] \\ &= 1 - \frac{1-e^{-\lambda}}{\lambda}\end{aligned}$$

The expression for the expected value for offers distributed exponentially with rate parameter α is slightly more complex. First note that the distribution function for the maximum of k such random variables is:

$$f(x) = k[1-e^{-\alpha x}]^{k-1}\alpha e^{-\alpha x}$$

Therefore the expected value of the maximum is:

$$k\alpha\int_0^{\infty}[e^{-\alpha x}(1-e^{-\alpha x})^{k-1}x]\,dx = \frac{\mathcal{H}_k}{\alpha}$$

where $\mathcal{H}_k$ represents the kth harmonic number.

Then the expected value is given by:

$$\begin{aligned}\sum_{i=0}^{\infty} \Pr(i \text{ successes})\frac{H_i}{\alpha} &= \frac{e^{-\lambda}}{\alpha}\sum_{i=0}^{\infty}\frac{\lambda^i H_i}{i!} \\ &= \frac{1}{\alpha}[\gamma + \Gamma(0,\lambda) + \log(\lambda)]\end{aligned}$$

where γ is the Euler constant and Γ represents the (upper) incomplete gamma function.

We already know the expected values of the sequential choice high information processes for both distributions. Figure 5-7 shows the differences in expected values between the simultaneous and sequential choice cases. Note that the difference can be an order of magnitude higher in this case than it was between the high and low information variants with known p (Figure 5-4), revealing that the difference in expected value changes much

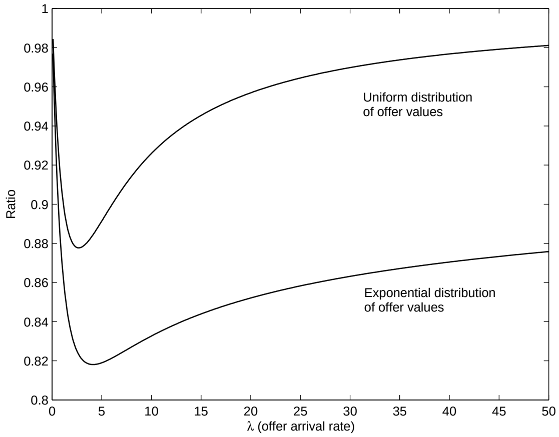


Figure 5-7: Ratio of expected values of the simultaneous choice mechanism and the sequential choice mechanisms with high information as a function of λ for the continuous time processes.

more dramatically when going from one mechanism to another than it does when going from the higher to lower information variant of the sequential choice process. However, the difference can be of the same order of magnitude when going from high to low information in the case where p is unknown. Also note that the shape of the graph is very similar to Figure 5-4, and the greatest differences are achieved for similar values of λ.

5.6.1 Some More Search Processes

These results bring up some more questions, which we will pose and answer for the uniform distribution in order to illustrate the differences between the mechanisms we have discussed and some other possible variants. Therefore, results in this section are confined to cases where offer values are generated from a uniform $[0, 1]$ distribution.

The first question that arises is how the expected values of the processes we are considering compare to the expected values in a comparable non-probabilistic case, in which the total number of appearances is fixed and the decision-maker knows this number? Gilbert and Mosteller discuss the latter case and present a recurrence relation that is also easily derived by setting $p = 1$ in equation 5.1:

$$H_{n+1} = \frac{1}{2}(1 + H_n^2)$$

Figure 5-8 shows the ratios of expected values in three different processes. The first is the high information continuous time process with arrival rate λ. In the other two cases, let us postulate the existence of a Gamesmaster, who first stores all the offers generated according to the Poisson process, and then informs the decision-maker of the total number of offers that appeared. The Gamesmaster then presents the offers to the decision-maker, either sequentially or simultaneously. Obviously, the expected value of the simultaneous process is highest, since it is the best decision that the job-seeker can make *retrospectively* (or if she were omniscient with respect to what offers she would receive). The expected value of the sequential process with a known number of offers is also bound to be significantly higher since it eliminates uncertainty about the exact number of offers the decision-maker will receive. Figure 5-8 shows the ratios of expected values of these three processes. The continuous-time process has a substantially lower expected value than the sequential process with a known number of offers for values of λ below 10, but approaches it much more rapidly than either of the sequential mechanisms approaches the simultaneous mechanism in terms of expected value. The dropoff in expected value between the continuous-time and the sequential process with known n is particularly dramatic for very small λ, indicating that knowing the exact number of offers you will receive is much more important if you only expect to receive 1-3 offers.

Figure 5-8 focuses on processes generated from an underlying process with Poisson offers arriving in continuous time, and therefore we (as the experimental designers) possess a fundamental uncertainty about the number of offers arising in each case. In contrast to this, Figure 5-9 shows the difference in expected values between two sequential processes, one with a fixed and known number of offers pn and the other one with n possible offers that each appear with probability p. While the expected value ratios are substantially smaller when pn is smaller, this is mostly because of the large probability of getting no offers. The tradeoff of possibly getting more offers is clearly not worth it in expectation, but much more so for lower values of pn. An interesting question to ask in this case is, for example, whether it is better to have one offer for sure, or 10 possible offers, each with a 20% chance of appearing (the latter, by a hair: it has expected value 0.5183, as opposed to 0.5 for the former).

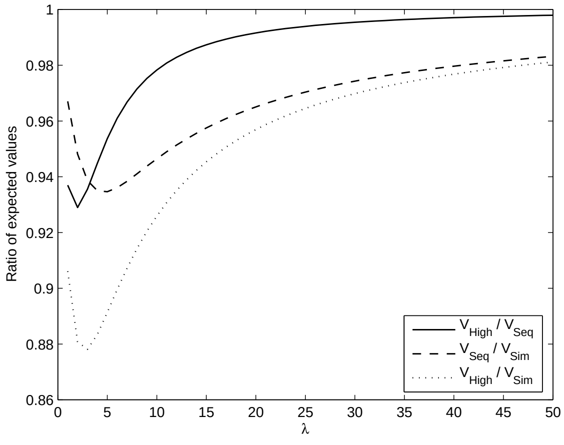


Figure 5-8: Ratios of expected values in three processes: the high information continuous-time process with Poisson arrival rate λ (denoted "High"), and two processes in which the number of offers are known beforehand after being generated by a Poisson distribution with parameter λ. The decision maker has no recall and must solve a stopping problem in the sequential choice process (denoted "Seq"), but chooses among all realized offers in the simultaneous choice process (denoted "Sim").

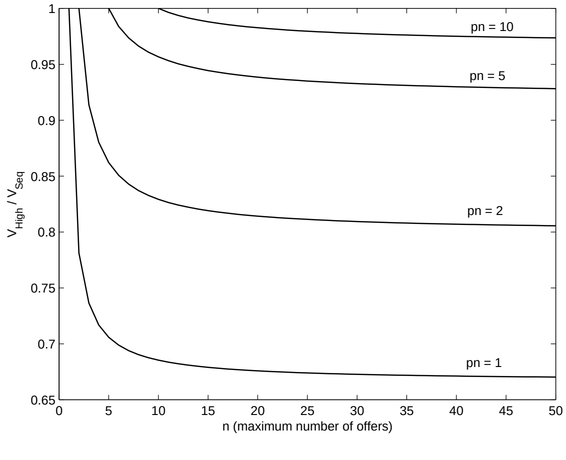


Figure 5-9: Ratio of expected values in the high information probabilistic process (denoted "High" with probability p and n total possible offers) and a process in which the number of offers is known beforehand and is equal to pn (denoted "Seq").

5.7 Conclusions

This chapter is intended to highlight the importance of the information structure in search processes, particularly processes that run over a fixed period of time, such as academic job markets. It is common practice in markets of this kind for employers or job candidates to not keep the other side fully informed about the decisions they have made. For example, universities will often not send rejections to candidates until they have completed their search, even if they were no longer seriously considering a candidate much earlier in the process. In order to study the expected loss of participating in such a process compared to a process in which both sides immediately make decisions and have to inform each other about those decisions, we have introduced a stylized model of this process that analyzes it from a one-sided perspective. Our main result is that the loss from participating in the low information process is not significant unless the decision-maker is not well-informed about her own "attractiveness," measured by the probability of receiving an offer. This suggests that the costs to changing the structure of markets that operate in the "low information" manner may not be worthwhile. If applicants are poorly informed about their own attractiveness to employers, one could imagine mechanisms to improve signaling rather than restructuring the market (of course, this assumes that employers, who participate in these processes repeatedly, can estimate their attractiveness to employees well).

The model we have introduced simplifies the problem along some dimensions. We do not incorporate two-sided strategic considerations, which may become important; for example, less attractive employers may be more inclined to make exploding offers, while more attractive employers are unlikely to do so. Further, the assumption that the probability p of receiving an offer is independent of the value of the offer may be unrealistic for some markets. Future studies should focus on these directions for extending our model.

Chapter 6

Two-Sided Bandits and Matching Markets

6.1 Introduction

This chapter analyzes the learning and decision problems of agents in matching models. We first define a class of problems in multi-agent learning and decision-making called two-sided bandit problems. This class of problems is intended to capture the essence of two-sided matching scenarios in which agents must learn their preferences through experience, rather than knowing them *a priori*. Two-sided bandit models can be applied to a wide range of markets in which two different types of agents must match with each other, including the market for romantic partnerships ("dating markets") and labor markets in which employers and employees must first learn their preferences. We present empirical and theoretical results on an example dating market modeled in this manner.

Consider a repeated game in which agents gain an uncertain payoff from being matched with a particular person on the other side of the market in each time period. A natural example of such a situation is the dating market, in which men and women repeatedly go out on dates and learn about each other. Another example is spot labor markets, in which employers and employees are matched for particular job contracts. A matching mechanism is used to pair the agents. For example, we can consider a mechanism in which all the men decide which woman to "ask out," and then each woman selects a man from her set of offers, with the rejected men left unmatched for that period.

Standard models of matching in economics (Roth and Sotomayor 1990) almost always assume that each agent knows his or her preferences over the individuals on the other side of the market. This assumption is too restrictive for many markets and the model introduced here is driven by the need to relax this assumption. The existing literature on two-sided search with nontransferable utility (Burdett and Wright 1998, e.g.) assumes matching is exogenous and random. The problem discussed here is more deeply related to bandit problems (Berry and Fristedt 1985), in which an agent must choose which arm of an n-armed bandit to pull in order to maximize long-term expected reward, taking into account the tradeoff between *exploring*, that is learning more about the reward distribution for each arm, and *exploiting*, pulling the arm with the maximum expected reward. Berry and Fristedt (1985, pp. 5) recount "this aspect prompted Whittle (1982, pp. 210) to claim that a bandit problem 'embodies in essential form a conflict evident in all human action'." Two-sided bandit problems extend the standard multi-armed bandit model by giving the arms of the bandit agency — they can decide whether to be pulled or not, or whom to be pulled by, and they themselves receive rewards based on who the puller is. This adds an enormous degree of complexity to the problem.

It is prohibitively difficult to solve for game-theoretic equilibria in all but the simplest two-sided bandit problems. In fact, even defining what would constitute optimal behavior with unbounded computational resources and a perfect model of the world is hard to do. For comparison, consider the history of the search for optimal algorithms for the traditional multi-armed bandit problem. Berry and Fristedt (1985, Chapter 1) provide a fascinating summary of the history of bandit problems, which were first introduced in 1933 by Thompson (1933). It was not until 1974 that Gittins and Jones (1974) showed the existence of an optimal strategy in a Bayesian framework for multi-armed bandits with independent arms and geometric (sometimes referred to as exponential) discounting . The difficulty of finding an optimal strategy for a significantly simpler class of problems illustrates how hard it might be to find one in two-sided bandit problems. The focus of research should be on good algorithms and what we can show about these algorithms in general settings. This approach is also related to the theory of learning in games (Fudenberg and Levine 1998), which considers more generally how individual learning rules affect outcomes in games and whether agents reach static equilibria.

6.1.1 Outline

We first give a general introduction to two-sided matching with learning and consider various different kinds of payoff and information structures and matching mechanisms that can be used. We then present a dating market modeled in this manner.

6.2 Modeling Choices: Defining Preferences, Payoffs and Matching Mechanisms

Various choices have to be made in modeling a two-sided matching game with learning. These choices relate to the structure of repetition in the game, the payoffs agents receive and when they receive them, how agent preferences over the other side of the market are defined, how much information agents have about themselves and others, what the space of actions available to agents is, and how agents are matched based on the actions they take. Let us consider each of these issues briefly.

- **The structure of the repeated game:** While the number of choices is limitless, two basic options present themselves. First, the game can be repeated either a fixed or infinite number of times, with agents seeking to maximize their total reward over the course of the game (appropriately discounted for the infinite case). In this case, payoffs are received in the same manner at each time period, agents face a dynamic programming problem and have to make the exploration-exploitation tradeoff at all time periods, so decision-making can be treated similarly at each tick. This is a natural model of markets with repeated long horizon learning interactions, like women and men going on dates. It can be made more realistic by incorporating switching costs or some other notion of commitment.

  Another natural model would be one where agents on both sides of the market have a small amount of time to learn about each other and then must settle down. The cost of exploration in the initial learning phase is small, but the cost of a bad match in the second phase, where agents are expected to settle down, can be very high. This is an appropriate model for some labor markets like the academic job market for junior faculty, with job applicants on one side of the market and employers on the other side.

- **Defining payoffs and preferences:** For simplicity, I will consider a few important types of preferences in a situation where all agents on both sides of the market have a type that is a real number between 0 and 1. The payoff an agent receives (either at each time period or at the second phase matching) is a function of the agent's own type and the type of the agent it is matched with. Suppose the two sides of the market contain agents denoted by E_i and A_i. Let the type of agent X_i be V_{X_i} where X is either E or A. Some possibilities for the preference structure are:

  1. **Completely symmetric heterogeneous preferences**: The utility received by E_i and A_j when they are matched is the same. A simple example would be for the agents both to receive a payoff that is a function of $(V_{E_i} - V_{A_j})^2$. This can model a case where agents prefer to be with those who are similar to themselves.
  2. **Completely homogeneous preferences**: An agent receives a utility that is a function of the type of the agent it is matched with. Consider a matching between E_i and A_j. In this case, the utility to E_i would be a function of V_{A_j} and the utility to A_j would be a function of V_{E_i}. Then V_{X_k} is a measure of intrinsic quality and the value someone gets from being matched with an agent is just a function of that intrinsic quality.
  3. **Mixed preferences**: An agent receives a utility that is a mixture of its similarity to the agent it is matched with and the intrinsic quality of that agent. In this case, the utility to E_i of being matched with A_j could be of the form $\kappa_1 f_1(V_{A_j}) + \kappa_2 f_2((V_{E_i} - V_{A_j})^2)$.
  4. **Discontinuous preferences:** As a motivating example, suppose that the ideal match for an employer is an employee that is sufficiently qualified for the job, but not overqualified. Suppose 0 is the worst type and 1 is the best. An example of such preferences would be if the utility received by employer E_i for hiring employee A_j could be $f_1(E_i - A_j)$ for $A_j \leq E_i$ but $f_1(A_j - E_i) - \kappa$ for $A_j > E_i$, where $\kappa > 0$.

  Of course, the actual payoffs received at any time can be noisy.

- **Information availability:** Agents may possess extremely varying degrees of information. There may or may not be an asymmetry in terms of the information available on both sides of the market. For example, in a dating market, we would expect that men and women are both similarly (un?)informed about both their own types and the types of those on the other side. However, in the academic job search market, applicants who have all been selected for interviews at the same schools presumably know their preferences among these schools better than the schools know their preferences among those they have called for interviews.[1]

- **Matching mechanisms:** There are many different ways of actually performing the matching at each time period. For example, the National Resident Matching Program matches medical school graduates to their first hospital appointment by running a matching algorithm on submitted ranked lists. On the other hand, dating typically works on an informal setup where individuals ask each other out (the role of the matching mechanism is explored in detail for a dating market in the next section), students starting graduate school in the US have to decide by a specific date among all the offers they receive, and job offers in the financial sector are typically "exploding" take-it-or-leave-it offers.

- **What to study:** When we study learning algorithms in the two-sided bandit framework, we have to keep in mind what kinds of properties we are interested in for a particular system. There can be differences between individual incentives and stable matchings. Consider a 2x2 case with symmetric heterogeneous preferences where the types of two job applicants are 0 and 0.1 and the types of the two employers are 0 and 1. Both applicants would rather be with employer 1 in this case. Therefore, if employer 1 didn't learn the types of the employees properly, or if applicant 1 didn't learn the types of the employers properly, we could end up with an unstable matching.

  There can also be a difference between socially optimal and stable matches in this scenario. In a 2x2 case where the types of the two applicants are 0.5 and 1 and the types of the two employers are 0 and 0.5, the stable match is the one where applicant 1 is matched with employer 2. However, the other match has greater social welfare if we use a utility function of subtracting the square difference in types from 1 (the

[1] Even if they may change their preferences after seeing some of the schools — this is just a relative argument!

stable matching gives a utility of 1 to two players and 0 to the other two, whereas the unstable matching gives a utility of 0.75 to everyone).

Roth has argued persuasively that stability is the right concept to think about in terms of the social outcome (Roth and Peranson 1999, Roth and Xing 1994, *inter alia*). Therefore we would like to identify the kinds of problems that are interesting when we think about learning and stability. For example, we could study whether learning algorithms convergence to stable matchings and how long such convergence takes in different cases.

6.3 Dating Markets as Two-Sided Bandits

6.3.1 Overview

This section models a dating markets as a two-sided bandit problem and describes three important matching mechanisms. We define regret as the difference between actual reward received and the reward under the stable matching, i.e. a matching such that there is no pair that would rather be with each other than with whom they are matched. We experimentally analyze the asymptotic stability and regret properties when agents use ϵ-greedy learning algorithms adapted to the different matching mechanisms.

The Gale-Shapley mechanism (Gale and Shapley 1962) yields stability when information is complete and preferences are truthfully revealed and converges quickly to stable matchings, whereas mechanisms that are more realistic for the dating example, in which men make single offers to the women, do not always converge to stable matchings. Asymptotically stable matches are more likely when agents explore more early on. They are also more likely when agents are optimistic (again, early on) — that is, they assume a higher probability of their offer being accepted or an offer being made to them than is justified by the past empirical frequencies of these events. However, increased optimism does not interact well with increased exploration in terms of stability, and the probability of stability is actually higher for lower exploration probabilities when optimism is greater.

6.3.2 The Model

There are M men and W women, who interact for T time periods. v^m_{ij} is the value of woman j to man i, and v^w_{ij} is the value of man j to woman i. These values are constant through time. In each period, men and women are matched to each other through a *matching mechanism*. A matching is a pairing between men and women in which each woman is paired with one or zero men and each man is paired with one or zero women. Formally, a matching mechanism is a mapping from agents' actions to a matching. If man i is matched with woman j in period t, he receives $v^m_{ij} + \epsilon^m_{ijt}$, and she receives $v^w_{ji} + \epsilon^w_{jit}$. If unmatched, individual i receives some constant value K_i.

For our empirical analysis we put some structure on the reward processes and the matching mechanism. First, we make the strong assumption of sex-wide homogeneity of preferences. That is, every man is equally "good" for each woman and vice versa — there are no idiosyncratic preferences and there are no couples who "get along" better than others. Formally, $v^m_{ij} = v^m_j \; \forall i$ and $v^w_{ij} = v^w_j \forall i$. We also assume that $\forall i, K_i = K$ with $K \ll \min_j v^z_{ij} \forall i, z \in \{m, w\}$ and that the noise terms ϵ are independently and identically distributed. Extensions to more general preferences are straightforward. Note that there is always a unique stable matching under this preference structure. With multiple stable matches, we would need to use a different notion of regret, as discussed later.

We consider three matching mechanisms. Without loss of generality, we assume that women always ask men out.

Gale-Shapley matching Each agent submits a list of preferences and a centralized matching procedure produces a matching based on these lists. The Gale-Shapley algorithm (Gale and Shapley 1962) guarantees a matching that is stable under the submitted preferences. The man-optimal variant yields the stable matching that is optimal for the men, and the woman-optimal variant the stable matching that is optimal for the women. We use the woman-optimal variant.

Simultaneous offers Each woman independently chooses one man to make an offer to. Each man selects one of the offers she receives. Women who are rejected are unmatched for the period, as are the men who receive no offers.

Sequential offers Each woman independently chooses one man to make an offer to. The offers are randomly ordered and the men must decide on these "exploding" offers without knowing what other offers are coming. If an offer is rejected the woman making the offer is unmatched in that period. A man is unmatched if he rejects all offers he receives.

Intuitively, it is useful to think of the simultaneous choice mechanism as capturing a situation in which women ask men out over e-mail and each man can review all his offers before making a decision, while the sequential choice mechanism captures the situation where women ask men out over the telephone. We are particularly interested in these two matching mechanisms because they are more plausible descriptions of reality than a centralized matchmaker, and do not require agents to reveal their preferences to a third party.

6.3.3 The Decision and Learning Problems

We initially describe the decision problems men and women face at each time step if they want to optimize their myopic reward in that time step. After this we discuss the exploration-exploitation issues men and women face under the different matching mechanisms and describe specific forms of the ϵ-greedy algorithm.

Let $Q_{ij}^{\{m,w\}}$ denote man (woman) i's estimate of the value of going out with woman (man) j, p_{ij}^{w} denote woman i's estimate of the probability that man j will go out with her if she asks him out and p_{ij}^{m} denote man i's estimate of the probability that woman j will ask him out under the sequential choice mechanism.

The Woman's Decision Problem

Under Gale-Shapley matching, the woman's action space is the set of rankings of men. Under the other two mechanisms, the woman chooses which man to make an offer to. She must base her decision on any prior beliefs and the history of rewards she has received in the past. She has to take into account both the expected value of going on a date with each man and (for the non Gale-Shapley mechanisms) the probability that he will accept her offer.

Under the woman-optimal variant of the Gale-Shapley mechanism the dominant myopic strategy, and thus the greedy action, is for woman i to rank the men according to the expected value of going out with each of them, Q^w_{ij}. For the other two mechanisms, the greedy action is to ask out man $j = \arg\max_j(p^w_{ij}Q^w_{ij})$.

Arms With Agency: The Woman's Decision Problem

The action space of men, the arms of the bandit, may be constrained by the women's actions. The decision problem faced by a man depends on the matching mechanism used. Under the woman-optimal Gale-Shapley mechanism, men may have an incentive to misrepresent their preferences, but since the sex-wide homogeneity of preferences ensures a unique stable matching (Roth and Sotomayor 1990), this is less likely to be a problem.[2] So, the greedy action for man i under Gale-Shapley is to rank women based on their Q_{ij}'s.

With the simultaneous choice mechanism, in each time period a man receives a list of women who have made her an offer. He must decide which one to accept. This is a bandit problem with a different subset of the arms available at each time period. The greedy action is to accept the woman $j = \arg\max_j Q^m_{ij}$

Under the sequential choice mechanism, a man might receive multiple offers within a time period, and each time he receives an offer he has to decide immediately whether to accept or reject it, and he may not renege on an accepted offer. The information set he has at that time is only the list of women who have asked him out so far. For each woman who has not asked him out, it could either be that she chose not to make him an offer, or that her turn in the ordering has not arrived yet. We can formulate the man's value function heuristically. Let i be the index of the man, let S be the set of women who have asked him out so far, and let h be the woman currently asking her out ($h \in S$).

$$V(S,h) = \max\{Q^m_{ih}, \sum_{k \notin S} \Pr(k \text{ next woman to ask } i \text{ out}) V(S \cup \{k\}, k)\}$$

The base cases are $V(X,h) = Q^w_{ih}$ where X is the set of all women. The greedy action is to accept an offer when

$$Q^m_{ih} > \sum_{k \notin S} \Pr(k \text{ next woman to ask } i \text{ out}) V(S \cup \{k\}, k)$$

[2] If the *submitted* rankings satisfy sex-wide homogeneity man- and woman-optimal algorithms yield the same matching and truthtelling is the dominant myopic strategy for men.

The relevant probabilities are:

$$\Pr(k \text{ next woman to ask } i \text{ out}) = \sum_{T \in \text{Perm}(S')} \left[\frac{1}{|S'|} \left(\prod_{j \text{ preceding } k \text{ in } T} (1 - p_{ij}^{m}) \right) p_{ik}^{m} \right]$$

where $S' = X \setminus S$. Solving the dynamic programming problem takes exponential time, which could lead to interesting issues as to how to best approximate the value function and the effect the approximation might have on market outcomes when there are large numbers of agents.

The Exploration-Exploitation Tradeoff

Women and men both have to consider the exploration-exploitation tradeoff (summarized by Sutton and Barto (1998)). *Exploitation* means maximizing expected reward in the current period (also called the greedy choice), and is solved as above. *Exploration* happens when an agent does not select the greedy action, but instead selects an action that has lower expected value in the current period in order to learn more and increase future rewards.

The one-sided version of the exploration-exploitation problem is central to n-armed bandit problems (Berry and Fristedt 1985, Gittins and Jones 1974)[*inter alia*]. An n-armed bandit is defined by random variables $X_{i,t}$ where $1 \leq i \leq n$ is the index of the arm of the bandit, and $X_{i,t}$ specifies the payoff received from pulling arm i at time t. The distribution of some or all of the $X_{i,t}$ is unknown so there is value to exploring. The agent pulls the arms sequentially and wishes to maximize the discounted sum of payoffs. In our model, if there is a single woman and n men, the woman faces a standard n-armed bandit problem.

One of the simplest techniques used for bandit problems is the so-called ϵ-greedy algorithm. This algorithm selects the arm with highest expected value with probability $1 - \epsilon$ and otherwise selects a random arm. Although simple, the algorithm is very successful in most empirical problems, and we therefore use it in our experiments. We have also experimented with alternatives like softmax-action selection with Boltzmann distributions (Sutton and Barto 1998, Luce 1959) and the Exp3 algorithm (Auer et al. 2002). These do not improve upon the empirical performance of ϵ-greedy in our simulations.

Under each matching mechanism the exploratory action is to randomly select an action, other than the greedy one, from the available action space. Since the value of exploration decreases as learning progresses, we let ϵ decay exponentially over time which also ensures that the matchings converge.

At this stage we cannot solve for the perfect Bayesian equilibrium set. We believe, however, that if the agents are sufficiently patient and the horizon is sufficiently long, the matchings will converge to stability on the equilibrium path. Solving for the equilibrium set would enable us to explicitly characterize the differences between the payoffs on the equilibrium path and the payoffs under the ϵ-greedy algorithm.

The two-sided nature of the learning problem leads to nonstationarities. Under the sequential and simultaneous mechanisms the women need to consider the reward of *asking out* a particular man, not the reward of going out with him. The reward of asking out a particular man depends on the probability that he will accept the offer. Thus, the reward distribution changes based on what the men are learning, introducing an externality to the search process. The same applies to men under the sequential mechanism since the probability that a particular woman will ask a man out changes over time. There is a problem of coordinated learning here that is related to the literature on learning in games (Fudenberg and Levine 1998) as well as to reinforcement learning of nonstationary distributions in multiagent environments (Bowling and Veloso 2002). Some recent work by Auer et al. (2002) on "adversarial" bandit problems, which makes no distributional assumptions in deriving regret bounds is relevant in this context.

Since the underlying v_{ij}'s are constant we define Q_{ij} as person i's sample mean of the payoff of *going out* with person j. In order to deal with the nonstationarity of p_{ij}'s, on the other hand, we use a fixed learning rate for updating the probabilities which allows agents to forget the past more quickly:

$$p_{ij}[t] = (1 - \eta)p_{ij}[t-1] + \eta I[\text{offer made / accepted}]$$

where η is a constant and I is an indicator function indicating whether a man accepted an offer (for the woman's update, applied only if woman i made an offer to man j at time t) or whether a woman made an offer to a man (for the man's update, applied at each time period t).

ϵ	Simultaneous Choice		Sequential Choice		Gale-Shapley	
	Pr (stability)	Score	Pr (stability)	Score	Pr (stability)	Score
.1	0.318	0.4296	0.050	0.9688	1.000	0.0000
.2	0.444	0.3832	0.054	0.9280	1.000	0.0000
.3	0.548	0.2920	0.050	0.8560	1.000	0.0000
.4	0.658	0.1880	0.058	0.8080	1.000	0.0000
.5	0.788	0.0992	0.096	0.7448	1.000	0.0000
.6	0.856	0.0672	0.108	0.7064	1.000	0.0000
.7	0.930	0.0296	0.130	0.6640	1.000	0.0000
.8	0.970	0.0120	0.164	0.5848	1.000	0.0000
.9	0.998	0.0008	0.224	0.4912	1.000	0.0000

Table 6.1: Convergence to stability as a function of ϵ

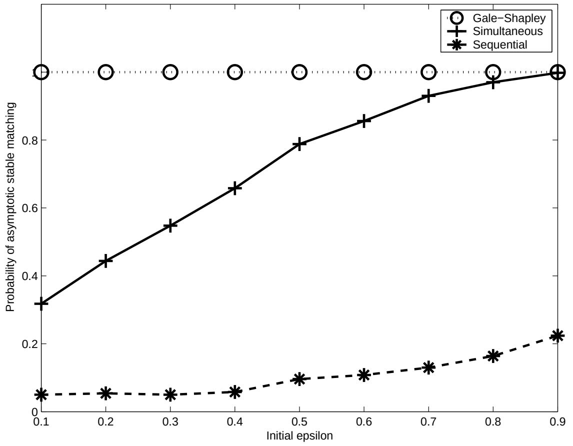


Figure 6-1: Probability of a stable (asymptotic) matching as a function of the initial value of ϵ

6.3.4 Empirical Results

Our simulations involve a market with 5 women and 5 men. The agents use η of 0.05 for updating their probability estimates and the probability of exploration evolves with time as $\epsilon_t = \epsilon^{t/1000}$. Agents have true values $v_0^m = v_0^w = 10, v_1^m = v_1^w = 9, v_2^m = v_2^w = 8, v_3^m = v_3^w = 7, v_4^m = v_4^w = 6$. The noise signals $\epsilon_{ijt}^{\{m,w\}}$ are i.i.d. and drawn from a normal distribution. Unless otherwise specified, the standard deviation of the noise distribution is 0.5. Reported results are averages from 500 simulations, each lasting a total of 30,000 time steps. Initial values of Q_{ij} are sampled from a uniform $[6, 10]$ distribution and initial values of p_{ij} are sampled from a uniform $[0, 1]$ distribution.

Our experiments show that settings in which agents are matched using the Gale-Shapley mechanism always result in asymptotically stable matchings, even for very small initial values of ϵ such as 0.1. After a period of exploration, where the agents match up with many different partners and learn their preferences, agents start pairing up regularly with just one partner, and this is always the agent with the same ranking on the other side. Interestingly, even if only one side explores (that is, either men or women always pick the greedy action), populations almost always converge to stable matchings, with a slight decline in the probability of stability when only men explore (under the woman-optimal matching algorithm women's rankings can have a greater effect on the matching than men's rankings).

The probabilities of convergence under the simultaneous and sequential choice mechanisms are significantly lower, although they increase with larger initial values of ϵ. We can see this behavior in Figure 6-1, which also reveals that the probability of convergence to a stable matching is much higher with the simultaneous choice mechanism. Table 6.1 shows these probabilities as well as the score, which is a measure of how large the deviation from the stable matching is. If men and women are indexed in order of their true value ranking, the score for a matching is defined as $\frac{1}{M}\sum_{i \in X} |i - \text{Partner}(i)|$ where $\text{Partner}(i)$ is the true value ranking of the man woman i is matched with, and X is the set of all women.

It is also interesting to look at who benefits from the instabilities. In order to do this, we define a notion of regret for an agent as the (per unit time) difference between the reward under the stable matching and the actual reward received (a negative value of regret indicates that the agent did better than (s)he would have done under the stable matching). This definition is unambiguous with sex-wide homogeneity of preferences since there is only one stable matching, but could be problematic in other contexts, when there could be more than one. In this case it might make sense to analyze individual agent performance in terms of the difference between average achieved reward and expected reward under one of the stable matchings depending on context.

In the case of sex-wide homogeneity of preferences we of course expect that regret will be greater for more desirable agents since they have more to lose when their value is not known. Table 6.2 shows the distribution of regrets for simultaneous and sequential choice. The regrets are averaged over the last 10,000 periods of the simulation. Under simultaneous choice, the worst woman benefits at the expense of all other women while the worst two men benefit at the expense of the top three. Under sequential choice, other agents benefit at the expense of the best ones.

ID	Simultaneous Regret		Sequential Regret	
	Woman's	Man's	Woman's	Man's
0	0.126	0.126	0.578	0.552
1	0.090	0.278	-0.023	0.009
2	0.236	0.136	-0.153	-0.148
3	0.238	-0.126	-0.005	-0.024
4	-0.690	-0.414	-0.171	-0.187

Table 6.2: Distribution of regret under simultaneous choice ($\epsilon = 0.1$) and sequential choice ($\epsilon = 0.9$) mechanisms

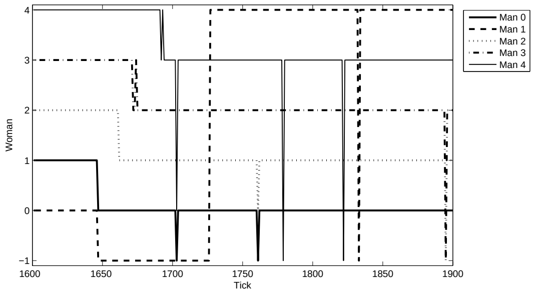


Figure 6-2: A "phase transition": men and women are ranked from 0 (highest) to 4 (lowest) with -1 representing the unmatched state. The graph shows the transition to a situation where the second highest ranked man ends up paired with the lowest ranked woman

σ	Pr (stability)	Score
0.5	0.658	0.1880
1.0	0.636	0.1952
1.5	0.624	0.2120
2.0	0.600	0.2328

Table 6.3: Convergence to stability as a function of σ with simultaneous choice and initial $\epsilon = 0.4$

Figure 6-2 shows interesting dynamic behavior in one particular simulation in which the second best man ends up paired with the worst woman. The graph shows which man is matched with which woman at each time period. The lines represent the men, and the numbers on the Y axis represent the women. The value -1 represents the state of being unmatched in that period for a man. The men and women are ranked from 0 (best) to 4 (worst). Initially, the second best man is paired with the best woman so he keeps rejecting offers from all the other women. These women thus learn that he is extremely particular about who he dates and there is no point in asking him out. When the best woman finally learns that she can get a better man this triggers a chain of events in which all the men sequentially move to the woman ranked one higher than the one they were seeing. However, all the women have such a low opinion of the second best man that he ends up getting matched with the very worst woman. The matching shown at the end of the graph is the final asymptotic matching in this simulation. An interesting point to note is that the asymmetry (only women are allowed to ask men out) precludes this from happening to a woman.

Another question to ask is how the probability of stability is affected by the noise distribution. We expect that there will be less convergence to stability when the signals are less precise. We ran experiments in which the standard deviation of the noise distribution was changed while holding other factors constant. We used an initial ϵ of 0.4 and the same underlying values as above. Table 6.3 shows the results using the simultaneous choice mechanism. We vary the standard deviation from one half of the distance between the two adjacent true values (0.5) to twice that distance (2.0), and the probability of stability falls by less than 10%. This suggests that the instabilities arise mostly from the structure of the problem and the nonstationarity of probability estimates rather than from the noise in the signals of value.

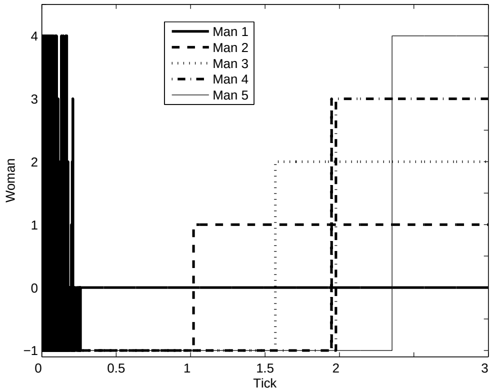


Figure 6-3: The mechanism of stability with optimism: agents keep trying better ranked agents on the other side until they finally "fall" to their own level

6.3.5 Optimism and Exploration

The insight that instabilities arise mostly from the structure of the problem and that agents are generally successful at learning their preferences, suggests an alternative method for engineering asymptotic stability into the system. Suppose agents are initially optimistic and their level of optimism declines over time. This is another form of patience — a willingness to wait for the best — and it should lead to more stable outcomes.

Optimism can be represented by a systematic overestimation of the probability that your offer will be accepted or that an offer will be made to you. We explore this empirically with the sequential choice mechanism. Instead of using the learned values of p_{ij} as previously defined, agents instead use an optimistic version. At time t, both men and women use the optimistic probability estimate:

$$p'_{ij} = \alpha_t + (1 - \alpha_t)p_{ij}$$

in decision making (the actual p_{ij} is, however, maintained and updated as before). α_t should decline with time. In our simulations $\alpha_0 = 1, \alpha_T = 0$ (where T is the length of the simulation) and α declines linearly with t. There are no other changes to any of the decision-making or learning procedures.

Figure 6-3 shows the process by which agents converge to asymptotic matchings (in this case a stable one) with the optimistic estimates. The structure of the graph is the

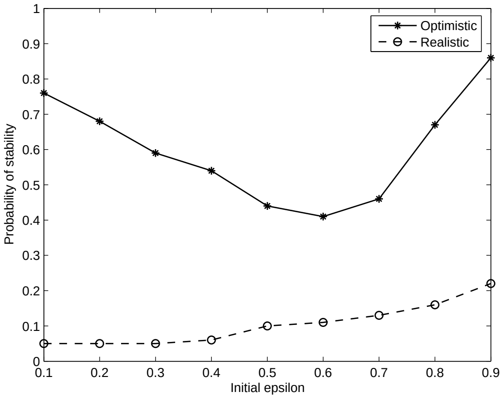


Figure 6-4: Probability of convergence to stability for different initial values of epsilon with all agents using the optimistic algorithm versus all agents using the realistic algorithm

same as that in Figure 6-2. Essentially, each agent keeps trying for the best agent it can match with until the optimism parameter has declined sufficiently that it "falls" to the equivalently ranked agent on the other side of the market. Figure 6-4 shows that agents are considerably more likely to converge asymptotically to stable matchings using this algorithm for any value of the initial exploration probability. Of course, this convergence comes at the expense of utility achieved in the period before the agents settle down.

The surprising feature in Figure 6-4 is that stable matchings are more likely with smaller initial exploration probabilities. The V-shape of the graph shows that the probability of stability actually declines with increasing exploration up to an initial ϵ value of 0.6, before starting to increase again, in contrast to the monotonically increasing probability of stability without optimism. This can be explained in terms of the fact that a small level of exploration is sufficient for agents to learn their preferences. Beyond that, waiting for the best that an agent can achieve is taken care of by the optimism rule, and additional exploration is not only not useful, it becomes counterproductive because the probability estimates at the key stages become less reliable.

6.3.6 Summary

We have defined two-sided bandit problems, a new class of problems in multi-agent learning and described the properties of three important matching mechanisms with ϵ-greedy learning rules. Two-sided bandit problems are of great relevance for social science in general and the search for marriage partners in particular. The social norms governing exploration before marriage have been changing rapidly over the last few decades and until now we have had no formal structure within which to study the sources and consequences of these changes. Our model is also more generally applicable to two-sided markets in which agents have to learn about each other.

This chapter only scratches the surface of a large and potentially fruitful set of theoretical and empirical questions. It is important to explore learning algorithms that would allow agents to perform well[3] across a broad range of environments without having to make assumptions about the decision-making algorithms or learning processes of other agents. Another direction of research is to explicitly characterize equilibria in simpler settings. We are also interested in more complex versions of the problem that allow for a greater diversity of preferences and a larger number of agents. Finally, the insights gained from the one-sided problem discussed in the previous chapter should be examined more formally within the context of a two-sided model with a separate preceding search-and-matching period.

[3] In the sense of regret minimization.

Chapter 7

Conclusion

7.1 A Framework for Modeling and its Successes

The main argument of this thesis is that we need to build richer models of economic and social systems, and we can do this in a principled way by solving engineering problems of designing good algorithms, and translating success in that task into scientific success in analyzing systems using computational techniques. The main technical contributions are in specific domains that can benefit from this approach, namely market microstructure and search. Within market microstructure I have shown how explicitly modeling the learning processes of agents can yield valuable insights into both price properties in markets and the likelihood that real agents faced with informational and computational constraints will eventually learn classical equilibrium behavior. In search models, the main contribution of this thesis is to show how we can learn more about the properties of different market mechanisms by explicitly modeling agents that solve complex learning and decision problems that are rarely analytically tractable. Richer models can yield new insights into familiar settings, and they can also allow us to ask new *types* of questions about systems, like questions about non-equilibrium behavior, system dynamics, and rates of convergence.

7.2 Future Work

While each chapter specifies directions for research within its domain, I would like to paint a broader picture in this final section. This thesis should serve as a starting point for future work in different fields and application domains. First, of course, I hope it stimulates further computational modeling of complex systems in which agents attempt to be as close to optimal or rational as possible. While there has been plenty of research on modeling complex systems with many different types of agents, I think there is a renewed need for the discipline of optimization (or at least something close to it) in many of these models. Instead of modeling participants as largely random, we should explicitly incorporate their uncertainties, their learning processes, and the computational and other resource constraints imposed on them within the framework of far more complex models than those of traditional economics and finance theory. This approach will become important not just for analyzing existing economic and social structures, but even more so for analyzing more and more electronic marketplaces as they become increasingly important in the coming years. Participants in these markets will often be artificial agents endowed with the best algorithms their creators can think of, and we need to have an established program of research in place for understanding the possibilities as this trend continues.

In order to pursue the first goal stated above successfully, the state of the art in thinking about and modeling bounded rationality must move forward. Progress will inevitably come from different disciplines, including artificial intelligence, cognitive science, and economics, but I think it is especially important to continue to think about algorithms in terms of whether or not they are boundedly optimal for their environments (even if this is hard or impossible to determine), and to move towards designing more general purpose agents that have to take decisions over a range of problems, not just a few they are specifically designed for. It is also critical to take cues in the design of algorithms from human behavior. Clearly we have evolved to be very well-designed for the environments we inhabit – what can this teach us about fast and efficient algorithms for problems that we eventually want artificial agents to solve?

Finally, on a different and more application-domain specific note, this thesis is largely composed of exercises in modeling. While I have mentioned that this kind of modeling will become more important as electronic societies and marketplaces become more important, the models presented here could also be useful in understanding markets and systems that exist in the real world today. The next step then is to calibrate and test these models in the real world. The third chapter takes some basic steps in this direction, and the models in the other chapters may yet be too stylized to easily be testable, but extensions should definitely be put to the test of the world so we can learn what the models get right, what they get wrong, and where we should go from here.

Bibliography

Amihud, Y., H. Mendelson. 1980. Dealership market: Market-making with inventory. *Journal of Financial Economics* **8** 31–53.

Arthur, W. Brian. 1994. Inductive reasoning and bounded rationality (the el farol problem). *The American Economic Review (Papers and Proceedings)* **84**.

Arthur, W. Brian. 1999. Complexity and the economy. *Science* **284** 107–109.

Auer, Peter, Nicolò Cesa-Bianchi, Yoav Freund, Robert E. Schapire. 2002. The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing* **32**(1) 48–77.

Berry, D.A., B. Fristedt. 1985. *Bandit Problems: Sequential Allocation of Experiments*. Monographs on Statistics and Applied Probability, Chapman and Hall, London, UK.

Bertsekas, Dimitri P., John Tsitsiklis. 1996. *Neuro-Dynamic Programming*. Athena Scientific, Belmont, MA.

Borodin, Allan, Ran El-Yaniv. 1998. *Online Computation and Competitive Analysis*. Cambridge University Press, Cambridge, UK.

Bouchaud, J.-P., Y. Gefen, M. Potters, M. Wyart. 2004. Fluctuations and response in financial markets: the subtle nature of 'random' price changes. *Quantitative Finance* **4** 176–190.

Bowling, Michael, Manuela M. Veloso. 2002. Multiagent learning using a variable learning rate. *Artificial Intelligence* **136** 215–250.

Brooks, Rodney A. 1991. Intelligence without representation. *Artificial Intelligence* **47** 139–159.

Bruss, F.Thomas. 1987. On an optimal selection problem by Cowan and Zabczyk. *Journal of Applied Probability* **24** 918–928.

Burdett, K., R. Wright. 1998. Two-sided search with nontransferable utility. *Review of Economic Dynamics* **1** 220–245.

Carlson, J.M., John Doyle. 2002. Complexity and robustness. *Proceedings of the National Academy of Sciences* **99, Suppl.1** 2538–2545.

Chazelle, Bernard. 2006. URL `http://www.supercomputingonline.com/article.php?sid=10496`. Interview preceding address to the American Association for the Advancement of Science.

Conitzer, Vincent, Tuomas Sandholm. 2003. Awesome: A general multiagent learning algorithm that converges in self-play and learns a best response against stationary opponents. *Proceedings of the 20th International Conference on Machine Learning*. 83–90.

Conlisk, John. 1996. Why bounded rationality? *Journal of Economic Literature* **34**(2) 669–700.

Cowan, A. J. Zabczyk. 1978. An optimal selection problem associated with the Poisson process. *Theory of Probability and its Applications* **23** 584–592.

Darley, V., A. Outkin, T. Plate, F. Gao. 2000. Sixteenths or pennies? Observations from a simulation of the NASDAQ stock market. *IEEE/IAFE/INFORMS Conference on Computational Intelligence for Financial Engineering*.

Darley, Vince. 1999. Towards a theory of autonomous, optimizing agents. Ph.D. thesis, Harvard University.

Das, Sanmay, Emir Kamenica. 2005. Two-sided bandits and the dating market. *Proc. IJCAI 2005*. Edinburgh, UK, 947–952.

DeGroot, Morris H. 1970. *Optimal Statistical Decisions*. McGraw-Hill, New York.

Etzioni, Oren. 1993. Intelligence without robots (A reply to Brooks). *Artificial Intelligence Magazine* **14**(4).

Evans, George W., Seppo Honkapohja. 2005. An interview with Thomas J. Sargent. *Macroeconomic Dynamics* **9** 561–583.

Farmer, J.D., F. Lillo. 2004. On the origin of power-law tails in price fluctuations. *Quantitative Finance* **4** C7–C11.

Ferguson, Thomas S. 1989. Who solved the secretary problem? *Statistical Science* **4**(3) 282–289.

Foster, F.D., S. Viswanathan. 1996. Strategic trading when agents forecast the forecasts of others. *The Journal of Finance* **51** 1437–1478.

Freeman, Jason A.S., David Saad. 1997. Online learning in radial basis function networks. *Neural Computation* **9** 1601–1622.

Fudenberg, Drew, David K. Levine. 1998. *The Theory of Learning in Games*. MIT Press, Cambridge, MA.

Gabaix, X., P. Gopikrishnan, V. Plerou, H.E. Stanley. 2003. A theory of power law behavior in financial market fluctuations. *Nature* **423** 267–270.

Gale, D., L.S. Shapley. 1962. College admissions and the stability of marriage. *The American Mathematical Monthly* **69**(1) 9–15.

Garman, M. 1976. Market microstructure. *Journal of Financial Economics* **3** 257–275.

Gigerenzer, Gerd, Daniel G. Goldstein. 1996. Reasoning the fast and frugal way: Models of bounded rationality. *Psychological Review* **103**(4) 650–669.

Gigerenzer, Gerd, Reinhard Selten. 2001. Rethinking rationality. Gerd Gigerenzer, Reinhard Selten, eds., *Bounded Rationality: The Adaptive Toolbox*. The MIT Press, Cambridge, MA.

Gilbert, John, Frederick Mosteller. 1966. Recognizing the maximum of a sequence. *Journal of the American Statistical Association* **61** 35–73.

Gittins, J.C., D.M. Jones. 1974. A dynamic allocation index for the sequential design of experiments. J. Gani, K. Sakadi, I. Vinczo, eds., *Progress in Statistics*. North Holland, Amsterdam, 241–266.

Glimcher, Paul W. 2003. *Decisions, Uncertainty, and the Brain*. MIT Press, Cambridge, MA.

Glosten, L.R., P.R. Milgrom. 1985. Bid, ask and transaction prices in a specialist market with heterogeneously informed traders. *Journal of Financial Economics* **14** 71–100.

Goldstein, Daniel G., Gerd Gigerenzer. 2002. Models of ecological rationality: The recognition heuristic. *Psychological Review* **109**(1) 75–90.

Grossman, S.J., M.H. Miller. 1988. Liquidity and market structure. *Journal of Finance* **43** 617–633.

Hogan, Jenny. 2005. Why it is hard to share the wealth. *New Scientist* .

Holden, C.W., A. Subrahmanyam. 1992. Long-lived private information and imperfect competition. *The Journal of Finance* **47** 247–270.

Horvitz, E.J. 1987. Reasoning about beliefs and actions under computational resource constraints. *Proceedings of the 3rd AAAI Workshop on Uncertainty in Artificial Intelligence*. 429–444.

Hu, Junling, Michael P. Wellman. 1998. Online learning about other agents in a dynamic multiagent system. *Proceedings of the Second International Conference on Autonomous Agents*. 239–246.

Johnson, Neil F., Paul Jefferies, Pak Ming Hui. 2003. *Financial Market Complexity: What physics can tell us about market behavior*. Oxford University Press, Oxford, UK.

Kaelbling, L.P., M.L. Littman, A.W. Moore. 1996. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research* **4** 237–285.

Kakade, Sham M. 2003. On the sample complexity of reinforcement learning. Ph.D. thesis, Gatsby Computational Neuroscience Unit, University College London.

Kurushima, Aiko, Katsunori Ano. 2003. A note on the full-information poisson arrival selection problem. *Journal of Applied Probability* **40** 1147–1154.

Kyle, Albert S. 1985. Continuous auctions and insider trading. *Econometrica* **53**(6) 1315–1336.

Liu, Y., P. Gopikrishnan, P. Cizeau, M. Meyer, C. Peng, H.E. Stanley. 1999. Statistical properties of the volatility of price fluctuations. *Physical Review E* **60**(2) 1390–1400.

Luce, D. 1959. *Invidual Choice Behavior*. Wiley, New York.

Madhavan, A. 2000. Market microstructure: A survey. *Journal of Financial Markets* 205–258.

Mantegna, Rosario N., H. Eugene Stanley. 2000. *An Introduction to Econophysics: Correlations and Complexity in Finance*. Cambridge University Press, Cambridge, UK.

Moody, John, Matthew Saffell. 2001. Learning to trade via direct reinforcement. *IEEE Transactions on Neural Networks* **12**(4) 875–889.

Mortensen, Dale T., Christopher A. Pissarides. 1999. New developments in models of search in the labor market. *Handbook of labor economics*, vol. 3B. Elsevier Science, North-Holland, Amsterdam, 2567–2627.

O'Hara, M. 1995. *Market Microstructure Theory*. Blackwell, Malden, MA.

Parkes, David. 1999. Bounded rationality. Technical report, University of Pennsylvania, Dept of Computer and Information Sciences.

Plerou, V., P. Gopikrishnan, X. Gabaix, H.E. Stanley. 2004. On the origin of power-law tails in price fluctuations. *Quantitative Finance* **4** C11–C15.

Raberto, M., S. Cincotti, S.M. Focardi, M. Marchesi. 2001. Agent-based simulation of a financial market. *Physica A* **299** 319–327.

Roth, A.E., Elliott Peranson. 1999. The redesign of the matching market for American physicians: Some engineering aspects of economic design. *American Economic Review* **89**(4) 748–780.

Roth, Alvin E., Marilda Sotomayor. 1990. *Two-Sided Matching: A Study in Game-Theoretic Modeling and Analysis*. Econometric Society Monograph Series, Cambridge University Press, Cambridge, UK.

Roth, Alvin E., Xiaolin Xing. 1994. Jumping the gun: Imperfections and institutions related to the timing of market transactions. *The American Economic Review* **84**(4) 992–1044.

Russell, Stuart. 1997. Rationality and intelligence. *Artificial Intelligence* **94** 57–77.

Russell, Stuart, Peter Norvig. 2003. *Artificial Intelligence: A Modern Approach*. 2nd ed. Prentice Hall, Upper Saddle River, New Jersey.

Schwartz, Robert A. 1991. *Reshaping the Equity Markets: A Guide for the 1990s*. Harper Business, New York, NY.

Simon, Herbert A. 1955. A behavioral model of rational choice. *The Quarterly Journal of Economics* **69** 99–118.

Stewart, T.J. 1981. The secretary problem with an unknown number of options. *Operations Research* **29**(1) 130–145.

Stoll, H. 1978. The supply of dealer services in securities markets. *Journal of Finance* **33** 1133–1151.

Stone, Peter, Manuela M. Veloso. 2000. Multiagent systems: A survey from a machine learning perspective. *Autonomous Robots* **8**(3) 345–383.

Sutton, R.S., A.G. Barto. 1998. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA.

Tesauro, Gerald. 1995. Temporal difference learning and td-gammon. *Communications of the ACM* **38**(3) 58–68.

Thompson, W.R. 1933. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika* **25** 285–294.

Whittle, P. 1982. *Optimization Over Time: Dynamic Programming and Stochastic Control*, vol. 1. Wiley, New York.

Widrow, B., M.E. Hoff. 1960. Adaptive switching circuits. *Institute of Radio Engineers, Western Electronic Show and Convention, Convention Record, Part 4*. 96–104.