Human-inspired algorithms for search
A framework for human-machine multi-armed bandit problems

Paul Benjamin Reverdy

A Dissertation
Presented to the Faculty
of Princeton University
in Candidacy for the Degree
of Doctor of Philosophy

Recommended for Acceptance
by the Department of
Mechanical and Aerospace Engineering
Advisers: Naomi Ehrich Leonard and Philip Holmes

September 2014

Abstract

Search is a ubiquitous human activity. It is a rational response to the uncertainty inherent in the tasks we seek to accomplish in our daily lives, from retrieving information to making important decisions. Engineers have developed numerous tools to perform automated search, but many tasks have too much uncertainty for these tools to perform adequately without human intervention. Engineering solutions to such tasks therefore consist of human-machine hybrid systems, where human supervisors interact with automated tools and make high-level decisions to guide them. Novel rigorous models of human decision making in such situations are required to facilitate the principled design of human-machine systems.

In this thesis, we develop a rigorous model of human decision-making behavior in search tasks. We formally model search using the *multi-armed bandit* problem from the machine learning literature, which allows us to derive bounds on optimal decision-making performance. We focus on spatial search, for which we introduce the *spatial multi-armed bandit* problem. We develop several models of human decision-making behavior in this problem by extending heuristics from the neuroscience and machine learning literatures, and prove conditions under which one model (UCL) achieves optimal performance.

We study human-subject data from a spatial multi-armed bandit problem and show that human performance in this problem falls into several categories. Some humans outperformed standard algorithms for multi-armed bandit problems, which we attribute to humans having good intuition for spatial search. We show that the UCL model can achieve performance that falls in the different categories by tuning the model parameters.

The model parameters quantify a human's intuition and make it available to a human-machine system. We develop a parameter estimator for the UCL model by relating it to the Generalized Linear Model from the statistics literature. The UCL model together with the estimator represent a plant–observer pair for human decision making which can be used for system design.

Finally, we consider a so-called "satisficing" objective as an alternative to the maximizing objective of the standard multi-armed bandit problem. We derive performance bounds in terms of this new objective and develop an algorithm that achieves optimal performance.

Acknowledgements

I have had the privilege of working with two outstanding advisors, Naomi Leonard and Phil Holmes. While I have worked more closely with Naomi, Phil has always been encouraging and has been helpful at the crucial moments in my time as a graduate student. In fact, he was indirectly responsible for my introduction to Naomi, and to Princeton. When I took the decision to apply to graduate school, I contacted a former professor of mine, Oliver O'Reilly, who suggested I write to several professors including Naomi. I did not know it at the time, but Oliver was one of Phil's students when he was teaching at Cornell. I will always be grateful for this connection.

Naomi's enthusiasm and vision have been vital in getting me where I am today. After I followed Oliver's advice to write to Naomi, she and I corresponded for a while and she gave me several suggestions of where to apply to graduate school. Her willingness to spend time advising me before I was even a student impressed me and played a major role in my decision to come to Princeton. Once here she has provided me with the resources and freedom to develop as an independent researcher. I am relatively self-reliant by nature and Naomi's advising style has given me the wide berth to go where my work has taken me. At first this was a bit frustrating, since I was still learning how to do research, but in the end has given me valuable confidence in my abilities. Throughout the process Naomi has been there to provide advice as necessary. Even when I was unsure where my research was going, Naomi would look at my most recent results, immediately see the positive in them, and suggest potential directions to go forward. Her ability to look at a project and see several steps ahead of its current state is invaluable in maintaining the vision that pushes research forward.

Two postdocs have been valued colleagues and have played a crucial role in developing the work presented in this dissertation: Bob Wilson and Vaibhav Srivastava. Bob arrived at Princeton roughly at the same time I did, and much of the connection of my work to neuroscience comes through our collaboration. Bob introduced me to the world of modeling human decision-making behavior and to many of the intricacies of Bayesian statistics. When Vaibhav arrived at Princeton several years later, I had started to make progress building models but needed help learning the tools from the machine learning literature that facilitate analyzing the models. Vaibhav was exactly the right person at the right time and we made fast progress on the analysis, proving the performance bounds which form an integral part of this dissertation.

I also wish to thank my Ph.D. committee members Jon Cohen and Paul Cuff. Jon, together with Bob Wilson, has ensured that my work has remained relevant to neuroscience.

Paul has been a valuable resource in the field of estimation theory, which plays an important role in my work.

I have had the pleasure of working with a great number of colleagues in the Leonard group: Andy Stewart, Dan Swain, Carlos Caicedo, and I shared many memorable moments working at Forrestal and discussing ideas related to robotic applications. Kendra Cofield, Stephanie Goldfarb, Darren Pais, Ioannis Poulakakis, Tian Shen, and George Young were all valued mentors when I joined the group. Brendan Andrade, Katie Fitch, and Will Scott joined later and have become valued colleagues. Brendan took up much of the heavy lifting for the Forrestal lab when I needed to pass work on, for which I am extremely grateful. Peter Landgren will benefit from his work for years to come. I am confident leaving the Forrestal lab in Peter's hands.

A unique aspect of my experience at Princeton has been the opportunity to advise several fantastic students with their senior theses. It was a pleasure to work with Sean Sketch, Akhil Reddy, and Christine Odabashian. I hope they gained as much from the experience as I did.

Candy Reed, whose title should be "Department Morale Officer," has played a major role in making life in the department so enjoyable. Jess O'Leary and, more recently, Jill Ray have been invaluable in helping graduate students keep on top of everything they have to do, and Valerie Carroll and Marcia Kuonen are godsends to anyone dealing with financial business with the University.

Jeremy Kasdin, Michael Littman, Luigi Martinelli, and Robert Stengel have been influential members of the faculty who have provided advice on both my research and my career, and pushed me to work hard as a teaching assistant.

Tristen Hohman and Jess Shang were great office mates during my first few years at Princeton and have become valued friends.

In my time at Princeton, I have become much better at rock climbing due to the opportunity to practice with great people, including Elena Krieger, Jonathan Tu, Jeff Santner, Amina Kinkhabwala, and Mike Hepler, among others. Climbing is a social sport, and doing it with these friends has made it even more enjoyable.

I thank Tyler and Kim Groff for being friends I can always count on: you guys are as solid as a rock. I thank Keith Moored and Christina Viau Haden for their friendship, which is so strong that it seems like we must have known each other for much longer than we actually have.

Hamish Robb is one of the first people I met in Princeton, my oldest friend here, and has been a wonderful housemate over the years. I thank my other housemates, Brian, Chris, Jeff, Fred, Desmond, and Mark, for making 75 S. Harrison Street a great place to live for the past four years.

I would not be who I am today without my family. I thank my sister Sophie for sharing life's interesting path with me. I thank my parents Nancy Tankelson and François Reverdy for giving me a solid foundation, always supporting me, and always taking an interest in my work.

My time at Princeton was supported by the Gordon Y.S. Wu first year fellowship and the National Defense Science & Engineering Graduate Fellowship, for which I am extremely grateful.

I thank Phil Holmes and Mengdi Wang for serving as my thesis readers and providing helpful feedback, which has helped me improve and clarify the final document. When writing this thesis, I sometimes forgot that the reader had not spent the last five years immersed in the subject to the same extent that I had been. Phil and Mengdi have helped me keep the reader in mind. The responsibility for any remaining errors or omissions is entirely mine.

This dissertation carries the number T-3292 in the records of the Department of Mechanical and Aerospace Engineering.

To my family and friends, who inspire me
"To strive, to seek, to find, and not to yield."
(Tennyson)

Qu'est-ce qui nous tente ?
Qu'est-ce qui nous donne ces envies ?
Qu'est-ce qui nous enchante,
Qu'est-ce qui nous réveille la nuit ?
(Louise Attaque)

Contents

List of Figures

Chapter 1

Introduction

"To live effectively is to live with adequate information."

(Norbert Wiener)

Search is a ubiquitous human activity. In our data-heavy age, the word *search* generally conjures up thoughts of search for information, as with a search engine. However, informational search is only one of a variety of ways search appears in our professional and personal lives. Search can be physical: I might search for my lost keys or a hiker in the woods. Search can be logistical: I might search for a course of action, e.g., a route between two known locations. Search can also be philosophical: I might search for meaning or my calling in life. The common thread that connects all of these scenarios is uncertainty. Search is a rational response to uncertainty, whether it be in where to find information or an object, uncertain outcomes from a course of action, or in the right path to follow in life.

Just as there are many types of search problems which arise in practice, there are also many solutions. In many situations, such as routing communications traffic, we have trusted automated machines to perform search for decades [47]. At the other end of the spectrum, searching for a car one wants to purchase is an essentially human endeavor. Technology might help supply relevant information, but ultimately it must be a human who does the searching, weigh the different options, and makes the final decision. In the middle are a host of search tasks that can benefit from some automation but are too complicated to be fully automated, for example because there is too much uncertainty about the task to able to consider every possibility. Such tasks are commonplace in applications, where they are solved by human-machine hybrid systems. As reported in [73], modern internet search engines are an example of one such system where humans and automated machinery work together. When the automation encounters new or unexpected queries, it asks human operators for help in parsing them to ensure that the system outputs useful results.

The field of control theory studies systems and develops technology to automate, or control, their operation. Search is relevant to control theory, particularly when there is significant uncertainty about the system to be controlled. In terms of the above taxonomy, logistical search, i.e., search for policies, is a form of control. For example, consider a car trying to reach a destination by traveling through rush hour traffic. The car's best route will be heavily affected by the traffic congestion, which is likely to be uncertain, due to unforeseen accidents, etc. Therefore, finding the best route is a form of logistical search, where the best route will change depending on the evolving and uncertain road conditions.

One major application area for control theory is in the field of robotics, where human-machine hybrid systems are also common. In tightly-regulated environments like the interiors of factories, automated robotic systems have proved greatly successful in performing repetitive tasks like welding more quickly and with more precision than the humans they replaced [80]. The tightly-regulated nature of the factory environment makes the welding task highly predictable. Because of this predictability, uncertainty is not an important part of the task and as such there is no need to actively search for policies. However, in other applications where the environment is more poorly regulated, uncertainty is important and the robotic control problem can be considered a search task. In such applications, human supervisors form an essential part of the system for dealing with unforeseen circumstances and ensuring that the overall system meets its goals. For example, in the field study described in [67], a mobile sensor network was deployed to gather oceanographic information. There was significant uncertainty about the environment, the state of the system (due to communication delays), and where best to deploy the sensors. These uncertainties made the network resource allocation problem a search task which was solved by a human-machine system. Human supervisors made high-level decisions to allocate the network's resources (i.e., the sensors), while the automated component carried out the low-level tasks to implement the allocation and complete the task at hand.

Adaptive control [60, 5] is a subfield of control theory that has developed to attempt to produce automated controllers with a capacity for adaptation and in-the-loop learning. Such a capacity is useful for dealing with systems with considerable uncertainty, i.e., systems for which the control problem is a form of logistical search, so there is a deep connection between adaptive control and search. Autonomy is a broad concept that deals with the ability of an agent to interact with its environment independently of external control. See [113] for a study of the concept of autonomy in the context of robots and other agent-based systems.

Adaptive control and autonomy are valuable areas of active research, but as pointed out in [5, Section 13.5], adaptive controllers often ultimately "attempt to mimic or describe human learning ability." Similarly, in the field of autonomy, humans are a natural model system

to be mimicked or improved upon. As such it is of interest to understand how a human operator performs search since it can advance the fields of adaptive control and autonomy. Furthermore, applications will always push the field of control to address complex tasks that are beyond the state of the art of fully autonomous control. The solution of such tasks will necessarily include a human supervisor as part of the control loop. Control theory can provide useful guarantees of stability and convergence about the automated component of a human-in-the-loop control system, but understanding the overall performance of the system requires a principled understanding of the human component of the system.

1.1 Motivation and goals

Machines and humans have different strengths that can be complementary in performing search tasks. Compared with humans, machines have abundant computational power and precise data storage. They can also reliably perform repetitive tasks without becoming bored or otherwise losing focus. Conversely, humans are more flexible in their thinking, better at discovering patterns in data, and develop experience-based knowledge that may be difficult to represent quantitatively. The motivation for this thesis is the desire to develop tools that benefit from these complementary strengths.

In most situations when a human performs a search task, it is rarely as an individual acting alone, but rather as part of a group and often with some kind of technological aid. For example, a search for a lost hiker would likely involve multiple people communicating with each other and using maps or other technology to share information and decide where to look next. The ultimate goal of this thesis is two-fold: first, to learn heuristics from humans to improve automated search, and second, to use automation to help humans perform search, thereby improving existing technological aids to search. We are interested in robotics applications, so the prototypical problem of interest is that of spatial search.

Key to achieving this goal is building models of human decision making in search tasks. These models should reflect empirical data from human subject experiments and capture the trends evident in human behavior in such tasks. To be useful to automated search, they should be sufficiently computationally simple to be implemented in real time: a model that requires hours of computation to make a decision is not useful as part of a deployed system. Key to improving technological aids to search is developing an understanding of transferrable knowledge in search tasks, i.e., a way to represent the relevant information that the human captures in his/her decision-making process so that it can be used by an automated decision maker. In this thesis, we aim to develop a framework for human-machine

search by formulating a decision-making model that captures transferrable knowledge and empirical trends in behavior in a form that can also be used as a deployable algorithm.

Central to both goals is the desire for mathematical rigor. The models we develop should be plausible and accurate, but simple enough to allow analysis. As control theorists we are interested in system properties such as stability, accuracy, and speed of convergence. In particular, we want to be able to prove performance guarantees about the resulting systems. This desire to produce provable guarantees leads us to pick a model problem in which proving such guarantees is possible. We wish to model the search process in a way that is simple enough to allow performance guarantees but general enough to encompass many applications.

1.2 Background and related work[1]

The mathematical tools at the core of this work are optimization theory [115] and Bayesian statistics [66]. Search can often be mathematically formulated as an optimization problem, so the technology of search is strongly linked with that of optimization. This link will become apparent in the chapters that follow, as we develop a framework for human-machine search. Search and Bayesian statistics go hand-in-hand, as was shown in the search for the wreckage of Air France Flight AF 447 [123] as well as many other studies, such as [16], which studied the problem of finding optimal search strategies for a lost target in physical space. Bayesian methods are useful because they allow a decision maker to incorporate a variety of prior knowledge about the search task. In the case of the Air France flight, Bayesian statistics allowed the searchers to incorporate the possibility that the underwater locator beacon in the aircraft had failed, which proved to be crucial in locating the wreck. In the case of modeling human decision making, Bayesian statistics provides a natural way to incorporate knowledge and intuition due to experience and previous learning.

We place an emphasis on spatial search, in particular in the context of robotics applications. In the robotic field experiments described in [68, 67], a human-automata team composed of a number of underwater vehicles and human supervisors was deployed to gather oceanographic data. The objective was to collect data to minimize the uncertainty in the value of the scalar data field of interest, e.g., water temperature. As each vehicle moved through the sampling domain, it periodically took measurements which reduced the uncertainty in the value of the field at that time and place. In this way, an analogy can be made between the measurement process and foraging, where each vehicle moves through space and receives rewards in the form of information. By accumulating rewards, i.e., information, the vehicles reduce the uncertainty in the value of the field. This analogy motivates us to

[1] Portions of this section are adapted from [99].

consider search as a process in which the agent sequentially chooses locations to sample with the goal of maximizing the cumulative rewards aggregated over the decision process. There is uncertainty in the value of the reward at any given place because of the uncertainty in the field, so the decision of where to take the next sample is non-trivial.

The decision process described above is well modeled by the multi-armed bandit problem, introduced by Robbins [101]. In the multi-armed bandit problem, the decision-making agent makes a series of sequential decisions, at each decision time picking one of a set of options and receiving a stochastic reward. An agent solving the multi-armed bandit problem makes decisions at each of sequence of times, at each time picking one of a set of options and receiving a stochastic reward. The agent's objective is to maximize the expected value of the cumulative rewards received during the decision-making process. There is uncertainty about the distribution of rewards at any given arm: in particular, the mean value of the reward associated with a given arm is not known. Because of the uncertainty, the agent has to simultaneously prioritize selecting arms to reduce the uncertainty of the associated rewards and selecting arms that appear highly rewarding given the current information. The tension between these priorities is known as the *explore-exploit* tradeoff, and is common to many forms of decision making under uncertainty, including reinforcement learning [124] and adaptive control, as well as human [30, 75] and animal decision-making behavior [58, 51, 45].

More generally, decision-making problems that involve interacting with uncertain environments are often formulated as Markov Decision Processes (MDPs). MDPs are decision problems in which the decision-making agent is required to make a sequence of choices along a process evolving stochastically in time [124]. The stochastic nature of the process captures the uncertainty of the environment, since the agent cannot be sure of the next state of the process, even though he/she knows the current state. Partially Observable Markov Decision Processes (POMDPs) [112, 114] generalize MDPs to the case where the agent cannot observe the current state directly, and therefore is more uncertain about the environment. The theory of dynamic programming [12, 48] provides methods to find optimal solutions to generic MDPs (and POMDPs [74]), but is subject to the so-called *curse of dimensionality* [124], where the size of the problem often grows exponentially in the number of states.

The curse of dimensionality makes finding the optimal solution difficult, and in general intractable for finite-horizon problems of any significant size. Many engineering solutions of MDPs consider the infinite-horizon case, i.e., the limit where the agent will be required to make an infinite sequence of decisions. In this case, the problem simplifies significantly and a variety of reinforcement learning methods can be used to converge to the optimal solution, for example [128, 48, 124, 69, 90]. However, these methods only converge to the optimal solution asymptotically at a rate that is difficult to analyze. The UCRL algorithm

[10] addressed this issue by deriving a heuristic-based reinforcement learning algorithm with a provable learning rate.

However, the infinite-horizon limit may be inappropriate for finite-horizon tasks. In particular, optimal solutions to the finite-horizon problem may be strongly dependent on the task horizon. If a single decision is to be made, a human is likely to be conservative, since selecting an unfamiliar option is risky and even if he/she chooses a rewarding option, he/she will have no further opportunity to use the information in the same context. However, if many successive decisions must be made, discovering rewarding options is valuable.

Although the finite-horizon problem may be intractable to computational analysis, humans and other animals are confronted with it all the time. Krebs *et al.* [58] showed that birds are able to closely approximate the optimal solution of a finite-horizon two-armed bandit task, and several works, e.g., [136, 122], have shown that humans can also approximate optimal solutions of multi-armed bandit tasks. The fact that humans are able to find efficient solutions quickly with inherently limited computational power suggests that they employ relatively sophisticated heuristics for solving these problems. Elucidating these heuristics is of interest both from a psychological point of view where they may help us understand human cognitive control and from an engineering point of view where they may lead to development of improved algorithms to solve MDPs [30]. In this thesis, we seek to elucidate the behavioral heuristics at play with a model that is both mathematically rigorous and computationally tractable.

1.3 Research overview

Our approach to reaching the ultimate goal of developing a framework for human-machine search requires several steps:

1. Developing and validating models of human search behavior
2. Estimating parameters of these models
3. Systems integration to apply these models in the context of specific problems

There is an initial step 0 that we have not listed above, which is to develop a mathematical model of the search problem. This step is crucial to developing the mathematical models that underpin all the work that follows. The difficulty in doing so, as was discussed above, is that search takes on many forms and arises in many different contexts. As discussed above, the search problem we study should be sufficiently general to encompass the myriad types of search while sufficiently simple to allow us to prove performance guarantees and to garner

insight through analysis, such as sensitivities to parameters. This thesis makes contributions to steps 1 and 2 listed above but also to the crucial initial step 0.

Our contribution to the initial step 0 is to make the connection between the multi-armed bandit problem [101] and search. An agent playing a multi-armed bandit task sequentially picks one of a finite set of options and receives a stochastic reward. His/her objective is to maximize some aggregate measure of the rewards received over a given series of decisions. As a model for the prototypical spatial search problem, we introduce the spatial multi-armed bandit problem, where each option represents a discrete patch in the search space. The multi-armed bandit problem has been extensively studied in both the psychology and machine learning literatures, so it is ideal for our purposes. From the machine learning literature, there is a known bound on optimal performance [62] and a number of algorithms [40, 3, 27, 9] that achieve that bound. From the neuroscience literature, there are numerous studies [1, 122, 2, 92, 65, 131] that investigate human behavior in multi-armed bandit tasks and propose heuristic-based algorithms to explain the behavioral data. Chapter 2 provides an extensive review of the relevant literatures.

Our approach is to rigorously link the machine learning and neuroscience literatures on the multi-armed bandit problem, thereby providing a framework for human-machine search. We do this by extending results from the neuroscience literature to develop several human-inspired heuristics for solving the spatial multi-armed bandit problem. We connect the resulting heuristics to those used by algorithms in the machine learning literature and use mathematical tools from that literature to analyze the performance of the resulting human-inspired algorithm.

We term the human-inspired algorithm the Upper Credible Limit (UCL) algorithm, show that it captures empirically-observed trends in human behavior, and prove conditions under which it achieves optimal performance. Crucially, UCL is a Bayesian algorithm that incorporates prior beliefs about the structure of the reward surface and updates these beliefs as it receives new information by observing rewards at different locations. The prior beliefs form the core of the transferrable knowledge that is key to improving technological aids to search.

The value of the knowledge encoded in priors is shown by data we collect from a human-subject spatial search task. In particular, we show that human performance in this task falls into a small number of well-defined categories and that the UCL algorithm captures these categories as a function of priors. Some subjects show very good performance in this short-horizon task, outperforming algorithms with known-optimal long-run performance. In light of the UCL algorithm, we attribute this good performance to the subject having good priors, either due to experience or fast adaptation to the task. Therefore, a human-machine system could benefit from using the priors from a human operator with good performance.

The UCL algorithm can be interpreted as a model of human choice behavior in spatial search tasks. By fitting the parameters of this model to human choice data, one can provide an estimate of the priors used to make the choices. This estimation is essential to performing system integration and allowing an automated system to use the human operator's transferrable knowledge. The UCL algorithm defines a likelihood function of the parameters in a straightforward way. A maximum likelihood estimate of the parameters can be produced by maximizing the likelihood function over the parameter space, but this optimization problem is poorly-behaved and consequently difficult to solve. We develop an estimator for UCL based on an approximate likelihood function and show that it accurately estimates the parameter values. In control-theoretic terms, the UCL algorithm and associated estimator gives a plant-observer pair for human decision making in spatial search tasks that can then be used for system design.

Finally, we consider some theoretical aspects of decision making in multi-armed bandit problems with a different, human-inspired objective. In the standard multi-armed bandit problem, the decision maker seeks to maximize the expected value of their cumulative reward over the horizon of the decision process. However, the expected value objective has several shortcomings. First, it ignores risk, i.e., the dispersion of the cumulative reward over repeated tasks. Second, it fails to incorporate the cost of search, i.e., the fact that search is costly and therefore a solution that is suboptimal in terms of expected reward may prove to be good enough to obviate the need to perform further search. Simon [110, 111] argued that humans and other organisms aim to satisfice (= satisfy + suffice) rather than optimize in their decision making. We consider the multi-armed bandit problem with a satisficing objective, derive bounds on optimal performance for the problem, and develop an algorithm that achieves optimal performance.

1.4 Outline

This thesis is structured as follows. We provide detailed background on multi-armed bandit problems in Chapter 2. In Chapters 3 and 4 we develop models of human decision making in multi-armed bandit problems. In Chapter 3 we review a heuristic developed in the neuroscience literature for a two-armed bandit task and generalize it to the case of multiple arms. In Chapter 4 we develop the UCL algorithm and prove conditions under which it achieves optimal performance. In Chapter 5 we present data from a human-subject spatial search experiment and discuss how the UCL algorithm can be used as a model to fit these data by adjusting the algorithm parameters. In Chapter 6 we develop a maximum likelihood estimator to estimate the algorithm parameters, including the priors, from data. In Chapter

7 we consider the extension of the multi-armed bandit problem using a satisficing objective. In Chapter 8 we survey perspectives for future work and conclude.

Chapter 2

Search and multi-armed bandits

"Essentially, all models are wrong, but some are useful."

(George E. P. Box and Norman R. Draper)

In this chapter we introduce the multi-armed bandit problem as a model of spatial search. This problem serves as the mathematical context for our study. First, we define the problem mathematically and introduce the relevant notation. We then discuss various types of optimal solution of the problem before reviewing the relevant empirical findings on human behavior in multi-armed bandit problems from the neuroscience literature. Finally, we close by making explicit the connection between the multi-armed bandit problem and spatial search.

2.1 The multi-armed bandit problem

The multi-armed bandit problem, introduced by Robbins [101], is a sequential decision-making task in which the decision-making agent is required to make a decision at sequential instances t, $t \in \{1, \ldots, T\}$, where $T > 1$ is the horizon of the task. At each decision instance t, the agent selects one element i_t of a finite set $i \in \{1, \ldots, N\}$, where N is the total number of elements, and receives a stochastic reward r_t associated with the selected element. By analogy with the lever of a slot machine, also known as a one-armed bandit, each element i is referred to as an *arm* and the overall structure as a *multi-armed bandit*. The reward r due to picking an arm i is drawn from a stationary distribution $p_i(r)$ with mean m_i, which is constant over the duration of the problem. The agent may have some information about the distribution, for example that it is Gaussian, but the value of the mean is unknown to the agent.

The agent's objective is to pick a sequence of arms in order to maximize the expected value of the rewards it accumulates over the course of the decision process. Each choice of

arm i_t is made conditional on the information available to the agent at time t, denoted $\mathcal{F}_t$. Mathematically, the objective is

$$\max_{\{i_t|\mathcal{F}_t\}_{t=1}^T} J = \mathbb{E}\left[\sum_{t=1}^T r_t\right], \tag{2.1}$$

where the expectation is taken over the distribution of rewards observed by the agent. The sum over choices t commutes with the expected value over rewards, so (2.1) can be equivalently written as

$$J = \sum_{t=1}^T \mathbb{E}\left[r_t\right] = \sum_{i=1}^N m_i \mathbb{E}\left[n_i^T\right], \tag{2.2}$$

where n_i^T is the number of times arm i has been selected up to time T and the final expectation is taken over the distribution of choices made by the agent based on the rewards it observes. We use $*$ to denote the arm that is most rewarding on average, so $i^* = \arg\max_i m_i$ and $m_{i^*} = \max_i m_i$. For purposes of exposition, we assume that i^* is unique, although nothing substantive changes if this is not the case.

Another transformation of the objective function is useful for analyzing the performance of policies solving the multi-armed bandit problem. Define $\Delta_i = m_{i^*} - m_i$ as the expected *regret* of picking arm i, i.e., the amount of reward foregone on average by selecting arm i instead of the optimal one i^*. Furthermore, define $R_t = \Delta_{i_t}$ as the expected regret at time t due to the choice of arm i_t. Then the objective function J can be rewritten in terms of *cumulative expected regret* J_R:

$$J_R = Tm_{i^*} - J = Tm_{i^*} - \sum_{t=1}^T \mathbb{E}\left[r_t\right] = \sum_{t=1}^T \mathbb{E}\left[R_t\right] = \sum_{i=1}^N \Delta_i \mathbb{E}\left[n_i^T\right]. \tag{2.3}$$

Maximizing cumulative expected reward J is equivalent to minimizing cumulative expected regret J_R, so the optimization problem (2.1) becomes

$$\min_{\{i_t|\mathcal{F}_t\}_{t=1}^T} J_R,\ J_R = \sum_{i=1}^N \Delta_i \mathbb{E}\left[n_i^T\right]. \tag{2.4}$$

Note that this definition of regret is in the sense of an omniscient being who is aware of the expected values of all options, rather than in the sense of an agent performing the task. As such, it is not a quantity of direct psychological relevance but rather an analytical tool that allows one to characterize performance. Alternative notions of regret can be psychologically relevant, such as when a choice that looks promising a priori yields a result that appears

poor in hindsight. See [120] for a study of psychological regret in animal decision making and [32, 31, 75] for studies involving human subjects.

Writing the objective as a sum over arms i emphasizes that achieving good performance relies on choosing good arms (i.e., those with high m_i) as often as possible (and conversely, bad arms as rarely as possible). Since the means are unknown to the agent, it must learn the values m_i by picking arm i and observing rewards from the distribution p_i. For individual choices, then, the requirement to learn about the m_i by selecting arms where m_i is uncertain is in tension with the interest in obtaining immediate reward by preferentially selecting arms with known high rewards. To maximize the objective J, the agent must balance the requirement to gain information about the rewards by picking arms where the value of m_i is uncertain (termed *exploration*) and the requirement to maximize immediate reward by picking arms where the value of m_i appears relatively high (termed *exploitation*). This tension between information and immediate reward, known as the *explore-exploit tradeoff*, is common to many forms of decision making under uncertainty.

In the remainder of this thesis, we focus on the Gaussian multi-armed bandit problem, i.e., the multi-armed bandit problem where the reward distribution $p_i(r)$ is normal with mean m_i and variance $\sigma_{s,i}^2$. We denote this distribution by $\mathcal{N}(m_i, \sigma_{s,i}^2)$. The mean m_i is unknown as assumed above, but we assume that the variance $\sigma_{s,i}^2$ is known to the decision maker. In many cases, the variance is uniform across arms i, in which case we denote it simply by σ_s^2.

2.2 Optimal solution of the multi-armed bandit problem

In the literature on MDPs, the word *policy* refers to a function that maps the state of the process to the action to be taken by the decision maker. Such a function constitutes a solution of the MDP, which may be implemented using a particular algorithm. In the following, we use the words policy, solution, and algorithm interchangeably.

If the values of the means m_i were known, the optimal solution to the multi-armed bandit problem would be trivial: select the best arm $i_t = i^* = \arg\max_i m_i$ at each time t. However, since the means are unknown, the agent must negotiate the explore-exploit tradeoff, which is non-trivial. The multi-armed bandit problem can be thought of as a POMDP where the action and state variables are conflated. In particular, the state space is the set $S = \{(i, m_i) | i \in \{1, \ldots, N\}\}$ of pairs of arms and mean rewards.

The methods of dynamic programming [12, 48, 74] can be used to find optimal solutions to general MDPs, including the multi-armed bandit problem, but these methods quickly

become computationally intractable as the horizon T grows, due to the well-known curse of dimensionality [124]. The theory of approximate dynamic programming [90] provides tools (e.g., the knowledge gradient algorithm [104]) to break the curse of dimensionality and approximate the dynamic programming optimal solution, but the performance bounds for the resulting approximate solutions can be difficult to interpret in the finite-horizon case $T < +\infty$.

Many engineering approaches to solving MDPs consider the infinite-horizon limit $T \to +\infty$, in which case finding optimal solutions often simplifies dramatically. Gittins [40, 39] considered the infinite-horizon limit of the multi-armed bandit problem and developed a dynamic allocation index (Gittins' index) for each arm. He showed that selecting the arm with the highest allocation index at each decision time results in the optimal policy. Gittins' index, therefore, provides an optimal solution to the infinite-horizon multi-armed bandit problem, but it has several drawbacks. First, it is difficult to compute, and second, it does not provide much insight into the nature of optimal policies, such as sensitivities to the problem parameters.

In a seminal study [62], Lai and Robbins considered a case of the multi-armed bandit problem where the reward distributions p_i are members of a one-parameter exponential family [50]. For this case, they derived a lower bound on $\mathbb{E}\left[n_i^T\right]$, the number of times a policy solving the multi-armed bandit problem will pick a suboptimal arm. In particular, they showed [62, Theorem 2] that

$$\mathbb{E}\left[n_i^T\right] \geq \left(\frac{1}{D(p_i||p_{i^*})} + o(1)\right) \log T \tag{2.5}$$

for each suboptimal arm $i \neq i^*$, where $D(p_i||p_{i^*}) := \int_{-\infty}^{\infty} \log\left(\frac{p_i(x)}{p_{i^*}(x)}\right) p_i(x) \mathrm{d}x$ is the Kullback-Leibler divergence between the reward distributions for arm i and i^*. The bound is asymptotic in the horizon T, i.e., $o(1) \to 0$ as $T \to +\infty$ and the constant depends on the distinguishability of the reward distributions. If p_i and p_{i^*} are easily distinguished, the Kullback-Leibler divergence $D(p_i||p_{i^*})$ will be large and the corresponding constant in (2.5) will be small. By substituting the bound (2.5) into the regret objective (2.4), one obtains the following asymptotic lower bound on regret:

$$J_R = \sum_{i=1}^{N} \Delta_i \mathbb{E}\left[n_i^T\right] \geq \sum_{i \neq i^*} \Delta_i \left(\frac{1}{D(p_i||p_{i^*})} + o(1)\right) \log T. \tag{2.6}$$

Therefore, this result shows that any policy solving the multi-armed bandit problem must incur cumulative expected regret that grows logarithmically with the number of choices made T.

In the case the the rewards are Gaussian with uniform variance σ_s^2, the Kullback-Leibler divergence is

$$D(p_i||p_{i^*}) = \frac{\Delta_i^2}{2\sigma_s^2},$$

so the bound (2.5) is

$$\mathbb{E}\left[n_i^T\right] \geq \left(\frac{2\sigma_s^2}{\Delta_i^2} + o(1)\right) \log T. \tag{2.7}$$

This result can be interpreted as follows. For a given value of Δ_i, a larger variance σ_s^2 makes the rewards more variable and therefore it is more difficult to distinguish between the arms. For a given value of σ_s^2, a larger value of Δ_i makes it easier to distinguish the optimal arm.

2.3 Heuristic solutions of the multi-armed bandit problem

Since finding the strictly optimal solution of the multi-armed bandit problem is generally intractable, the literature has focused on finding approximately optimal solutions, often based on heuristics. Often the heuristics are forms of a metaheuristic known in the machine learning literature as *optimism in the face of uncertainty* [20]. The idea behind this metaheuristic is that an agent interacting with an uncertain world should formulate a set of possible states of the world that are consistent with the observed data, then act as if the true state of the world were the one that is most favorable to the agent.

The Lai-Robbins bound (2.5) provides a way to identify approximately optimal policies in terms of the growth rate of their cumulative expected regret. Two major categories of results exist in the literature: 1) infinite-time policies that attempt to match the asymptotic growth rate (2.5), and 2) finite-time policies that bound the cumulative expected regret for any given finite horizon T. In the literature on infinite-time policies, optimal performance refers to regret that matches the asymptotic bound, while for finite-time policies, optimal performance refers to cumulative expected regret that is uniformly upper bounded in time T by a logarithmic function with a leading constant that is within a constant factor of that in (2.5). In the following, we refer to cumulative expected regret that obeys such a uniform upper bound as *logarithmic regret*.

2.3.1 Asymptotically-optimal policies

In the paper [62] where they proved the bound (2.5), Lai and Robbins also developed infinite-time algorithms that asymptotically achieve the bound, i.e., asymptotically optimal performance. Their algorithms use what they term "upper confidence bounds" on the mean rewards m_i, which give a value that is similar in function to Gittins' index in that the algorithms pick an arm based on a rule incorporating the upper confidence bound and the empirically observed mean reward for each arm. Lai [61] simplified the rule by developing a single upper confidence bound that incorporates all the relevant information. Lai's algorithm picks the arm that has the largest upper confidence bound at each decision time. In Example 1 of [61], Lai develops a specific version of his algorithm for the case of Gaussian bandits with known uniform variance $\sigma_s^2 = 1$.

The algorithms developed by Lai and Robbins [62] and Lai [61] use heuristic functions that are complicated to compute. Agrawal [3] developed simplified heuristic functions that depend only on the empirically observed mean reward and as such are easy to compute, yet still achieve asymptotically optimal performance. His results depend on several results from large deviation theory to bound the rate at which the observed mean reward converges to the true mean.

2.3.2 Finite-time optimal policies[1]

Auer *et al.* [9] considered the finite-time multi-armed bandit problem where the rewards are drawn from distributions with a bounded support. For this case they developed the Upper Confidence Bound (UCB) algorithm UCB1 and its variants, and showed that they achieve logarithmic regret. UCB1 is a heuristic-based algorithm that computes a heuristic value Q_i^t for each arm i at each decision time t. This heuristic value, derived by applying ideas from Agrawal [3], provides an upper bound on the true mean reward value m_i for that option

$$Q_i^t = \bar{m}_i^t + C_i^t, \tag{2.8}$$

where $\bar{m}_i^t$ is the empirical mean reward observed from arm i up to time t and C_i^t is a measure of the uncertainty associated with the estimate of the mean at time t. UCB1 then picks the arm that maximizes the heuristic function Q_i^t. Figure 2.1 depicts the components of the UCB heuristic (2.8) in an $N = 3$ option case.

Auer *et al.* [9] showed that UCB1 achieves logarithmic regret, albeit with a larger constant than the optimal asymptotic one (2.5). They also developed a slightly more complicated

[1] This section is adapted from Section II.C of [99] with some text taken verbatim.

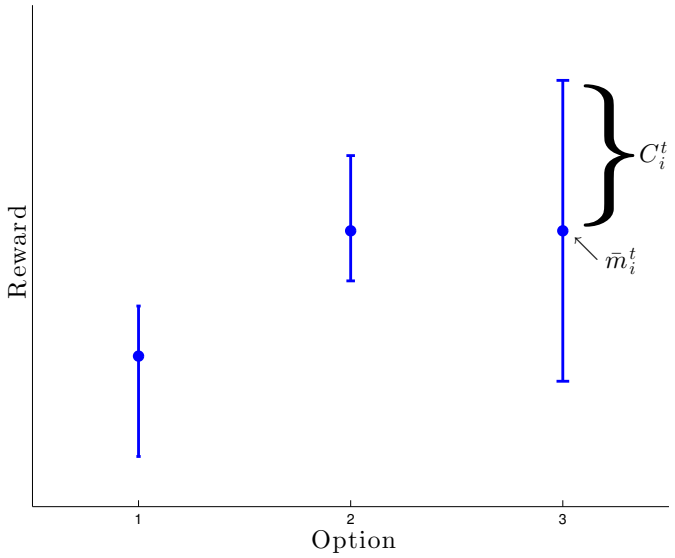


Figure 2.1: Components of the UCB1 algorithm in an $N = 3$ option (arm) case. The algorithm forms a confidence interval for the mean reward m_i for each option i at each time t. The heuristic value $Q_i^t = \bar{m}_i^t + C_i^t$ is the upper limit of this confidence interval, representing an optimistic estimate of the true mean reward. In this example, options 2 and 3 have the same mean $\bar{m}$ but option 3 has a larger uncertainty C, so the algorithm chooses option 3. Previously published as Figure 1 of [99].

policy, called UCB2, which achieves logarithmic regret with a constant that can be made arbitrarily close to the optimal one. Their analysis of UCB1 and UCB2 relies on Chernoff-Hoeffding bounds, which is a result from large deviation theory that applies to distributions with bounded support.

In the same paper, Auer *et al.* also considered the multi-armed bandit problem where the rewards are drawn from Gaussian distributions with unknown mean and unknown variance and developed a policy called UCB1-Normal. Since the Gaussian distribution has support on the whole real line, they were unable to appeal to Chernoff-Hoeffding bounds in the analysis and instead based their analysis on bounds on the tails of the χ^2 and Student distributions that they could only verify numerically. Assuming these bounds, they showed that UCB1-Normal achieves logarithmic regret. Liu and Zhao [70] studied multi-armed bandit problems where the rewards are drawn from a light-tailed distribution, which includes Gaussian distributions with known variance as a special case. For such light-tailed rewards, they extended UCB1 to achieve logarithmic regret. In contrast to the Bayesian algorithms developed in this thesis, UCB1 and its variants rely on frequentist estimators, and therefore cannot incorporate prior knowledge about the rewards.

2.3.3 Bayesian algorithms[2]

UCB algorithms rely on a frequentist estimator $\bar{m}_i^t$ of m_i and therefore must sample each arm at least once in an initialization step, which requires a sufficiently long horizon, i.e., $T > N$. Bayesian estimators allow the integration of prior beliefs into the decision process. This enables a Bayesian UCB algorithm to treat the case $T < N$ as well as to capture the initial beliefs of an agent, informed perhaps through prior experience. Kauffman *et al.* [49] considered the N-armed bandit problem from a Bayesian perspective and proposed the quantile function of the posterior reward distribution as the heuristic function (2.8).

In their analysis, Kauffman *et al.* considered the case of an uninformative prior, in which case the Bayesian estimator reduces to its frequentist equivalent. They do this because of the difficulty of analyzing the information provided by an informative prior. Intuitively, an informative prior represents information gained from previous experience. If this information is accurate and relevant to the task, it can greatly improve performance. If, however, the information is inaccurate or irrelevant, it may take time to realize that this is the case, resulting in poor initial decisions and therefore poor performance. When we introduce the UCL algorithm in Chapter 4, we also perform our analysis in the case of an uninformative prior. However, we also derive a metric for determining the quality of an informative prior, and we use this metric to develop intuition about the kind of information that is valuable for improving performance.

For every random variable X having support $\mathbb{R} \cup \{\pm\infty\}$ with probability distribution function (pdf) $f(x)$, the associated cumulative distribution function (cdf) $F(x)$ gives the probability that the random variable takes a value of at most x, i.e., $F(x) = \mathbb{P}(X \leq x)$. See Figure 2.2. Conversely, the *quantile* function $F^{-1}(p)$ is defined by

$$F^{-1} : [0, 1] \to \mathbb{R} \cup \{\pm\infty\},$$

i.e., $F^{-1}(p)$ inverts the cdf to provide an upper bound for the value of the random variable $X \sim f(x)$:

$$\mathbb{P}\left(X \leq F^{-1}(p)\right) = p. \tag{2.9}$$

In this sense, $F^{-1}(p)$ is an *upper confidence bound*, i.e., an upper bound that holds with probability, or *confidence level*, p. Now suppose that $F_i(r)$ is the cdf for the posterior estimate of the reward distribution $p_i(r)$ of option i. Then, $Q_i = F_i^{-1}(p)$ gives a bound such that $\mathbb{P}(m_i > Q_i) = 1 - p$. If $p \in (0, 1)$ is chosen large, then $1 - p$ is small, and it is unlikely that the true mean reward for option i is higher than the bound. See Figure 2.3.

[2]This section is adapted from Section II.D of [99] with some text taken verbatim.

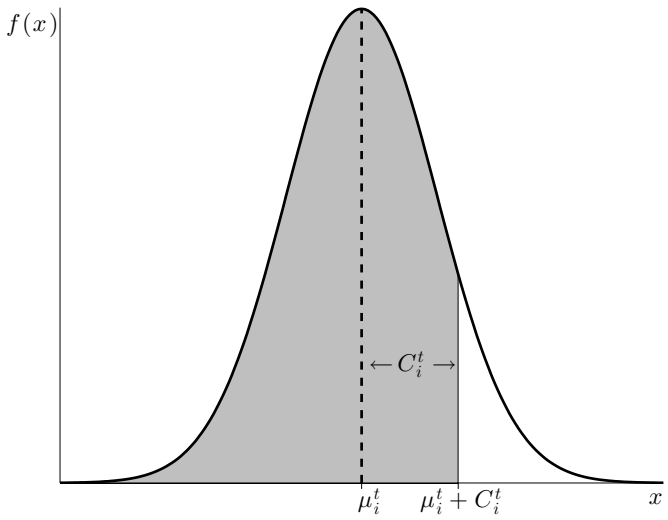


Figure 2.2: The pdf $f(x)$ of a Gaussian random variable X with mean μ_i^t. The probability that $X \leq x$ is $\int_{-\infty}^{x} f(X)\,\mathrm{d}X = F(x)$. The area of the shaded region is $F(\mu_i^t + C_i^t) = p$, so the probability that $X \leq \mu_i^t + C_i^t$ is p. Conversely, $X \geq \mu_i^t + C_i^t$ with probability $1 - p$, so if p is close to 1, X is almost surely less than $\mu_i^t + C_i^t$. Previously published as Figure 2 of [99].

In order to be increasingly sure of choosing the optimal arm as time goes on, [49] sets $p = 1 - \alpha_t$ as a function of time with $\alpha_t = 1/(t(\log T)^c)$, so that $1 - p$ is of order $1/t$. The authors of [49] defined an algorithm that picks arms using the heuristic function $Q_i^t = F_i^{-1}(1 - \alpha_t)$ and called it Bayes-UCB. In the case that the rewards are Bernoulli distributed, they proved that with $c \geq 5$ Bayes-UCB achieves logarithmic regret, i.e. optimal performance, for uniform (uninformative) priors.

The choice of $1/t$ as the functional form for α_t can be motivated as follows. Roughly speaking, α_t is the probability of making an error (i.e., choosing a suboptimal arm) at time t. If a suboptimal arm is chosen with probability $1/t$, then the expected number of times it is chosen until time T will follow the integral of this rate, which is $\sum_1^T 1/t \approx \log T$, yielding a logarithmic functional form.

Another Bayesian algorithm that is the subject of active research is known as Thompson sampling [125]. Recent work has shown that Thompson sampling is near-optimal for binary bandits with a uniform prior [4]. Bubeck and Liu [21] studied Thompson sampling focusing on the effect of priors on regret. In the following, inspired by models of human decision making, we focus on the UCB approach.

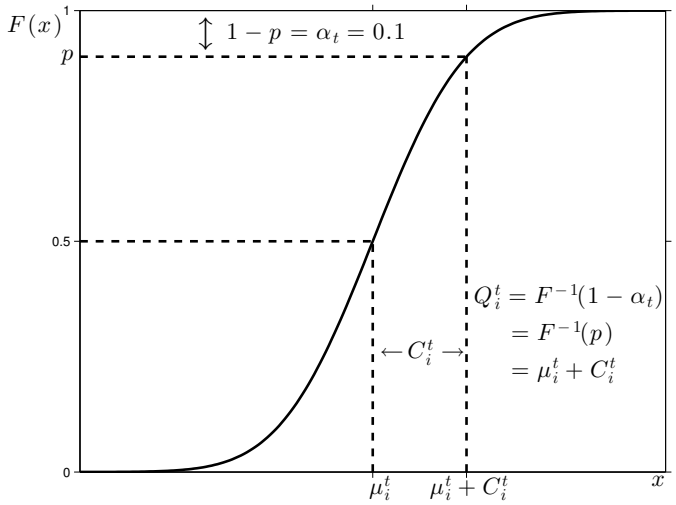


Figure 2.3: Decomposition of the Gaussian cdf $F(x)$ and relation to the UCB/Bayes-UCB heuristic value. For a given value of α_t (here equal to 0.1), $F^{-1}(1 - \alpha_t)$ gives a value $Q_i^t = \mu_i^t + C_i^t$ such that the Gaussian random variable $X \leq Q_i^t$ with probability $1 - \alpha_t$. As $\alpha_t \to 0$, $Q_i^t \to +\infty$ and X is almost surely less than Q_i^t. Previously published as Figure 3 of [99].

2.4 Results from neuroscience[3]

As discussed in the introduction, human decision making in the multi-armed bandit task has been the subject of numerous studies in the cognitive psychology literature. We list the five salient features of human decision making in this literature that we wish to capture in our models.

(i) **Familiarity with the environment:** Familiarity with the environment and its structure plays a critical role in human decision making [30, 122]. In the context of multi-armed bandit tasks, familiarity with the environment translates to prior knowledge about the mean rewards from each arm.

(ii) **Ambiguity bonus:** Wilson *et al.* [131] showed that the decision at time t is based on a linear combination of the estimate of the mean reward of each arm and an *ambiguity bonus* that captures the value of information from that arm. In the context of UCB and related algorithms, the ambiguity bonus can be interpreted similarly to the C_i^t term of (2.8) that defines the size of the upper bound on the estimated reward.

(iii) **Stochasticity:** Human decision making is inherently noisy [30, 1, 122, 136, 131]. This is possibly due to inherent limitations in human computational capacity, or it could be the

[3]This section is adapted from Section III of [99]. In particular, the list of features is mostly taken verbatim, as well as the final paragraph.

signature of noise being used as a cheap, general-purpose problem-solving algorithm. In the context of algorithms for solving the multi-armed bandit problem, this can be interpreted as picking arm i_t at time t using a stochastic arm selection strategy rather than a deterministic one.

(iv) **Finite-horizon effects:** Both the level of decision noise and the ambiguity bonus are sensitive to the time horizon T of the bandit task [30, 131]. This is a sensible feature to have, as shorter time horizons mean less time to take advantage of information gained by exploration, therefore biasing the optimal policy towards exploitation. The fact that both decision noise and the ambiguity bonus are affected by the time horizon suggests that they are both working as mechanisms for exploration, as investigated in [100] and the following chapter. In the context of algorithms, this means that the uncertainty term C_i^t and the stochastic arm selection scheme should be functions of the horizon T.

(v) **Environmental structure effects:** Acuña *et al.* [2] showed that an important aspect of human learning in multi-armed bandit tasks is structural learning, i.e., humans learn the correlation structure among different arms, and utilize it to improve their decision.

In this thesis, we develop plausible models for human decision making that capture these five features. The ambiguity bonus heuristic algorithm developed in Chapter 3 represents a first step towards modeling features (i)-(iii) and (v) of human decision making. The UCL algorithm developed in Chapter 4 develops a model that addresses all five features and is more analytically tractable, allowing us to prove conditions under which it achieves optimal performance.

Feature (i) of human decision making is captured through priors on the mean rewards from the arms. The introduction of priors in the decision-making process suggests that non-Bayesian upper confidence bound algorithms [9] cannot be used, and therefore, we focus on Bayesian upper confidence bound (upper credible limit) algorithms [49]. Feature (ii) of human decision making is captured by making decisions based on a metric that comprises two components, namely, the estimate of the mean reward from each arm, and the width of a credible set. It is well known that the width of a credible set is a good measure of the uncertainty in the estimate of the reward. Feature (iii) of human decision making is captured by introducing a stochastic arm selection strategy in place of the standard deterministic arm selection strategy [9, 49]. In the spirit of Kauffman *et al.* [49], we choose the credibility parameter α_t as a function of the horizon length to capture feature (iv) of human decision making. Feature (v) is captured through the correlation structure of the prior used for the Bayesian estimation. This correlation structure proves crucial to applying the multi-armed bandit problem to spatial search.

2.5 A model of spatial search

An important contribution of this thesis is to make the connection between the multi-armed bandit problem and spatial search. The connection is made by considering the arms of the bandit to be spatially embedded, so each arm represents a discrete patch of the spatial domain of interest. We refer to this multi-armed bandit problem with spatially-embedded arms as the *spatial multi-armed bandit problem*.

In the spatial multi-armed bandit problem, the decision-making agent receives rewards by sampling arms, i.e., patches of search space. As an example, this could be a model of an animal's foraging behavior. At each decision time t, the animal decides to forage in a given region of its spatial domain and receives a stochastic reward in the form of food. The decision time might represent a short period of several minutes, a day, or a season, depending on the time scales involved. As a model of decision making on evolutionary time scales, the decision-making agent could be an entire population and each decision time could represent a generation. For a more detailed discussion of the connection between spatially embedded multi-armed bandits and foraging theory, see [117].

The key mathematical difference between the standard and the spatial multi-armed bandit problems is the introduction of a correlation structure. When the arms are embedded in a metric space, it is natural to assume that arms that are spatially close have similar rewards. In a Bayesian context, this corresponds to having a prior with correlation structure where spatially close arms are highly correlated. The spatial multi-armed bandit problem is closely related to the so-called continuous-armed bandit problem [54], where each point x of a continuous space X is considered as an arm. In the continuous-armed bandit problem, the mean reward value is the function $m : X \to \mathbb{R}$, where $m(x)$ is the reward at "arm" $x \in X$. Correlation structure is encoded in smoothness assumptions about the function $m(\cdot)$.

In one body of recent work, the mean reward function $m(\cdot)$ is assumed to be Lipschitz (implying a little more than continuity), and the problem is referred to as a Lipschitz multi-armed bandit problem. Kleinberg and Slivkins [55] and Kleinberg *et al.* [56] considered the Lipschitz multi-armed bandit problem, where the arms are embedded in a metric space. They showed that when the arms are embedded in an infinite metric space, the lower bound on the regret growth rate is $\mathcal{O}\left(\sqrt{T}\right)$ as opposed to the slower rate $\mathcal{O}\left(\log T\right)$ from the Lai-Robbins bound (2.5) for the finite-armed bandit problem. This implies that moving from a discrete to a continuous space makes the bandit problem significantly more difficult. Bubeck *et al.* [23] also considered Lipschitz multi-armed bandits and derived an algorithm that achieves $\mathcal{O}\left(\sqrt{T}\right)$ regret. Azar *et al.* [11] extended the continuous-armed bandit formalism to include

cases where the stochastic reward depends on the prior history of choices. This allowed them to develop algorithms to perform policy search in MDPs.

In the Lipschitz multi-armed bandit problem the smoothness assumption about $m(\cdot)$ is encoded in the Lipschitz constant. In a Bayesian framework, the mean function $m(\cdot)$ is considered to be a random variable and the smoothness assumption is encoded in the prior distribution over $m(\cdot)$. One distribution commonly used is the Gaussian process, which is defined by a kernel function $k : X \times X \to \mathbb{R}$ and a mean function $\mu : X \to \mathbb{R}$. The value $k(x, x')$ represents the covariance of the belief about $m(x)$ and $m(x')$, while the value $\mu(x)$ represents the mean belief about $m(x)$. The smoothness assumption about $m(\cdot)$ is encoded in the structure of $k(\cdot, \cdot)$. Srinivas *et al.* [116] studied the continuous-armed bandit problem in a Bayesian context using Gaussian processes as a prior and developed the Gaussian process upper confidence bound (GP-UCB) algorithm and characterized its regret.

In the following, we consider the spatial multi-armed bandit problem, which can be thought of as a discretized version of the continuous-armed bandit problem. In the Bayesian context, the covariance structure comes from discretizing the kernel function: for example, it is natural to think of a covariance structure defined by $\Sigma_{ij} = \sigma_0^2 \exp(-|x_i - x_j|/\lambda)$, where x_i is the location of arm i, $\lambda \geq 0$ is the correlation length scale parameter that encodes the spatial smoothness of the rewards, and $\sigma_0^2 \geq 0$ is a confidence parameter that encodes the strength of the prior.

Chapter 3

Human-inspired heuristics for multi-armed bandit problems[1]

"When we talk mathematics, we may be discussing a secondary language built on the primary language of the nervous system."

(John von Neumann)

In this chapter, we review an empirical result, due to Wilson *et al.* [131], from the neuroscience literature concerning a two-armed bandit task. The authors of [131] showed that human decision-making behavior in such a task is well modeled by an *ambiguity bonus* heuristic that resembles the UCB heuristic (2.8). This heuristic includes two mechanisms for exploration: directed exploration to gain information about the reward values based on the agent's model of the world and random exploration that uses noise to stochastically try new options in a model-free way. By weighting these two mechanisms with a mechanism for exploitation, the heuristic negotiates the explore-exploit tradeoff.

Seeking to develop an analogous heuristic-based algorithm, we extend the ambiguity bonus heuristic to the multi-armed spatial bandit task with $N \geq 2$ arms and study its properties both numerically and analytically. We show that, with proper parameter tunings, the heuristic-based algorithm performs well but that finding the optimal parameter values is non-trivial. However, in some cases the optimal parameter values can be found analytically and suboptimal parameter values can provide robustness to modeling error. These results suggest a feedback control law for dynamically optimizing the parameters.

[1]This chapter is adapted from [100] with most of the text taken verbatim. Some notation has been changed for consistency with the rest of this thesis.

3.1 Results from a two-armed bandit task

Wilson *et al.* [131] studied human behavior in a two-armed bandit task and showed that decisions were well explained by an *ambiguity bonus* heuristic. This heuristic makes decisions based on a value function Q_i^t, which assigns to each option (i.e. arm) i at each decision time t a value that trades off the expected payoff of that option, ΔR_i^t, with the information to be gained by testing it, ΔI_i^t:

$$Q_i^t = \Delta R_i^t + A\Delta I_i^t,\ A \in \mathbb{R}. \tag{3.1}$$

The heuristic then picks the option i_t using softmax action selection [124, Section 2.3], which is a stochastic strategy that preferentially chooses options with higher Q values. The name of the heuristic derives from the influence of ΔI_i^t. Choosing options that have greater ambiguity, i.e., less is known about their associated rewards, yields more information ΔI_i^t; for $A > 0$ these options are assigned greater values by the heuristic. Two limiting values of A are of interest. When $A = 0$, the heuristic attaches no weight to information gain and therefore reduces to a greedy strategy, i.e., one that selects the option whose currently-estimated reward is highest. In terms of the explore-exploit tradeoff, this can be thought of as a pure exploit strategy [124, Section 2.1]. Conversely, when $A \to +\infty$ the heuristic weights only information and can be thought of as a pure explore strategy.

To make the tradeoff between these two limiting cases more explicit, in the following we consider an alternative parametrization of (3.1) using a convex instead of a linear combination:

$$Q_i^t = \beta\Delta R_i^t + (1-\beta)\Delta I_i^t,\ \beta \in \mathbb{R}. \tag{3.2}$$

Using this parametrization, $\beta = 1$ corresponds to a pure exploit strategy and $\beta = 0$ corresponds to pure explore. Intermediate values of β correspond to mixed explore-exploit strategies.

3.2 Generalization to N arms

In Wilson *et al.*'s study [131], the subject's belief about ΔR_i^t was experimentally controlled and ΔI_i^t was approximated using the number of samples the subject had seen from a given arm i. To formulate a generalization of the ambiguity bonus heuristic in the general case of $N \geq 2$ arms, we must pose the problem mathematically as a multi-armed bandit problem and develop equivalent notions of ΔR and ΔI in the mathematical framework.

Consider a d-dimensional discrete grid with $N = n^d$ grid points, where each grid point $i \in \{1, \ldots, N\}$ is located at $\mathbf{x}_i \in (\mathbb{Z}_n)^d$. In the following, we consider the cases $d \in \{1, 2\}$, but

the generalization to arbitrary dimensions is straightforward. Each of the N grid points $\mathbf{x}_i$ constitutes an arm i of a spatial multi-armed bandit and has an associated mean reward m_i, which remains fixed for the duration of the problem. The vector $\mathbf{m} = \begin{bmatrix} m_1 & m_2 & \cdots & m_N \end{bmatrix} \in \mathbb{R}^N$ of the rewards is unknown to the agent but is assumed to be drawn from a distribution $\mathcal{D}$ with mean $\bar{\boldsymbol{\mu}} \in \mathbb{R}^N$ and covariance $\overline{\Sigma} \in \mathbb{R}^{N \times N}$.

We assume the rewards to be Gaussian, so the agent collects rewards by visiting one grid point i_t at each time $t \in \{1, \ldots, T\}$ and receiving reward r_t which is the mean reward associated with the point plus Gaussian noise: $r_t \sim \mathcal{N}(m_{i_t}, \sigma_s^2)$. The agent's objective is to maximize cumulative expected rewards by selecting a sequence of grid points $\{\mathbf{x}_{i_t}\}$, i.e., arms $\{i_t\}$ (cf. Equation (2.1)). Note that due to the stationary nature of the sampled reward, in the long time horizon limit $T \gg N$ this problem reduces to the problem of finding the peak value among the m_i. We are particularly interested in the case of large spaces or short time horizons, in which case the explore-exploit tension is consequential. A similar situation arises in the long time horizon limit if the rewards are non-stationary.

3.3 The ambiguity bonus heuristic algorithm

In order to solve the optimization problem, the agent needs to learn about the reward surface and make a decision based on their beliefs. With reasonable assumptions on the distribution of rewards $\mathbf{m}$, Bayesian inference provides a tractable optimal solution to the learning problem. The ambiguity bonus heuristic (3.2) then provides a tractable solution to the decision problem.

3.3.1 Inference algorithm

We begin by assuming that the agent's prior distribution on $\mathbf{m}$ is multivariate Gaussian with mean $\boldsymbol{\mu}_0$ and covariance Σ_0:

$$\mathbf{m} \sim \mathcal{N}(\boldsymbol{\mu}_0, \Sigma_0),$$

where $\boldsymbol{\mu}_0 \in \mathbb{R}^N$ and $\Sigma_0 \in \mathbb{R}^{N \times N}$ is a positive-definite matrix. Note that this does not assume that the rewards are truly described by these statistics, simply that these are the agent's initial beliefs, informed perhaps by previous measurements of the mean value and covariance.

With this prior, the posterior distribution is also Gaussian, so the Bayesian optimal inference algorithm is linear and can be written down as follows. At each time t, the agent, located at $\mathbf{x}_{i_t} \in (\mathbb{Z}_n)^d$, observes a reward r_t. Define $\boldsymbol{\phi}_t \in \mathbb{R}^N$ to be the indicator vector corresponding to $\mathbf{x}_{i_t}$, where $(\boldsymbol{\phi}_t)_i = 1$ if $i = i_t$ is the location in a vector representation of

the grid, and zero otherwise. Then the belief state $(\boldsymbol{\mu}_t, \Sigma_t)$ updates as follows [50, 133]:

$$\mathbf{q} = \frac{r_t \boldsymbol{\phi}_t}{\sigma_s^2} + \Lambda_{t-1}\boldsymbol{\mu}_{t-1} \tag{3.3}$$

$$\Lambda_t = \frac{\boldsymbol{\phi}_t \boldsymbol{\phi}_t^T}{\sigma_s^2} + \Lambda_{t-1},\ \Sigma_t = \Lambda_t^{-1} \tag{3.4}$$

$$\boldsymbol{\mu}_t = \Sigma_t \mathbf{q}, \tag{3.5}$$

where $\Lambda_t = \Sigma_t^{-1}$ is the *precision* matrix. This assumes that the sampling noise σ_s is known, e.g. from previous observations or known sensor characteristics.

This gives us the first component of the decision heuristic: at time t, the expected payoff ΔR_i^t of option i is μ_i^t, the i^{th} component of $\boldsymbol{\mu}_t$. We now turn to the information value component ΔI_i^t.

3.3.2 Information value

We use entropic information as our information metric. Since the posterior distribution is Gaussian, its entropy at time t is

$$H_t = \frac{N \log 2\pi + \log\det \Sigma_t}{2} = \frac{N \log 2\pi - \log\det \Lambda_t}{2},$$

where the second equality comes from the definition of Λ_t. This form of the expression for the entropy is convenient because the Λ_t update rule (3.4) is linear and $\boldsymbol{\phi}_t \boldsymbol{\phi}_t^T$ is a sparse rank one matrix: at each time t, $\left(\boldsymbol{\phi}_t \boldsymbol{\phi}_t^T\right)_{i_t i_t} = 1$ is the only non-zero element.

Because of this sparsity, we can calculate the change in the determinant over one time step analytically using the matrix determinant lemma [18, Equation 4.3]:

$$\det \Lambda_t = \det\left(\frac{\boldsymbol{\phi}_t \boldsymbol{\phi}_t^T}{\sigma_s^2} + \Lambda_{t-1}\right) = \det \Lambda_{t-1} + \frac{1}{\sigma_s^2} M_{i_t i_t}, \tag{3.6}$$

where $M_{i_t i_t}$ is the (i_t, i_t) minor of Λ_{t-1}.

Then the change in entropy due to selecting arm i at time t is

$$\begin{aligned} H_t - H_{t-1} &= -\frac{1}{2}\left(\log\det \Lambda_t - \log\det \Lambda_{t-1}\right) \\ &= \frac{1}{2}\left(\log\det \Lambda_{t-1} - \log\left(\det \Lambda_{t-1} + \frac{1}{\sigma_s^2} M_{ii}\right)\right) \\ &= -\frac{1}{2}\frac{M_{ii}}{\sigma_s^2 \det \Lambda_{t-1}} + \mathcal{O}\left(M_{ii}^2\right), \end{aligned}$$

where the first equality follows from Equation (3.6) and the second follows from the Taylor series expansion of $\log(x)$ about $x = \det \Lambda_{t-1}$. The $\mathcal{O}(M_{ii}^2)$ term becomes increasingly insignificant as t increases and the information gain decreases.

Motivated by this approximation we define the information value of location i at time t to be

$$\Delta I_i^t = \frac{M_{ii}}{\sigma_s^2 \det \Lambda_{t-1}}. \tag{3.7}$$

See also the Backward Selection for Gaussian method of Choi and How [29, 28], who examine other, more general cases of information-based search.

3.3.3 Decision heuristic

An important aspect of human decision making is that it is *noisy*, so that humans do not necessarily deterministically optimize a value function. For example, when faced with a completely unknown situation, a good model is that human subjects will pick randomly among their options.

We choose to incorporate decision noise in our model by adding i.i.d. (over i and t) random noise to the heuristic value function (3.2). Putting all the terms together the value function Q_i^t becomes

$$Q_i^t = \beta \mu_i^t + (1-\beta)\Delta I_i^t + \sigma_D \varepsilon_i^t,\ \varepsilon_i^t \sim \mathcal{N}(0,1). \tag{3.8}$$

The decision given by the heuristic at time t is

$$i_t = \arg\max_i Q_i^t.$$

For purposes of numerical implementation we scale both μ_i^t and ΔI_i^t by their maximum values at each time step:

$$\frac{\mu_i^t}{\max_j \mu_j^t}, \frac{\Delta I_i^t}{\max_j \Delta I_j^t}.$$

With this normalization, both deterministic elements of the value function are scaled to lie in $[0,1]$. The values of the decision noise parameter, σ_D, that need to be studied numerically also lie in $[0,1]$, since for cases $\sigma_D \geq 1$ the noise term dominates the deterministic terms in Q and decisions will be made primarily at random. The intermediate cases $\sigma_D \in [0,1]$ are the ones where the different components of the decision heuristic can all come into play, and as such are the ones of interest to be studied numerically.

The introduction of decision noise results in another tradeoff in addition to the explore-exploit tradeoff, this time between two different types of exploration: directed exploration

driven by the ΔI term which seeks information about the rewards, and random exploration driven by the $\sigma_D \varepsilon$ term. The following numerical example shows that these two terms can trade off in an interesting way.

3.4 A motivating numerical example

In this section, we motivate the role of parameters β and σ_D in the explore-exploit tradeoff. We study a numerical example using a reward structure previously used in human experiments, as discussed in [126] and Chapter 4 of [83]. This reward structure is designed such that an agent that carries out insufficient exploration is likely to get caught at a local maximum. If β is too high, the agent will pay excessive attention to immediate rewards μ_i^t and not seek enough information ΔI_i^t; however, the agent may be able to compensate by adding decision noise $\sigma_D \varepsilon_i^t$.

Consider a two-dimensional ($d = 2$) example with grid size $n = 10$, so there are $N = n^d = 100$ options. The reward surface is as shown in Figure 3.1: it has the characteristic that there is no gradient along the y direction, both ends along the x direction are local maxima, but the line $x = 10$ is the unique global maximum.

This reward surface intuitively requires exploratory behavior because it has two local maxima: if started on the left side of the domain, a simple gradient following algorithm will get stuck at the suboptimal local maximum. We choose the horizon of $T = 90$ time steps so that the agent can sample at most 90% of the space. The variance of the sampling noise is $\sigma_r^2 = 1/1200$ while the mean surface value is 0.25 so that the average signal-to-noise ratio is $0.25/\sigma_r \approx 8.66$.

The algorithm requires values of the priors $\boldsymbol{\mu}_0$ and Σ_0. For the means it is reasonable to set the uniform prior $\boldsymbol{\mu}_0 = \mathbf{0}$. The appropriate prior on covariance is less obvious. Following [68], we choose a prior that is exponential with a spatial length scale λ:

$$\Sigma_0(i, j) = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|/\lambda)$$

where $\|\mathbf{x}_i - \mathbf{x}_j\|$ is the 1-norm of the distance between points i and j. For the present example, we set $\lambda = 3$.

In order to understand the tradeoff between directed exploration and noise-based exploration, we computed via simulation the expected total rewards accumulated by the algorithm for $(\beta, \sigma_D) \in [0, 1] \times [10^{-5}, 10^{0.25}]$. The resolution of the set of simulations was 30 linearly-spaced points in β and 20 log-spaced points in σ_D, and for each pair of values (β, σ_D), the

expected value was computed by simulating 200 runs of the problem. For each simulation, the initial location of the agent was drawn from a uniform distribution.

Expected reward per time step as a function of the two parameters (β, σ_D) for this experiment is shown in Figure 3.2. As expected, some exploration was required to perform well in the task: in the deterministic decision limit $\sigma_D \to 0$, maximum rewards are achieved for a value of β of about 0.5. Comparison between Figures 3.1 and 3.2 shows that at the optimal tunings of the parameters, the expected rewards per time step of about 0.5 are near the value at the global optimum, so the algorithm is achieving near-optimal performance.

Furthermore, Figure 3.2 shows a tradeoff between weighting on directed exploration and random exploration. As σ_D increases, making action selection more random, one can maintain high performance by increasing β, thereby paying more attention to immediate rewards and reducing the weight on directed exploration.

We can develop a better understanding of the role of exploration by measuring it. The agent's trajectory $\{\mathbf{x}_{i_t} = (x(t), y(t)) | t = 1, \ldots, T\}$ forms a curve on the grid. We define a measure of exploration e_T over the T time steps by taking the variance of the time series representing this trajectory:

$$e_T = \frac{1}{T} \sum_{t=1}^{T} \left((x(t) - \bar{x})^2 + (y(t) - \bar{y})^2 \right), \tag{3.9}$$

where $\bar{x} = \frac{1}{T} \sum_{t=1}^{T} x(t)$ and $\bar{y} = \frac{1}{T} \sum_{t=1}^{T} y(t)$ are the average values of x and y. This measure has the physical interpretation of being the moment of inertia of the trajectory curve. It is bounded below by zero (representing an agent that does not move at all), and larger values of e_T correspond to more time being spent away from the average position.

Figure 3.3 plots e_T for the same set of parameters as in Figure 3.2. Again there is a tradeoff between β and σ_D: as random exploration is increased by increasing σ_D, a constant level of total exploration (as measured by e_T) can be maintained by increasing β, thereby paying more attention to immediate rewards and reducing the weight on directed exploration. The monotonic nature of the tradeoff is intuitive, although its specific shape is not trivial to explain.

Furthermore, the plots show that the level sets of e_T and expected reward have essentially the same structure. This strongly suggests that tuning β and σ_D has an effect by altering the overall level of exploration, and it is this overall level of exploration that governs performance.

The effects of β in the $\sigma_D \to 0$ deterministic decision case are also interesting. Although it is difficult to develop intuition for the effect of β in this case because of the large values of N and T, in the following section we derive analytical results for more tractable cases.

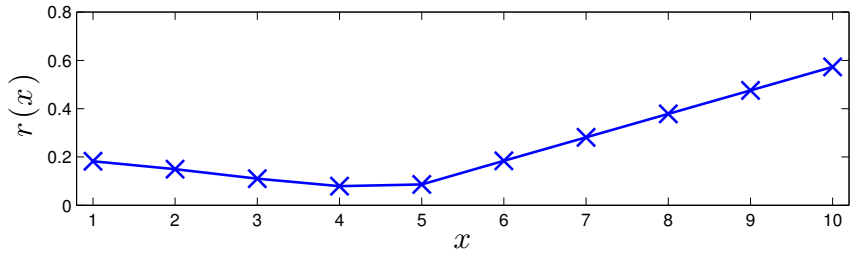


Figure 3.1: Profile of the mean reward surface for the numerical example. The grid points are at $x = 1, 2, \ldots, 10$. There is no gradient in the y direction, while in the x direction there is a local maximum at $x = 1$, a local minimum at $x = 4$, and a global maximum at $x = 10$. Previously published as Figure 1 of [100].

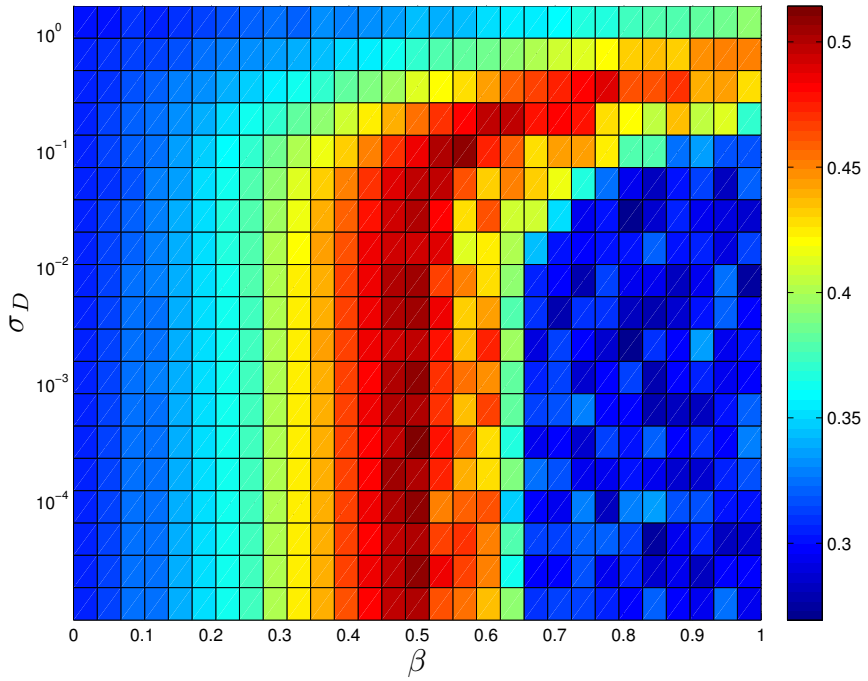


Figure 3.2: Expected reward per time step for various parameter values. Note the tradeoff between weighting on immediate reward β and decision noise σ_D. For small decision noise, expected rewards are highest for $\beta \approx 0.5$, but as noise increases one can maintain performance by increasing β. Previously published as Figure 2 of [100].

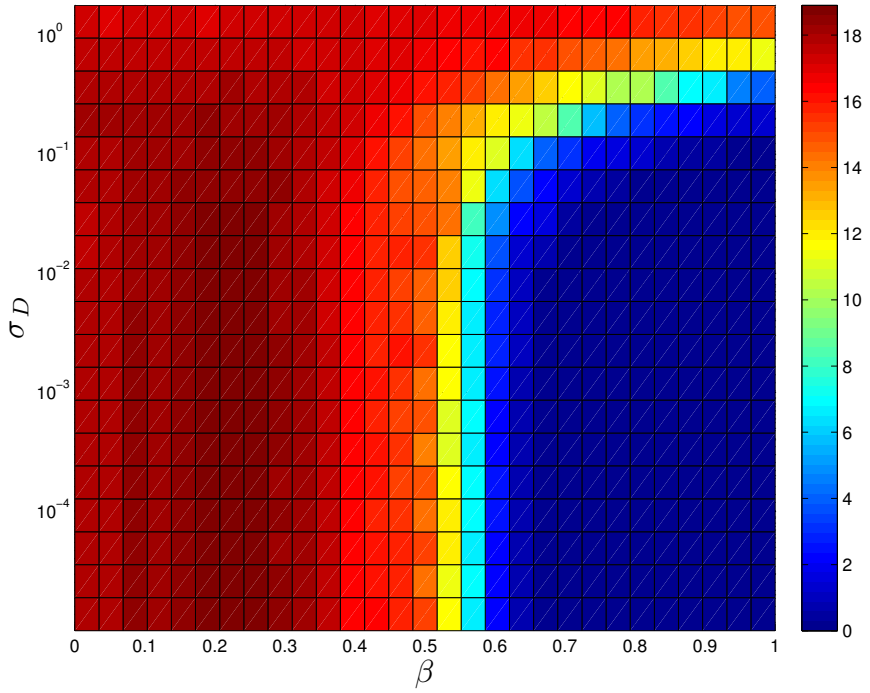


Figure 3.3: Exploration measure e_T (3.9) for the same parameter values as in Figure 3.2. Here the tradeoff between the two types of exploration is made clear: level sets of e_T represent sets of constant total exploration. As one increases random exploration through σ_D, one can maintain a constant level of total exploration by increasing β to decrease directed exploration. The level sets of e_T look very similar to the level sets of expected rewards, suggesting that it is the overall level of exploration that drives performance. Previously published as Figure 3 of [100].

3.5 Optimized heuristic and the role of β

As the previous example shows, the two parameters β and σ_D in the heuristic interact in a complex way to affect the performance of the algorithm. In this section we derive analytical results in the $\sigma_D = 0$ limit. The analysis provides insight into the role of β, and we can compute optimal tunings in the cases addressed. In Section 3.5.1 we analyze a low-dimensional case that yields key insights. In Section 3.5.2 we discuss generalizations to higher dimensions, other true distributions of rewards, and the $\sigma_D \neq 0$ case.

3.5.1 Analytical optimization of a low-dimensional case

To start, consider the $d = 1$ dimensional problem where $n = 2$, i.e. a grid with $N = 2$ options. Furthermore, let $\sigma_s = 0$ so there is no sampling noise and let $T = 2$ so the objective is simply $\max \mathbb{E}[r_1 + r_2]$. Let the true reward values $\mathbf{m}$ be jointly Gaussian distributed as

$$\mathbf{m} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ \bar{\mu}_2 \end{bmatrix}, \begin{bmatrix} 1 & \sigma\rho \\ \sigma\rho & \sigma^2 \end{bmatrix}\right).$$

Similarly, let the agent's prior over those values be the joint Gaussian distribution

$$\mathcal{N}(\boldsymbol{\mu}_0, \Sigma_0), \text{ where } \boldsymbol{\mu}_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \Sigma_0 = \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}.$$

Also, assume that $\rho \geq 0$ for convenience.

Note that in the case $\bar{\mu}_2 = 0$ and $\sigma = 1$ the prior is identical to the actual distribution of rewards, but in any other case they are distinct. The difference could be due to, e.g., measurement error in calibrating the prior or a change in the true statistics since the last time the agent was confronted with the problem.

We are interested in choosing the value of β for our heuristic that maximizes total expected rewards over all possible reward values and initial locations. We assume that the agent can begin in either of the two locations with equal probability, so $\mathbb{E}[r_1] = \bar{\mu}_2/2$ independent of β. Therefore the optimization problem reduces to

$$\tilde{\beta} = \arg\max_{\beta} \max_{\mathbf{x}_2} \mathbb{E}[r_2 | r_1]. \tag{3.10}$$

That is, given r_1, the algorithm has to decide whether to stay in its current location or to switch to the alternative location. This is a well-studied problem in signal detection theory

(see, e.g., [50, Chapter 12] or Example II.B.2 of [89]). The optimal β maximizes the expected payoffs of the decision made by the algorithm.

The detection theory solution consists of setting a threshold $\tilde{m}$ on the observed reward r_1 and switching if $r_1 < \tilde{m}$. The optimal threshold is a function of the prior beliefs about $\mathbf{m}$ and the costs associated with each decision. If the agent is equally likely to be in either initial location, the optimal threshold is

$$\tilde{m} = \frac{1}{2}(0 + \bar{\mu}_2) = \bar{\mu}_2/2. \tag{3.11}$$

We show that the optimal tuning of our algorithm reduces to the optimal solution of the detection problem.

We begin by computing the expected value of the decision made by the algorithm for a given value of β. At time $t = 1$, the agent picks $i_1 = 1$ or 2, each with probability 1/2, and observes either m_1 or m_2, respectively. In either case the observed reward m_{i_1} is now known with certainty, so its inferred value is $\mu^1_{i_1} = m_{i_1}$ and the inferred value of the unobserved reward m_j is $\mu^1_j = \rho m_i$. Similarly, $\Lambda_0 = \Sigma_0^{-1} = \frac{1}{1-\rho^2}\begin{bmatrix} 1 & -\rho \\ -\rho & 1 \end{bmatrix}$. The two minors M_{ii} are both equal to $1/(1-\rho^2)$, so $\Delta I^1_i = 0$ for the observed location and $\Delta I^1_j = M_{jj} = \frac{1}{1-\rho^2}$ for the unobserved location. For a given value of β, the expected value in the optimization problem (3.10) is the average of the expected values $\mathbb{E}[r_2|r_1]$ for the two cases $\mathbf{x}_{i_1} = 1, 2$.

We proceed by computing the expected value of the algorithm for the case of starting in location 1, so $\mathbf{x}_{i_1} = 1$. In this case the heuristic function Q^t_i (3.8) (with $\sigma_D = 0$) takes the following values:

$$Q^1_1 = \beta m_1,\ Q^1_2 = \beta \mu_{2,1} + \frac{1-\beta}{1-\rho^2} = \beta\rho m_1 + \frac{1-\beta}{1-\rho^2}.$$

The algorithm picks the maximum of $\{Q^1_1, Q^1_2\}$ and switches if $Q^1_2 > Q^1_1$, or equivalently if $\beta\rho m_1 + \frac{1-\beta}{1-\rho^2} > \beta m_1$. This is equivalent to setting a threshold value $\tilde{m}$ and switching if

$$r_1 = m_1 < \tilde{m} = \frac{1-\beta}{\beta}\frac{1}{(1-\rho)(1-\rho^2)}, \tag{3.12}$$

which sets a threshold $\tilde{m}$ as a function of β and ρ.

If the algorithm decides to switch locations, the agent will then obtain the reward $r_2 = m_2$. Otherwise it stays in the original location and receives $r_2 = m_1$. The expected value of r_2 given the algorithm's decision is then given by

$$\mathbb{E}[r_2|\mathbf{x}_{i_1} = 1] = \mathbb{E}[m_1|m_1 \geq \tilde{m}] + \mathbb{E}[m_2|m_1 < \tilde{m}].$$

Since m_1 and m_2 are jointly Gaussian, this expectation is analytically tractable and is equal to

$$\phi(\tilde{m}) + \bar{\mu}_2\Phi(\tilde{m}) - \rho\sigma\phi(\tilde{m}) = \bar{\mu}_2\Phi(\tilde{m}) + (1-\rho\sigma)\phi(\tilde{m}),$$

where $\phi(z)$ and $\Phi(z)$ are the the pdf and cdf, respectively, of the standard normal distribution.

In the case where the agent's initial location is $\mathbf{x}_{i_1} = 2$, the agent observes $r_1 = m_2$. The function Q_i^t takes the following values:

$$Q_1^1 = \beta\mu_1^1 + \frac{1-\beta}{1-\rho^2} = \beta\rho m_2 + \frac{1-\beta}{1-\rho^2}, Q_2^1 = \beta m_2.$$

This is symmetric to the case $\mathbf{x}_{i_1} = 1$ under interchange of $i_1 = 1$ and $i_1 = 2$ because of the symmetry of the prior. Again, the algorithm switches to the alternate location if $Q_1^1 > Q_2^1$, or

$$r_1 = m_2 < \tilde{m} = \frac{1-\beta}{\beta}\frac{1}{(1-\rho)(1-\rho^2)},$$

where the threshold m^* is the same as above, again due to the symmetry of the prior. The expected value of r_2 given the algorithm's decision is

$$\mathbb{E}\left[r_2|\mathbf{x}_{i_1} = 2\right] = \mathbb{E}\left[m_2|m_2 \geq \tilde{m}\right] + \mathbb{E}\left[m_1|m_2 < \tilde{m}\right].$$

This expectation can again be expressed in closed form, and takes the value

$$\bar{\mu}_2\left(1 - \Phi\left(\frac{\tilde{m}-\bar{\mu}_2}{\sigma}\right)\right) + \sigma(1-\rho\sigma)\phi\left(\frac{\tilde{m}-\bar{\mu}_2}{\sigma}\right).$$

Since $\mathbf{x}_{i_1} = 1$ or 2 with equal probability, for a given threshold $\tilde{m}$, the expected value in the optimization problem (3.10) is the simple average of the expected reward for each initial position $\mathbb{E}\left[r_2|\mathbf{x}_{i_1} = 1\right]$ and $\mathbb{E}\left[r_2|\mathbf{x}_{i_1} = 2\right]$:

$$\mathbb{E}[r_2|r_1] = \frac{1}{2}\left[\bar{\mu}_2\left(1 + \Phi(\tilde{m}) - \Phi\left(\frac{\tilde{m}-\bar{\mu}_2}{\sigma}\right)\right) + (1-\rho\sigma)\phi(\tilde{m}) + \sigma(1-\rho\sigma)\phi\left(\frac{\tilde{m}-\bar{\mu}_2}{\sigma}\right)\right].$$

The parameter ρ is fixed, so the optimization (3.10) reduces to picking the value $\beta = \tilde{\beta}$ that results in the threshold $\tilde{m}$ that maximizes $\mathbb{E}\left[r_2|r_1\right]$. The expression for $\mathbb{E}\left[r_2|r_1\right]$ is somewhat unwieldy, but several cases are informative.

First, consider the case $\bar{\mu}_2 = 0, \sigma = 1$, which is the case where the prior is equal to the actual distribution. In this case the expectation reduces to

$$\mathbb{E}\left[r_2|r_1\right] = (1-\rho)\phi(\tilde{m}).$$

We want to pick the value of β that maximizes this expectation, which means maximizing $\phi(\tilde{m})$ since $1-\rho$ is fixed. If $\rho = 1$ the expected rewards are zero independent of $\tilde{m}$, so consider cases $\rho < 1$. The function $\phi(z)$ takes its unique maximum at $z = 0$, so we set the threshold $\tilde{m} = 0$. Equation (3.12) then implies that the optimal value of β is $\beta^* = 1$, so the optimal tuning of the heuristic is

$$Q_i^t = \mu_i^t.$$

In this case the optimal tuning of the algorithm is pure exploit and no explore. The heuristic ignores the information gain component ΔI and only weights inferred rewards $\boldsymbol{\mu}$. The threshold is set equal to 0, cf. Equation (3.11) where $\bar{\mu}_2 = 0$. This is identical to the standard optimal detection theory result [89], and the heuristic only weights μ_i^t because in this case the linear inference model is optimal. In this case the heuristic is not particularly beneficial, and setting β to anything less than one is suboptimal. However, we show next that the heuristic provides robustness in cases where the field statistics are not known perfectly.

Consider the case above with $\sigma = 1$ but $\bar{\mu}_2 \neq 0$, so the prior is correct except for the mean value $\bar{\mu}_2$. In this case the inference is no longer optimal, so neither is weighting only the inferred reward. The expected reward $\mathbb{E}[r_2|r_1]$ is

$$\frac{1}{2}\left[\bar{\mu}_2\left(1+\Phi(\tilde{m})-\Phi\left(\tilde{m}-\bar{\mu}_2\right)\right)+(1-\rho)\left(\phi(\tilde{m})+\phi(\tilde{m}-\bar{\mu}_2)\right)\right].$$

For any given $\bar{\mu}_2$ and ρ, the expectation can be maximized with respect to the threshold $\tilde{m}$, and in general the optimal threshold is non-zero. For example, if $\bar{\mu}_2 = 1, \rho = 0.5$, the maximum occurs at $\tilde{m} = 0.5$, or $\tilde{\beta} = 16/19 \approx 0.84$. If, instead, $\bar{\mu}_2 = -1, \rho = 0.5$, the maximum occurs at $\tilde{m} = -0.5$, or $\tilde{\beta} = 16/13 \approx 1.23$. Again, the optimal threshold in both cases is $\tilde{m} = \bar{\mu}_2/2$, as in the detection theory solution. This shows how setting $\beta \neq 1$ provides robustness by helping the algorithm recover the optimal threshold in the face of suboptimal inference.

3.5.2 Discussion

The results in the previous section make intuitive sense because in the case where the true distribution $\mathcal{D}$ is Gaussian and the prior statistics are correctly calibrated, the inference model is optimal. In that case the inferred value term μ_i^t is the optimal expected value of the option i at time t, and the optimal action at the terminal time $t = T = 2$ is simply to pick the maximum of the μ_i^t, so the optimal β reflects that and is equal to one.

If, however, the true distribution $\mathcal{D}$ is not Gaussian or the prior statistics are incorrect, the inference model will be suboptimal. If the world is "better" than expected by the prior,

as in the case where $\bar{\mu}_2 = 1$, setting $\beta < 1$ provides robustness by encouraging exploration, whereas if it is "worse", as in the case where $\bar{\mu}_2 = -1$, setting $\beta > 1$ provides robustness by weighting expected rewards more highly and discouraging exploration.

This suggests the form of a simple feedback control law for β: at each time step, if the world appears "better" than implied by the prior, decrease β to encourage guided exploration. If, instead, the world appears "worse", increase β to discourage it. At time t, an estimate p_t of the degree to which the world is "better" or "worse" could be made, e.g., by the mean difference between the inferred rewards at the current and previous time steps:

$$p_t = \frac{1}{N} \sum_{i=1}^{N} \left(\mu_i^t - \mu_i^{t-1} \right).$$

Then if the inferred values μ_i^t are increasing, the world appears to be "better" than expected and $p_t > 0$. Furthermore, since the field is stationary, the inference is getting monotonically more accurate in time, so $p_t \to 0$ as $t \to \infty$. Then, setting $K > 0$ in the proportional control law $\beta_t = \beta_{t-1} - K p_t$ biases β in the desired direction.

As we saw in the previous section, the optimal value of β is dependent on the agent's model of the world (i.e., prior). If the model is wrong, then the optimal value of β will be wrong as well, which makes the algorithm's performance sensitive to model errors. We would like to reduce this sensitivity in order to make the algorithm more robust. The above feedback control law is one way to do so by tuning β. The introduction of decision noise $\sigma_D > 0$ is another way to do so, since the decision noise acts as a model-free mechanism for exploration. The gains in robustness can be seen in Figure 3.2 by comparing the values of expected reward along the lines $\sigma_D = 0$ and $\sigma_D = 10^{-0.5}$. The range of values of β for which the expected reward is high is larger for $\sigma_D = 10^{-0.5}$ than for $\sigma_D = 0$, meaning that the algorithm's performance is less sensitive to the tuning of β.

3.6 Conclusions

In this chapter we have presented a heuristic that was developed to describe human behavior in a simple explore-exploit task. The heuristic includes two forms of exploratory behavior: directed exploration, guided by seeking information about rewards, and random exploration, provided by random noise. We use this heuristic to construct an algorithm to solve explore-exploit problems in spatially distributed scalar fields. The algorithm uses an optimal Bayesian inference algorithm for building beliefs about the field, and then applies the heuristic to solve the decision problem of which location to visit next.

Using a numerical example, we show that the two types of exploratory behavior trade off in an interesting way, but that both influence an overall level of exploration which, when measured, is shown to strongly correlate with task performance. In particular, in the case where there is no random exploration, we show that there is a level of directed exploration that produces optimal performance in the task.

To gain intuition for the role of the level of directed exploration in the case without random exploration, we consider an example problem where the field is distributed over two points. We show that in the case where the inference is optimal, the optimal tuning of the heuristic is to put full weight on expected rewards at the expense of all directed exploration; in this case the heuristic reduces to an optimal Bayesian detector. However, in the general case where the inference is not optimal, for example if it was given incorrect field statistics, including some directed exploration provides robustness to modeling errors.

The analytical result gives intuition into the signs of sensitivities to parameters, such as the prior parameters $\boldsymbol{\mu}_0$ and Σ_0 and the decision parameter β. The analysis enabled us to find the exact optimal tuning of the heuristic for a simple case of the multi-armed bandit problem. However, it is difficult to extend these methods to find the optimal $\tilde{\beta}$ in a more general case, in particular with a longer horizon $T > 2$ and incorporating sampling and decision noise $\sigma_s, \sigma_D > 0$. In the following chapter we consider a slightly weaker notion of optimality in terms of the growth rate of cumulative expected regret. We develop a new algorithm that again has an interpretation as an ambiguity bonus heuristic but that allows us to find parameter tunings that result in optimal performance in terms of cumulative expected regret.

Chapter 4

The Upper Credible Limit (UCL) algorithms for Gaussian multi-armed bandits[1]

"A mathematician, then, will be defined in what follows as someone who has published the proof of at least one non-trivial theorem."

(Jean Dieudonné)

In this chapter, we construct the UCL algorithm, a Bayesian UCB algorithm that captures the features of human decision-making described in Section 2.4 above. This algorithm can be interpreted similarly to the ambiguity bonus algorithm presented in the previous chapter, but allows for more rigorous analysis, in particular performance guarantees. We begin with the case of deterministic decision-making and show that for an uninformative prior the resulting algorithm achieves logarithmic regret, i.e., optimal finite-time performance. We then extend the algorithm to the case of stochastic decision-making using a Boltzmann (or softmax) decision rule, and show that there exists a feedback rule for the temperature of the Boltzmann distribution such that the stochastic algorithm achieves logarithmic regret. In both cases we first consider uncorrelated priors and then extend to correlated priors.

[1]This chapter is adapted from Section IV of [99], from which Sections 4.1–4.4 in this chapter are mostly taken verbatim. An abbreviated version of this material, including Theorems 4.1, 4.2, and 4.7 from this chapter, appeared in the conference paper [98].

4.1 The deterministic UCL algorithm with uncorrelated priors

Let the prior on m_i, the mean reward at arm i, be a Gaussian random variable with mean μ_i^0 and variance σ_0^2. In symbols, the agent's prior on m_i is $\mathcal{N}(\mu_i^0, \sigma_0^2)$. This is identical to the prior assumed for the ambiguity bonus algorithm in Section 3.3.1 with $(\boldsymbol{\mu}_0)_i = \mu_i^0$ and $\Sigma_0 = \sigma_0^2 I_N$, where I_N is the N-dimensional identity matrix. Since I_N is a diagonal matrix, the prior on each m_i is uncorrelated with that on all the other $m_j, j \neq i$, and the prior $\mathcal{N}(\boldsymbol{\mu}_0, \Sigma_0)$ is referred to as *uncorrelated*. We are particularly interested in the case of an uninformative prior, i.e., $\sigma_0^2 \to +\infty$. Let the number of times arm i has been selected until time t be denoted by n_i^t. Let the empirical mean of the rewards from arm i until time t be $\bar{m}_i^t$.

Recall that the reward sampling distribution for each arm i is Gaussian with unknown mean m_i and known variance σ_s^2, which is assumed to be the same for all arms. Conditioned on the number of visits n_i^t to arm i and the empirical mean $\bar{m}_i^t$, the mean reward at arm i at time t is a Gaussian random variable (M_i) with mean and variance

$$\mu_i^t := \mathbb{E}[M_i | n_i^t, \bar{m}_i^t] = \frac{\delta^2 \mu_i^0 + n_i^t \bar{m}_i^t}{\delta^2 + n_i^t}, \text{ and}$$
$$\left(\sigma_i^t\right)^2 := \mathrm{Var}[M_i | n_i^t, \bar{m}_i^t] = \frac{\sigma_s^2}{\delta^2 + n_i^t},$$

respectively, where σ_s^2 is the known reward sampling variance and $\delta^2 = \sigma_s^2/\sigma_0^2$. The quantity σ_0^2 can be interpreted as a measure of the certainty of the prior, so δ^2 can be interpreted as a certainty measure normalized with respect to the uncertainty in the reward sampling process. An uninformative prior, i.e., $\delta^2 \to 0^+$, corresponds to complete uncertainty about the means m_i, while $\delta^2 = 1$ corresponds to a moderately informative prior where the two uncertainties have equal magnitudes. Moreover,

$$\mathbb{E}[\mu_i^t | n_i^t] = \frac{\delta^2 \mu_i^0 + n_i^t m_i}{\delta^2 + n_i^t} \text{ and } \mathrm{Var}[\mu_i^t | n_i^t] = \frac{n_i^t \sigma_s^2}{(\delta^2 + n_i^t)^2},$$

because the sample mean $\bar{m}_i^t$ is a Gaussian random variable with mean m_i and variance σ_s^2/n_i^t.

We now propose the UCL algorithm for the Gaussian multi-armed bandit problem. At each decision instance $t \in \{1, \ldots, T\}$, the UCL algorithm selects an arm with the maximum value of the upper limit of the smallest $(1 - 1/Kt)$-credible interval, i.e., it selects an arm

$i_t = \text{argmax}\{Q_i^t \mid i \in \{1, \ldots, N\}\}$, where

$$Q_i^t = \mu_i^t + \sigma_i^t \Phi^{-1}(1 - 1/Kt), \tag{4.1}$$

$\Phi^{-1} : (0, 1) \to \mathbb{R}$ is the inverse cumulative distribution function for the standard Gaussian random variable, and $K \in \mathbb{R}_{>0}$ is a tunable parameter. For an explicit pseudocode implementation, see Algorithm 1 in Appendix A. In the following, we will refer to Q_i^t as the $(1 - 1/Kt)$-*upper credible limit* (UCL).

It is known [57, 116] that an efficient policy to maximize the total information gained over sequential sampling of options is to pick the option with highest variance $(\sigma_i^t)^2$ at each time t. Thus, Q_i^t is the weighted sum of the expected gain in the total reward μ_i^t (exploitation) and the gain in the total information about arms σ_i^t (exploration) if arm i is picked at time t. In terms of the ambiguity bonus heuristic (3.1) from the previous chapter, $\Delta I_i^t = \sigma_i^t$ is the information gain component and $A = \Phi^{-1}(1 - 1/Kt)$ is the ambiguity bonus, which is now a function of decision time t rather than a constant. The following analysis allows us to show that this functional form for ΔI_i^t and A results in optimal performance.

4.2 Regret analysis of the deterministic UCL algorithm

In this section, we analyze the performance of the UCL algorithm. We first derive bounds on the inverse cumulative distribution function of the standard Gaussian random variable and then utilize them to derive upper bounds on the cumulative expected regret for the UCL algorithm. We state the following theorem about bounds on the inverse Gaussian cdf.

Theorem 4.1 (***Bounds on the inverse Gaussian cdf***)**.** *The following bounds hold for the inverse cumulative distribution function of the standard Gaussian random variable for each* $\alpha \in (0, 1/\sqrt{2\pi})$*, and any* $\beta \geq 1.02$*:*

$$\Phi^{-1}(1 - \alpha) < \beta\sqrt{-\log(-(2\pi\alpha^2)\log(2\pi\alpha^2))}, \text{ and} \tag{4.2}$$

$$\Phi^{-1}(1 - \alpha) > \sqrt{-\log(2\pi\alpha^2(1 - \log(2\pi\alpha^2)))}. \tag{4.3}$$

Proof. We start by establishing inequality (4.2). It suffices to establish this inequality for $\beta = 1.02$. Since the cumulative distribution function of the standard normal random variable is a continuous and monotonically increasing function, it suffices to show that

$$\Phi(\beta\sqrt{-\log(-2\pi\alpha^2\log(2\pi\alpha^2))}) + \alpha - 1 \geq 0, \tag{4.4}$$

for each $\alpha \in (0,1)$. Equation (4.4) can be equivalently written as $h(x) \geq 0$, where $x = 2\pi\alpha^2$ and $h : (0,1) \to (0, 1/\sqrt{2\pi})$ is defined by

$$h(x) = \Phi(\beta\sqrt{-\log(-x\log x)})) + \frac{\sqrt{x}}{\sqrt{2\pi}} - 1.$$

Note that $\lim_{x\to 0^+} h(x) = 0$ and $\lim_{x\to 1^-} h(x) = 1/\sqrt{2\pi}$. Therefore, to establish the theorem, it suffices to establish that h is a monotonically increasing function. It follows that

$$g(x) := 2\sqrt{2\pi}h'(x) = \frac{1}{\sqrt{x}} + \frac{\beta(-x\log x)^{\beta^2/2-1}(1+\log x)}{\sqrt{-\log(-x\log x)}}.$$

Note that $\lim_{x\to 0^+} g(x) = +\infty$ and $\lim_{x\to 1^-} g(x) = 1$. Therefore, to establish that h is monotonically increasing, it suffices to show that g is non-negative for $x \in (0,1)$. This is the case if the following inequality holds:

$$g(x) = \frac{1}{\sqrt{x}} + \frac{\beta(-x\log x)^{\beta^2/2-1}(1+\log x)}{\sqrt{-\log(-x\log x)}} \geq 0,$$

which holds if

$$\frac{1}{\sqrt{x}} \geq -\frac{\beta(-x\log x)^{\beta^2/2-1}(1+\log x)}{\sqrt{-\log(-x\log x)}}.$$

The inequality holds if the right hand side is negative. If it is positive, one can take the square of both sides and the inequality holds if

$$\begin{aligned} -\log(-x\log x) &\geq \beta^2 x(1+\log x)^2(-x\log x)^{\beta^2-2} \\ &= \beta^2 x(1+2\log x + (\log x)^2)(-x\log x)^{\beta^2-2}. \end{aligned}$$

Letting $t = -\log x$, the above inequality transforms to

$$-\log(te^{-t}) \geq \beta^2 e^{-t}(1-2t+t^2)(te^{-t})^{\beta^2-2},$$

which holds if

$$-\log t \geq \beta^2 t^{\beta^2-2}(1-2t+t^2)e^{-(\beta^2-1)t} - t.$$

Dividing by t, this is equivalent to

$$-\frac{\log t}{t} \geq \beta^2 t^{\beta^2-3}(1-2t+t^2)e^{-(\beta^2-1)t} - 1,$$

which is true if

$$\inf_{t\in[1,+\infty)} -\frac{\log t}{t} \geq \max_{t\in[1,+\infty)} \beta^2 t^{\beta^2-3}(1-2t+t^2)e^{-(\beta^2-1)t} - 1. \tag{4.5}$$

The extrema in (4.5) can be calculated analytically, so we have

$$\inf_{t\in[1,+\infty)} -\frac{\log t}{t} = -\frac{1}{e} \approx -0.3679$$

for the left hand side of (4.5) and

$$\begin{aligned} t^* &= \arg\max_{t\in[1,+\infty)} \beta^2 t^{\beta^2-3}(1-2t+t^2)e^{-(\beta^2-1)t} - 1 \\ &= 1 + \sqrt{2/(\beta^2-1)} \\ \implies &\max_{t\in[1,+\infty)} \beta^2 t^{\beta^2-3}(1-2t+t^2)e^{-(\beta^2-1)t} - 1 \approx -0.3729, \end{aligned}$$

for the right hand side of (4.5). Therefore, (4.5) holds. In consequence, $g(x)$ is non-negative for $x \in (0,1)$, $h(x)$ is a monotonically increasing function. This establishes inequality (4.2). Inequality (4.3) follows analogously. □

The bounds in equations (4.2) and (4.3) were conjectured by Fan [36] without the factor β. In fact, it can be numerically verified that without the factor β, the conjectured upper bound is incorrect. We present a visual depiction of the tightness of the derived bounds in Figure 4.1.

We now analyze the performance of the UCL algorithm. Recall from Equation (2.3) that $\Delta_i = m_{i^*} - m_i$ is the expected regret due to picking arm i and that $R_t = \Delta_{i_t}$ is the expected regret due to picking arm i_t. We define $\{R_t^{\text{UCL}}\}_{t\in\{1,\ldots,T\}}$ as the sequence of expected regret for the UCL algorithm. The UCL algorithm achieves logarithmic regret uniformly in time (i.e., number of decisions T) as formalized in the following theorem. We use the phrase "uniformly in time" as in Auer *et al.* [9] to contrast with the asymptotic results due to, e.g., Lai and Robbins [62].

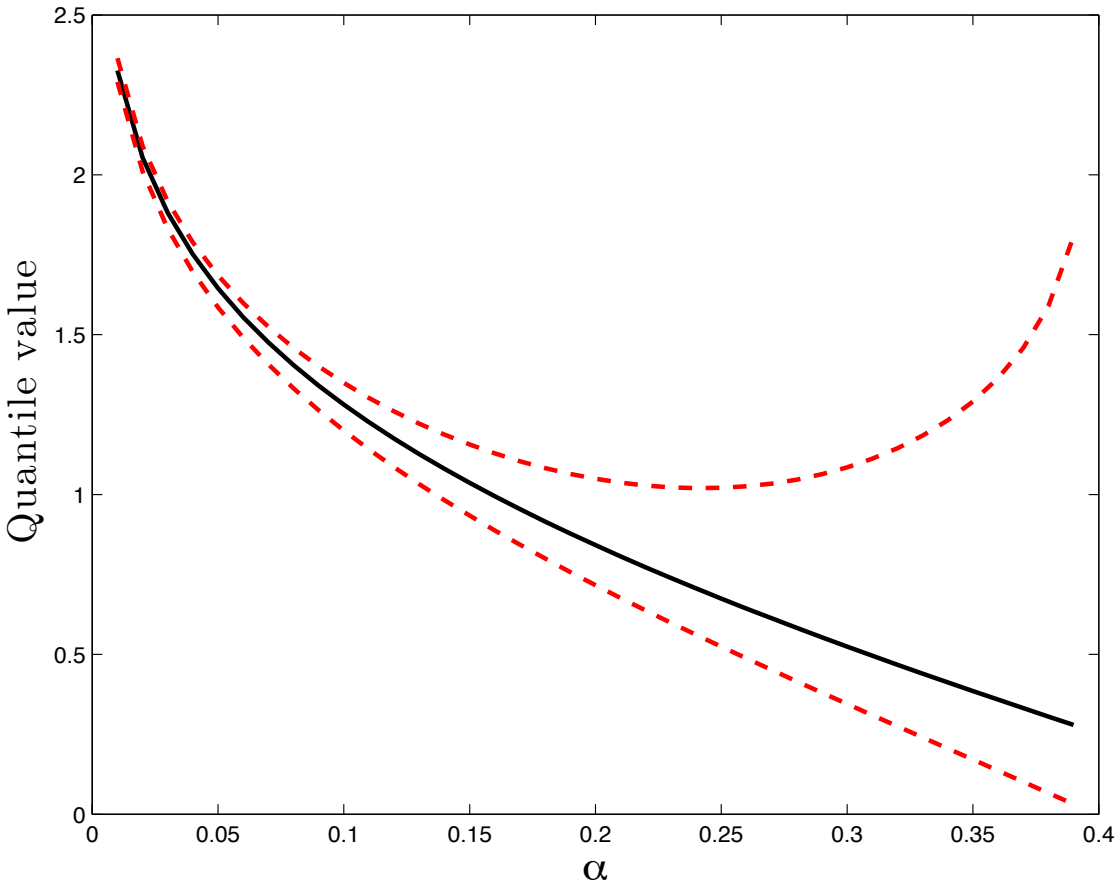


Figure 4.1: Depiction of the normal quantile function $\Phi^{-1}(1-\alpha)$ (solid line) and the bounds (4.2) and (4.3) (dashed lines), with $\beta = 1.02$. Previously published as Figure 4 of [99].

Theorem 4.2 (***Regret of the deterministic UCL algorithm***)**.** *The following statements hold for the Gaussian multi-armed bandit problem and the deterministic UCL algorithm with uncorrelated uninformative prior and* $K = \sqrt{2\pi e}$*:*

1. *the expected number of times a suboptimal arm* i *is chosen until time* T *satisfies*

$$\mathbb{E}\left[n_i^T\right] \leq \Big(\frac{8\beta^2\sigma_s^2}{\Delta_i^2} + \frac{2}{\sqrt{2\pi e}}\Big)\log T + \frac{4\beta^2\sigma_s^2}{\Delta_i^2}(1 - \log 2 - \log\log T) + 1 + \frac{2}{\sqrt{2\pi e}}\,;$$

2. *the cumulative expected regret until time* T *satisfies*

$$\sum_{t=1}^{T} R_t^{\mathrm{UCL}} \leq \sum_{i=1}^{N} \Delta_i \left(\Big(\frac{8\beta^2\sigma_s^2}{\Delta_i^2} + \frac{2}{\sqrt{2\pi e}}\Big)\log T + \frac{4\beta^2\sigma_s^2}{\Delta_i^2}(1 - \log 2 - \log\log T) + 1 + \frac{2}{\sqrt{2\pi e}} \right).$$

Proof. We start by establishing the first statement. In the spirit of [9], we bound n_i^T as follows:

$$\begin{aligned} n_i^T &= \sum_{t=1}^{T} \mathbf{1}(i_t = i) \\ &\leq \sum_{t=1}^{T} \mathbf{1}\big(Q_i^t > Q_{i^*}^t\big) \\ &\leq \eta + \sum_{t=1}^{T} \mathbf{1}\Big(Q_i^t > Q_{i^*}^t, n_i^{(t-1)} \geq \eta\Big)\,, \end{aligned}$$

where η is some positive integer and $\mathbf{1}(x)$ is the indicator function, with $\mathbf{1}(x) = 1$ if x is a true statement and 0 otherwise.

At time t, the agent picks option i over i^* only if

$$Q_{i^*}^t \leq Q_i^t.$$

This is true when at least one of the following equations holds:

$$\mu_{i^*}^t \leq m_{i^*} - C_{i^*}^t \tag{4.6}$$

$$\mu_i^t \geq m_i + C_i^t \tag{4.7}$$

$$m_{i^*} < m_i + 2C_i^t \tag{4.8}$$

where $C_i^t = \frac{\sigma_s}{\sqrt{\delta^2+n_{it}}}\Phi^{-1}(1-\alpha_t)$ and $\alpha_t = 1/Kt$. Otherwise, if none of the equations (4.6)-(4.8) holds,

$$Q_{i^*}^t = \mu_{i^*}^t + C_{i^*}^t > m_{i^*} \geq m_i + 2C_i^t > \mu_i^t + C_i^t = Q_i^t,$$

and option i^* is picked over option i at time t.

We proceed by analyzing the probability that Equations (4.6) and (4.7) hold. Note that the empirical mean $\bar{m}_i^t$ is a normal random variable with mean m_i and variance σ_s^2/n_i^t, so, conditional on n_i^t, μ_i^t is a normal random variable distributed as

$$\mu_i^t \sim \mathcal{N}\left(\frac{\delta^2\mu_i^0 + n_i^t m_i}{\delta^2 + n_i^t}, \frac{n_i^t\sigma_s^2}{(\delta^2+n_i^t)^2}\right).$$

Equation (4.6) holds if

$$\begin{aligned}
& m_{i^*} \geq \mu_{i^*}^t + \frac{\sigma_s}{\sqrt{\delta^2+n_i^t}}\Phi^{-1}(1-\alpha_t) \\
\iff & m_{i^*} - \mu_{i^*}^t \geq \frac{\sigma_s}{\sqrt{\delta^2+n_i^t}}\Phi^{-1}(1-\alpha_t) \\
\iff & z \leq -\sqrt{\frac{n_{i^*}^t+\delta^2}{n_{i^*}^t}}\Phi^{-1}(1-\alpha_t) + \frac{\delta^2}{\sigma_s}\frac{\Delta m_{i^*}}{\sqrt{n_{i^*}^t}},
\end{aligned}$$

where $z \sim \mathcal{N}(0,1)$ is a standard normal random variable and $\Delta m_{i^*} = m_{i^*} - \mu_{i^*}^0$. For an uninformative prior $\delta^2 \to 0^+$, and consequently, equation (4.6) holds if and only if $z \leq -\Phi(1-\alpha_t)$. Therefore, for a uninformative prior,

$$\mathbb{P}\left(\text{Equation (4.6) holds}\right) = \alpha_t = \frac{1}{Kt} = \frac{1}{\sqrt{2\pi e t}}.$$

Similarly, Equation (4.7) holds if

$$\begin{aligned}
& m_i \leq \mu_i^t - \frac{\sigma_s}{\sqrt{\delta^2+n_i^t}}\Phi^{-1}(1-\alpha_t) \\
\iff & \mu_i^t - m_i \geq \frac{\sigma_s}{\sqrt{\delta^2+n_i^t}}\Phi^{-1}(1-\alpha_t) \\
\iff & z \geq \sqrt{\frac{n_i^t+\delta^2}{n_i^t}}\Phi^{-1}(1-\alpha_t) + \frac{\delta^2}{\sigma_s}\frac{\Delta m_i}{\sqrt{n_i^t}},
\end{aligned}$$

where $z \sim \mathcal{N}(0,1)$ is a standard normal random variable and $\Delta m_i = m_i - \mu_i^0$. The analogous argument to that for the above case shows that, for an uninformative prior,

$$\mathbb{P}\left(\text{Equation (4.7) holds}\right) = \alpha_t = \frac{1}{Kt} = \frac{1}{\sqrt{2\pi e t}}.$$

Equation (4.8) holds if

$$\begin{aligned}
& m_{i^*} < m_i + \frac{2\sigma_s}{\sqrt{\delta^2 + n_i^t}} \Phi^{-1}(1-\alpha_t) \\
\iff & \Delta_i < \frac{2\sigma_s}{\sqrt{\delta^2 + n_i^t}} \Phi^{-1}(1-\alpha_t) \\
\iff & \frac{\Delta_i^2}{4\beta^2\sigma_s^2}(\delta^2 + n_i^t) < -\log(-2\pi\alpha_t^2 \log(2\pi\alpha_t^2)) \qquad (4.9) \\
\implies & \frac{\Delta_i^2}{4\beta^2\sigma_s^2}(\delta^2 + n_i^t) < \log(et^2) - \log\log(et^2) \\
\implies & \frac{\Delta_i^2}{4\beta^2\sigma_s^2}(\delta^2 + n_i^t) < \log(eT^2) - \log\log(eT^2) \qquad (4.10) \\
\implies & \frac{\Delta_i^2}{4\beta^2\sigma_s^2}(\delta^2 + n_i^t) < 1 + 2\log T - \log 2 - \log\log T
\end{aligned}$$

where $\Delta_i = m_{i^*} - m_i$, the inequality (4.9) follows from the bound (4.2), and the inequality (4.10) follows from the monotonicity of the function $\log x - \log\log x$ in the interval $[e, +\infty)$. Therefore, for an uninformative prior, inequality (4.8) never holds if n_i^t obeys

$$n_i^t \geq \frac{4\beta^2\sigma_s^2}{\Delta_i^2}(1 + 2\log T - \log 2 - \log\log T).$$

Setting $\eta = \lceil \frac{4\beta^2\sigma_s^2}{\Delta_i^2}(1 + 2\log T - \log 2 - \log\log T) \rceil$, we get

$$\begin{aligned}
\mathbb{E}\left[n_i^T\right] &\leq \eta + \sum_{t=1}^{T} \mathbb{P}\left(Q_i^t > Q_{i^*}^t, n_i^{(t-1)} \geq \eta\right) \\
&= \eta + \sum_{t=1}^{T} \mathbb{P}\left(\text{Equation (4.6) holds}, n_i^{(t-1)} \geq \eta\right) \\
&\qquad + \sum_{t=1}^{T} \mathbb{P}\left(\text{Equation (4.7) holds}, n_i^{(t-1)} \geq \eta\right)
\end{aligned}$$

Substituting in the value of η, we arrive at the bound

$$\mathbb{E}\left[n_i^T\right] < \frac{4\beta^2\sigma_s^2}{\Delta_i^2}(1 + 2\log T - \log 2 - \log\log T) + 1 + \frac{2}{\sqrt{2\pi e}}\sum_{t=1}^{T}\frac{1}{t}.$$

The sum over t in this last equation can be bounded by the integral

$$\sum_{t=1}^{T}\frac{1}{t} \leq 1 + \int_1^T \frac{1}{t}\mathrm{d}t = 1 + \log T,$$

yielding the bound in the first statement

$$\mathbb{E}\left[n_i^T\right] \leq \Big(\frac{8\beta^2\sigma_s^2}{\Delta_i^2} + \frac{2}{\sqrt{2\pi e}}\Big)\log T + \frac{4\beta^2\sigma_s^2}{\Delta_i^2}(1 - \log 2 - \log\log T) + 1 + \frac{2}{\sqrt{2\pi e}}.$$

The second statement follows from the definition of the cumulative expected regret in Equation (2.3). □

Remark 4.3 (***Uninformative priors with short time horizon***)**.** When the deterministic UCL algorithm is used with an uncorrelated uninformative prior, Theorem 4.2 guarantees that the algorithm incurs logarithmic regret uniformly in horizon length T. However, for small horizon lengths, the upper bound on the regret can be lower bounded by a super-logarithmic curve. Accordingly, in practice, the cumulative expected regret curve may appear super-logarithmic for short time horizons. For example, for horizon T less than the number of arms N, the cumulative expected regret of the deterministic UCL algorithm grows at most linearly with the horizon length. □

Remark 4.4 (***Scaling of regret with*** N)**.** Often, (see, e.g., results in [20]), authors report the scaling of the regret with the number of arms N. Theorem 4.2 guarantees that the regret will scale at most linearly with N. However, as noted in Section 2.4 of [20], it is clear that the regret incurred from selecting an arm i can be at most $T\Delta_i$. This idea can be used (see [6]) to show that the overall cumulative expected regret of UCL will be upper bounded by a term of order $\sqrt{NT\log T}$. □

Remark 4.5 (***Comparison with UCB***)**.** In view of the bounds in Theorem 4.1, for an uninformative prior, the $(1 - 1/Kt)$-upper credible limit obeys

$$Q_i^t < \bar{m}_i^t + \beta\sigma_s\sqrt{\frac{1 + 2\log t - \log\log et^2}{n_i^t}}.$$

This upper bound is similar to the one in UCB1, which sets

$$Q_i^t = \bar{m}_i^t + \sqrt{\frac{2\log t}{n_i^t}}.$$

This is also similar to the one in Lai's upper confidence bound-based algorithm [61, Example 1], which sets

$$Q_i^t = \bar{m}_i^t + \sqrt{\frac{2g(n_i^t/T)}{n_i^t}},$$

where $g(t)$ is a function having the asymptotic expansion

$$g(t) = \log(t^{-1}) - \frac{1}{2}\log\log(t^{-1}) - \frac{1}{2}\log(16\pi) + o(1) \text{ as } t \to 0. \qquad \square$$

Remark 4.6 (***Informative priors***)**.** For an uninformative prior, i.e., very large variance σ_0^2, we established in Theorem 4.2 that the deterministic UCL algorithm achieves logarithmic regret uniformly in time. For informative priors, the cumulative expected regret depends on the quality of the prior. The quality of a prior on the rewards can be captured by the metric $\zeta := \max\{|m_i - \mu_i^0|/\sigma_0 \mid i \in \{1, \ldots, N\}\}$, where m_i is the true mean reward associated with arm i and μ_i^0 the prior mean belief about m_i. Recall that $\sigma_0 \geq 0$ can be interpreted as a measure of the agent's confidence about his/her prior belief. A *good prior* corresponds to small values of ζ, while a *bad prior* corresponds to large values of ζ. In other words, a good prior is one that has (i) mean close to the true mean reward, or (ii) a large variance. Intuitively, a good prior either has a fairly accurate estimate of the mean reward, or has low confidence about its estimate of the mean reward. For a good prior, the parameter K can be tuned such that

$$\Phi^{-1}\Big(1 - \frac{1}{Kt}\Big) - \max_{i \in \{1,\ldots,N\}} \frac{\sigma_s(|m_i - \mu_i^0|)}{\sigma_0^2} > \Phi^{-1}\Big(1 - \frac{1}{\bar{K}t}\Big),$$

where $\bar{K} \in \mathbb{R}_{>0}$ is some constant, and it can be shown, using the arguments of Theorem 4.2, that the deterministic UCL algorithm achieves logarithmic regret uniformly in time. A bad prior corresponds to a fairly inaccurate estimate of the mean reward and high confidence. For a bad prior, the cumulative expected regret may be a super-logarithmic function of the horizon length. $\square$

Remark 4.7 (***Sub-logarithmic regret for good priors***)**.** For a good prior with a small variance, even uniform sub-logarithmic regret can be achieved. Specifically, if the variable Q_i^t in Algorithm 1 is set to $Q_i^t = m_i^t + \sigma_i^t \Phi^{-1}(1 - 1/Kt^2)$, then an analysis similar to

Theorem 4.2 yields an upper bound on the cumulative expected regret that is dominated by (i) a sub-logarithmic term for good priors with small variance, and (ii) a logarithmic term for uninformative priors with a higher constant in front than the constant in Theorem 4.2. Notice that such good priors may correspond to human operators who have previous training in the task. □

4.3 The stochastic UCL algorithm with uncorrelated priors

To capture the inherent stochastic nature of human decision-making, we consider the UCL algorithm with stochastic arm selection. Stochasticity has been used as a generic optimization mechanism that does not require information about the objective function. For example, simulated annealing [13, 77, 53] is a global optimization method that attempts to break out of local optima by sampling locations near the currently selected optimum and accepting locations with worse objective values with a probability that decreases in time. By analogy with physical annealing processes, the probabilities are chosen from a Boltzmann distribution with a dynamic temperature parameter that decreases in time, gradually making the optimization more deterministic. An important problem in the design of simulated annealing algorithms is the choice of the temperature parameter, also known as a *cooling schedule*.

Choosing a good cooling schedule is equivalent to solving the explore-exploit problem in the context of simulated annealing, since the temperature parameter balances exploration and exploitation by tuning the amount of stochasticity (exploration) in the algorithm. In their classic work, Mitra *et al.* [77] found cooling schedules that maximize the rate of convergence of simulated annealing to the global optimum. In a similar way, the stochastic UCL algorithm (see Algorithm 2 in Appendix A for an explicit pseudocode implementation) extends the deterministic UCL algorithm (Algorithm 1) to the stochastic case. The stochastic UCL algorithm chooses an arm at time t using a Boltzmann distribution with temperature υ_t, so the probability P_{it} of picking arm i at time t is given by

$$P_{it} = \frac{\exp(Q_i^t/\upsilon_t)}{\sum_{j=1}^{N} \exp(Q_j^t/\upsilon_t)}.$$

In the case $\upsilon_t \to 0^+$ this scheme chooses $i_t = \arg\max\{Q_i^t \mid i \in \{1, \ldots, N\}\}$ and as υ_t increases the probability of selecting any other arm increases. Thus Boltzmann selection generalizes the maximum operation and is sometimes known as the soft maximum (or softmax) rule.

The temperature parameter might be chosen constant, i.e., $\upsilon_t = \upsilon$. In this case the performance of the stochastic UCL algorithm can be made arbitrarily close to that of the deterministic UCL algorithm by taking the limit $\upsilon \to 0^+$. However, [77] showed that good cooling schedules for simulated annealing take the form

$$\upsilon_t = \frac{\nu}{\log t},$$

so we investigate cooling schedules of this form. We choose ν using a feedback rule on the values of the heuristic function $Q_i^t, i \in \{1, \ldots, N\}$ and define the cooling schedule υ_t as

$$\upsilon_t = \frac{\Delta Q_{\min}^t}{2 \log t}, \tag{4.11}$$

where $\Delta Q_{\min}^t = \min\{|Q_i^t - Q_j^t| \mid i, j \in \{1, \ldots, N\}, i \neq j\}$ is the minimum gap between the heuristic function value for any two pairs of arms. We define $\infty - \infty = 0$, so that $\Delta Q_{\min}^t = 0$ if two arms have infinite heuristic values, and define $0/0 = 1$.

4.4 Regret analysis of the stochastic UCL algorithm

In this section we show that for an uninformative prior, the stochastic UCL algorithm achieves efficient performance. We define $\{R_t^{\mathrm{SUCL}}\}_{t \in \{1,\ldots,T\}}$ as the sequence of expected regret for the stochastic UCL algorithm. The stochastic UCL algorithm achieves logarithmic regret as formalized in the following theorem.

Theorem 4.8 (***Regret of the stochastic UCL algorithm***). *The following statements hold for the Gaussian multi-armed bandit problem and the stochastic UCL algorithm with uncorrelated uninformative prior and* $K = \sqrt{2\pi e}$*:*

1. *the expected number of times a suboptimal arm* i *is chosen until time* T *satisfies*

$$\mathbb{E}\left[n_i^T\right] \leq \left(\frac{8\beta^2\sigma_s^2}{\Delta_i^2} + \frac{2}{\sqrt{2\pi e}}\right) \log T + \frac{\pi^2}{6} + \frac{4\beta^2\sigma_s^2}{\Delta_i^2}(1 - \log 2 - \log\log T) + 1 + \frac{2}{\sqrt{2\pi e}};$$

2. the cumulative expected regret until time T satisfies

$$\sum_{t=1}^{T} R_t^{\text{SUCL}} \leq \sum_{i=1}^{N} \Delta_i \left(\left(\frac{8\beta^2\sigma_s^2}{\Delta_i^2} + \frac{2}{\sqrt{2\pi e}} \right) \log T + \frac{\pi^2}{6} + \frac{4\beta^2\sigma_s^2}{\Delta_i^2}(1 - \log 2 - \log\log T) + 1 + \frac{2}{\sqrt{2\pi e}} \right).$$

Proof. We start by establishing the first statement. We begin by bounding $\mathbb{E}[n_i^T]$ as follows

$$\mathbb{E}\left[n_i^T\right] = \sum_{t=1}^{T} \mathbb{E}\left[P_{it}\right] \leq \eta + \sum_{t=1}^{T} \mathbb{E}\left[P_{it}\mathbf{1}\left(n_i^t \geq \eta\right)\right], \tag{4.12}$$

where η is a positive integer.

Now, decompose $\mathbb{E}\left[P_{it}\right]$ as

$$\begin{aligned}\mathbb{E}[P_{it}] &= \mathbb{E}\left[P_{it}|Q_i^t \leq Q_{i^*}^t\right]\mathbb{P}\left(Q_i^t \leq Q_{i^*}^t\right) \\ &\quad + \mathbb{E}\left[P_{it}|Q_i^t > Q_{i^*}^t\right]\mathbb{P}\left(Q_i^t > Q_{i^*}^t\right) \\ &\leq \mathbb{E}\left[P_{it}|Q_i^t \leq Q_{i^*}^t\right] + \mathbb{P}\left(Q_i^t > Q_{i^*}^t\right). \end{aligned} \tag{4.13}$$

The probability P_{it} can itself be bounded as

$$P_{it} = \frac{\exp(Q_i^t/\upsilon_t)}{\sum_{j=1}^{N}\exp(Q_j^t/\upsilon_t)} \leq \frac{\exp(Q_i^t/\upsilon_t)}{\exp(Q_{i^*}^t/\upsilon_t)}. \tag{4.14}$$

Substituting the expression (4.11) for the cooling schedule υ_t in inequality (4.14), we obtain the bound

$$P_{it} \leq \exp\left(-\frac{2(Q_{i^*}^t - Q_i^t)}{\Delta Q_{\min}^t}\log t\right) = t^{-\frac{2(Q_{i^*}^t - Q_i^t)}{\Delta Q_{\min}^t}}. \tag{4.15}$$

For the purposes of the following analysis, define $\frac{0}{0} = 1$.

Since $\Delta Q_{\min}^t \geq 0$, with equality only if two arms have identical heuristic values, conditioned on $Q_{i^*}^t \geq Q_i^t$ the exponent on t in Equation (4.15) takes one of the following magnitudes:

$$\frac{|Q_{i^*}^t - Q_i^t|}{\Delta Q_{\min}^t} = \begin{cases} \frac{0}{0} = 1, & \text{if } Q_{i^*}^t = Q_i^t, \\ +\infty, & \text{if } Q_{i^*}^t \neq Q_i^t \text{ and } \Delta Q_{\min}^t = 0, \\ x, & \text{if } \Delta Q_{\min}^t \neq 0, \end{cases}$$

where $x \in [1, +\infty)$. The sign of the exponent is determined by the sign of $Q_{i^*}^t - Q_i^t$.

Consequently, it follows from inequality (4.15) that

$$\sum_{t=1}^{T} \mathbb{E}[P_{it} | Q_{i^*}^t \geq Q_i^t] \leq \sum_{t=1}^{T} \frac{1}{t^2} \leq \sum_{t=1}^{\infty} \frac{1}{t^2} = \frac{\pi^2}{6},$$

where the last equality is a well-known result often credited to Euler. It follows from inequality (4.13) that

$$\begin{aligned}
\sum_{i=1}^{T} \mathbb{E}[P_{it}] &\leq \frac{\pi^2}{6} + \sum_{i=1}^{T} \mathbb{P}\left(Q_i^t > Q_{i^*}^t\right) \\
&\leq \frac{\pi^2}{6} + \Big(\frac{8\beta^2\sigma_s^2}{\Delta_i^2} + \frac{2}{\sqrt{2\pi e}}\Big) \log T \\
&\quad + \frac{4\beta^2\sigma_s^2}{\Delta_i^2}(1 - \log 2 - \log\log T) + 1 + \frac{2}{\sqrt{2\pi e}},
\end{aligned}$$

where the last inequality follows from Theorem 4.2. This establishes the first statement. The second statement follows from the definition (2.3) of the cumulative expected regret. □

Comparing the first statement of Theorems 4.2 and 4.8, we see that the bounds on $\mathbb{E}\left[n_i^T\right]$ differ only by a constant $\pi^2/6 \approx 1.64$. This shows that the stochastic UCL algorithm pays a relatively small penalty for using a stochastic arm selection strategy instead of the deterministic one of UCL. This fact is of interest to neuroscience because it shows that stochastic decision rules, like those employed by humans, can achieve near-optimal performance. While the stochastic rule incurs a cost in terms of performance, it likely accrues an advantage in human decision making because a stochastic rule is easier to implement using neural wetware. From a neurological standpoint, the tuning rule we develop for υ_t using the minimum gap $\Delta Q_{\min}^t$ is implausible. A more plausible rule might involve the max-vs-next gap, i.e., $\Delta Q^t = Q_*^t - Q_{**}^t$, where $Q_*^t = \max_i Q_i^t$ and Q_{**}^t is the second-largest value at time t. Max-vs-next rules have been investigated in neuroscience (e.g., [76]), and developing such a tuning rule for υ_t would strengthen the connection of this work to the neuroscience literature.

4.5 The UCL algorithms with correlated priors

In the preceding sections, we considered the case of uncorrelated priors, i.e., the case with diagonal covariance matrix of the prior distribution for mean rewards $\Sigma_0 = \sigma_0^2 I_N$. However, in many cases there may be dependence among the arms that we wish to encode in the form of a non-diagonal covariance matrix. In fact, one of the main advantages a human may

have in performing a bandit task is prior experience with the dependency structure across the arms resulting in a good prior correlation structure. We show that including covariance information can improve performance and may, in some cases, lead to sub-logarithmic regret.

Let $\mathcal{N}(\boldsymbol{\mu}_0, \Sigma_0)$ and $\mathcal{N}(\boldsymbol{\mu}_0, \Sigma_{0d})$ be correlated and uncorrelated priors on the mean rewards from the arms, respectively, where $\boldsymbol{\mu}_0 \in \mathbb{R}^N$ is the vector of prior estimates of the mean rewards from each arm, $\Sigma_0 \in \mathbb{R}^{N\times N}$ is a positive definite matrix, and Σ_{0d} is the same matrix with all its non-diagonal elements set equal to 0. The inference procedure described in Section 4.1 generalizes to a correlated prior as follows: Define $\{\boldsymbol{\phi}_t \in \mathbb{R}^N\}_{t\in\{1,\dots,T\}}$ to be the indicator vector corresponding to the currently chosen arm i_t, where $(\boldsymbol{\phi}_t)_k = 1$ if $k = i_t$, and zero otherwise. Then the belief state $(\boldsymbol{\mu}_t, \Sigma_t)$ updates as in equations (3.3)–(3.5), which we repeat here for reference:

$$\begin{aligned} \mathbf{q} &= \frac{r_t \boldsymbol{\phi}_t}{\sigma_s^2} + \Lambda_{t-1}\boldsymbol{\mu}_{t-1} \\ \Lambda_t &= \frac{\boldsymbol{\phi}_t \boldsymbol{\phi}_t^T}{\sigma_s^2} + \Lambda_{t-1},\ \Sigma_t = \Lambda_t^{-1} \\ \boldsymbol{\mu}_t &= \Sigma_t \mathbf{q}, \end{aligned} \tag{4.16}$$

where $\Lambda_t = \Sigma_t^{-1}$ is the *precision* matrix.

The upper credible limit for each arm i can be computed based on the univariate Gaussian marginal distribution of the posterior with mean $\mu_i^t = (\boldsymbol{\mu}_t)_i$ and variance $(\sigma_i^t)^2 = (\Sigma_t)_{ii}$. Consider the evolution of the belief state with the diagonal (uncorrelated) prior Σ_{0d} and compare it with the belief state based on the non-diagonal Σ_0 which encodes information about the correlation structure of the rewards in the off-diagonal terms. The additional information means that the inference procedure will converge more quickly than in the uncorrelated case, as seen in Theorem 4.9. If the assumed correlation structure correctly models the environment, then the inference will converge towards the correct values, and the performance of the UCL and stochastic UCL algorithms will be at least as good as that guaranteed by the preceding analyses in Theorems 4.2 and 4.8.

Denoting $\sigma_i^{t\,2} = (\Sigma_t)_{ii}$ as the posterior at time t based on Σ_0 and $\sigma_{id}^{t\ 2} = (\Sigma_{td})_{ii}$ as the posterior based on Σ_{0d}, for a given sequence of chosen arms $\{i_\tau\}_{\tau\in\{1,\dots,T\}}$, we have that the variance of the non-diagonal estimator will be no larger than that of the diagonal one, as summarized in the following theorem:

Theorem 4.9 (***Correlated versus uncorrelated priors***)**.** *For the inference procedure in* (4.16)*, and any given sequence of selected arms* $\{i_\tau\}_{\tau\in\{1,\dots,T\}}$*,* $\sigma_i^{t\,2} \le \sigma_{id}^{t\ 2}$*, for any* $t \in \{0,\dots,T\}$*, and for each* $i \in \{1,\dots,N\}$*.*

Proof. We use induction. By construction, $\sigma_i^{0\,2} = \sigma_{id}^{0\,2}$, so the statement is true for $t = 0$. Suppose the statement holds for some $t \geq 0$ and consider the update rule for Σ_t. From the Sherman-Morrison formula for a rank-1 update [109], we have

$$(\Sigma_{t+1})_{jk} = (\Sigma_t)_{jk} - \left(\frac{\Sigma_t \boldsymbol{\phi}_t \boldsymbol{\phi}_t' \Sigma_t}{\sigma_s^2 + \boldsymbol{\phi}_t' \Sigma_t \boldsymbol{\phi}_t} \right)_{jk}.$$

We now examine the update term in detail, starting with its denominator:

$$\boldsymbol{\phi}_t' \Sigma_t \boldsymbol{\phi}_t = (\Sigma_t)_{i_t i_t},$$

so $\sigma_s^2 + \boldsymbol{\phi}_t' \Sigma_t \boldsymbol{\phi}_t = \sigma_s^2 + (\Sigma_t)_{i_t i_t} > 0$. The numerator is the outer product of the i_t^{th} column of Σ_t with itself, and can be expressed in index form as

$$(\Sigma_t \boldsymbol{\phi}_t \boldsymbol{\phi}_t' \Sigma_t)_{jk} = (\Sigma_t)_{j i_t} (\Sigma_t)_{i_t k}.$$

Note that if Σ_t is diagonal, then so is Σ_{t+1} since the only non-zero update element will be $(\Sigma_t)_{i_t i_t}^2$. Therefore, Σ_{td} is diagonal for all $t \geq 0$.

The update of the diagonal terms of Σ_t only uses the diagonal elements of the update term, so

$$\sigma_i^{(t+1)^2} = (\Sigma_{t+1})_{ii} = (\Sigma_t)_{ii} - \frac{(\Sigma_t)_{i i_t} (\Sigma_t)_{i_t i}}{\sigma_s^2 + \boldsymbol{\phi}_t' \Sigma_t \boldsymbol{\phi}_t}.$$

In the case of Σ_{td}, the update only changes the $i = i_t$ element whereas with the non-diagonal prior Σ_t the update may change all N terms. Define the function $f(x) = x - \frac{x^2}{\sigma_s^2 + x} = \frac{\sigma_s^2 x}{\sigma_s^2 + x}$. Note that $f(x)$ is a monotonically increasing function for $x > -\sigma_s^2$, and consider two cases: $i = i_t$ and $i \neq i_t$.

In the case $i = i_t$, performing the update to $\sigma_i^{t\,2}$ is equivalent to applying the function $f(x)$, so we have

$$\sigma_i^{(t+1)^2} = f(\sigma_i^{t\,2}), \quad \sigma_{id}^{(t+1)^2} = f(\sigma_{id}^{t\,2})$$

and the statement holds by the monotonicity of f since $\sigma_i^{t\,2} \leq \sigma_{id}^{t\,2}$ implies that

$$\sigma_i^{(t+1)^2} = f(\sigma_i^{t\,2}) \leq f(\sigma_{id}^{t\,2}) = \sigma_{id}^{(t+1)^2}.$$

In the case $i \neq i_t$, we have

$$\begin{aligned} \sigma_{id}^{(t+1)^2} &= \sigma_{id}^{t\,2}, \text{ and} \\ \sigma_i^{(t+1)^2} &= \sigma_i^{t\,2} - \frac{(\Sigma_t)_{i_t i_t}^2}{\sigma_s^2 + \boldsymbol{\phi}_t' \Sigma_t \boldsymbol{\phi}_t}, \end{aligned}$$

and the statement holds for $t + 1$. □

Note that the above result merely shows that the belief state converges more quickly in the case of a correlated prior, without making any claim about the correctness of this convergence. For example, consider a case where the prior belief is that two arms are perfectly correlated, i.e., the relevant block of the prior is a multiple of $\left(\begin{smallmatrix} 1 & 1 \\ 1 & 1 \end{smallmatrix}\right)$, but in actuality the two arms have very different mean rewards. If the algorithm first samples the arm with lower reward, it will tend to underestimate the reward to the second arm. However, in the case of a well-chosen prior the faster convergence will allow the algorithm to more quickly disregard related sets of arms with low rewards.

4.6 Discussion

In this chapter we have developed two variants of the UCL algorithm for the Gaussian multi-armed bandit problem. These algorithms use Bayesian inference to learn the value of the mean rewards m_i by combining prior beliefs with the information gathered from observing rewards. They choose the next arm to sample using deterministic or stochastic action selection strategies and a heuristic function of the belief state that can be interpreted in terms of the ambiguity bonus heuristic from the neuroscience literature. In the case that the agent's prior on m_i is uncorrelated and uninformative, we showed that both algorithms achieve logarithmic regret, i.e., optimal performance.

Using informative priors allows the algorithms to encode an agent's beliefs about the correlation structure among the arms, for example the structure inherited from the spatial embedding in a spatial multi-armed bandit problem. We showed that including correlation in the prior results in the belief state converging more quickly than in the case of an uncorrelated prior, but noted that depending on the quality of the correlation structure as a model of the world, this may result in either increased or decreased performance relative to the uncorrelated case. As a guide to the quality of a given prior, we developed the metric ζ which gives a quantitative measure of prior quality. This metric yields insight into the nature of good priors: in particular, they should either be accurate (close to the true value of m_i) or held with low confidence. Bad performance results when the prior encodes beliefs that are inaccurate and held with a high degree of confidence.

In the following chapter we study data from a human subject study where individuals performed a spatial multi-armed bandit task. We consider the data in the context of the stochastic UCL algorithm and show that the algorithm can be used as a model of human decision-making behavior in this task. Furthermore, we show that we can capture the types

of human performance exhibited in the data by varying the priors of the algorithm. In particular, we show that some humans exhibit high performance, which we interpret as reflecting high-quality priors.

The two algorithms and associated analysis presented in this chapter have been extended to two other generalizations of the Gaussian multi-armed bandit problem: the multi-armed bandit problem with transition costs, where there is a cost associated with switching from one arm to another, and the graphical multi-armed bandit problem, in which the arms are embedded in a graph and the set of arms available for selection at each decision time is the set of neighbors of the most recently selected arm. These extensions may prove valuable in applying the multi-armed bandit framework to robotics and other applications, and were studied in detail in [99].

Chapter 5

Data from a human-subject spatial search task[1]

"With four parameters I can fit an elephant, and with five I can make him wiggle his trunk."

(John von Neumann)

In this chapter, we consider data from a human-subject spatial multi-armed bandit task and show how human performance can be classified as falling into one of several categories, which we term *phenotypes*.[2] A significant fraction of the subjects exhibited performance better than that achievable with a frequentist multi-armed bandit algorithm. We then show through simulation that the stochastic UCL algorithm can produce performance that is analogous to the observed human performance. In light of these simulation results, we interpret the subjects' good performance as evidence that they have a high-quality prior.

5.1 Human behavioral experiment

In order to study human performance in multi-armed bandit tasks, we ran a spatial multi-armed bandit task through web servers at Princeton University. Human participants were recruited using Amazon's Mechanical Turk (AMT) web-based task platform [24]. Upon selecting the task on the AMT website, participants were directed to follow a link to a

[1]This chapter is adapted from Section V of [99]. Sections 5.1–5.3 in this chapter are mostly taken verbatim. This data has also appeared in part in the conference paper [97] and associated poster.

[2]The Merriam-Webster dictionary defines the word *phenotype* as follows: "The set of observable characteristics of an individual resulting from the interaction of its genotype with the environment." In this chapter we use the word to refer to the category of observed performance achieved by an individual subject.

Princeton University website[3], where informed consent was obtained according to protocol number 4779 approved by the Princeton University Institutional Review Board.

After informed consent was obtained, participants were shown instructions that told them they would be playing a simple game during which they could collect points, and that their goal was to collect the maximum number of total points in each part of the game.

Each participant was presented with a set of $N = 100$ options in a 10×10 grid. At each decision time $t \in \{1, \ldots, T\}$, the participant made a choice by moving the cursor to one element of the grid and clicking. After each choice was made a numerical reward associated to that choice was reported on the screen. The time allowed for each choice was manipulated and allowed to take one of two values, denoted fast and slow. If the participant did not make a choice within 1.5 (fast) or 6 (slow) seconds after the prompt, then the last choice was automatically selected again. The reward was visible until the next decision was made and the new reward reported. The time allotted for the next decision began immediately upon the reporting of the new reward. Figure 5.1 shows the screen used in the experiment.

The task was designed to be compatible with the social foraging task studied in [25], [126], [83], and [121], and to be a generalization of the problem considered in [100] and Chapter 3 above. These considerations affected the structure of the task, the parameter values (including time horizon T and reward surfaces), and pacing of the decision times.

The dynamics of the game were also experimentally manipulated, although we focus exclusively here on the first dynamic condition. The first dynamic condition was a standard bandit task, where the participant could choose any option at each decision time, and the game would immediately sample that option. In the second and third dynamic conditions, the participant was restricted in choices and the game responded in different ways. These two conditions are beyond the scope of this thesis.

Participants first familiarized themselves with the task by performing three training blocks of $T = 10$ choices each, one for each form of the game dynamics. Subsequently, the participants performed two task blocks of $T = 90$ choices each in a balanced experimental design. For each participant, the first task had parameters randomly chosen from one of the 12 possible combinations (2 timing, 3 dynamics, 2 landscapes), and the second task was conditioned on the first so that the alternative timing was used with the alternative landscape and the dynamics chosen randomly from the two remaining alternatives. In particular, only approximately 2/3 of the participants were assigned a standard bandit task, while other subjects were assigned other dynamic conditions. The horizon $T < N$ was chosen so that

[3]At the time of this writing, the task is still available as used for the experiment at the website `http://dcsl.princeton.edu/surveys/survey`, although the Institutional Review Board approval of the protocol has lapsed.

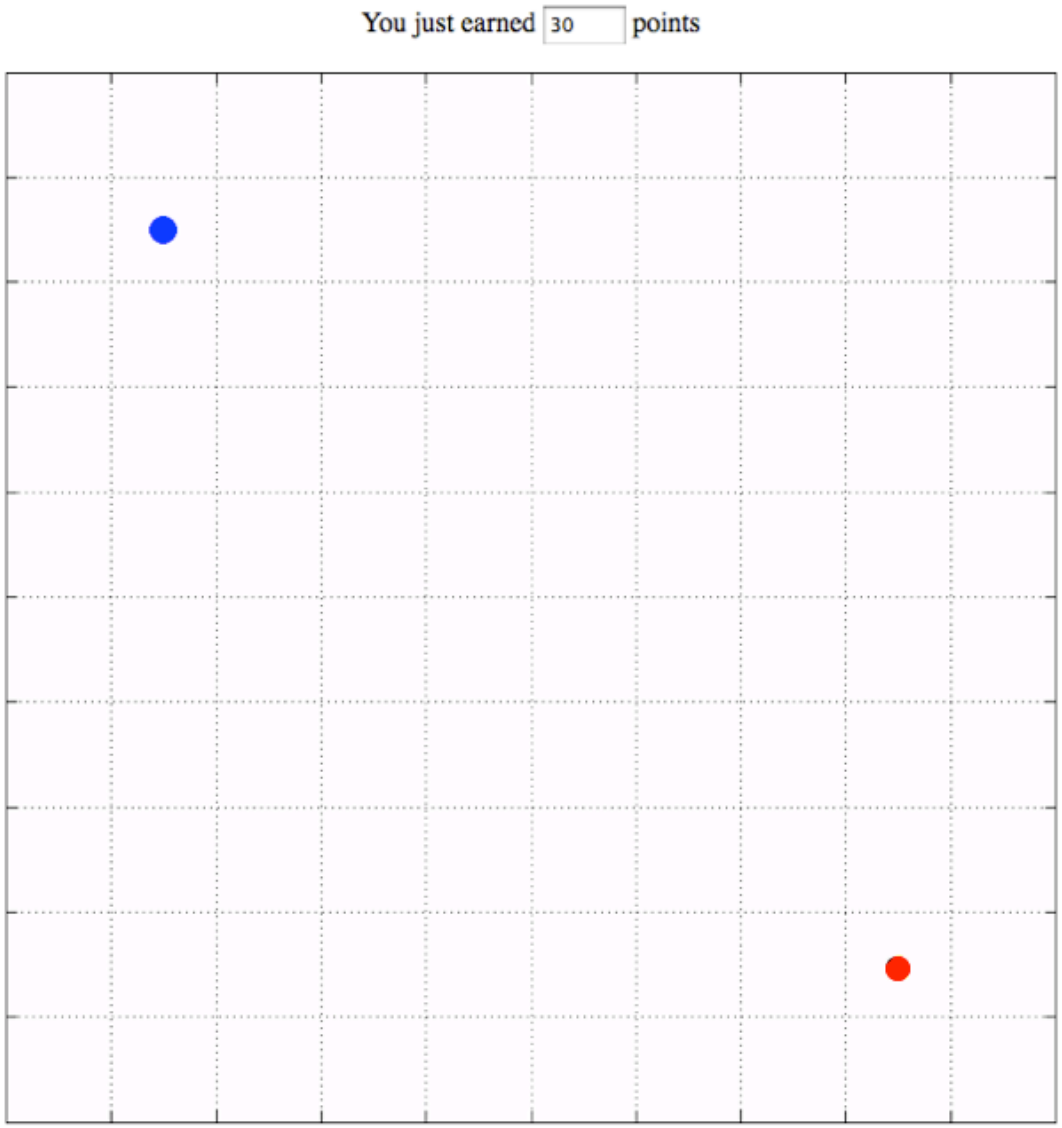


Figure 5.1: The screen used in the experimental interface. Each square in the grid corresponded to an available option. The text box above the grid displayed the most recently received reward, the blue dot indicated the participant's most recently recorded choice, and the smaller red dot indicated the participant's next choice. In the experiment, the red dot was colored yellow, but here we have changed the color for legibility. When both dots were located in the same square, the red dot was superimposed over the blue dot such that both were visible. Initially, the text box was blank and the two dots were together in a randomly chosen square. Participants indicated a choice by clicking in a square, at which point the red dot would move to the chosen option. During the time allotted for a given decision, participants could change their decision without penalty by clicking on another square, and the red dot would move accordingly. When the decision time had elapsed, the blue dot would move to the new square, the text box above the grid would be updated with the most recent reward amount, and the choice would be recorded. Previously published as Figure 5 of [99].

prior beliefs would be important to performing the task. Each training block took 15 seconds and each task block took 135 (fast) or 540 (slow) seconds. The time between blocks was negligible, due only to network latency.

Mean rewards in the task blocks corresponded to one of two landscapes: Landscape A (Figure 5.2(a)) and Landscape B (Figure 5.2(b)). Both landscapes are drawn from the task presented in [126] and studied in [83]. In those works, Landscape A is referred to as "Converging Gaussians" (CG) and Landscape B is referred to as "Rising Optimum" (RO). Landscape B is identical to the reward surface studied in [100] and Chapter 3. Each landscape was flat along one dimension and followed a profile along the other dimension. In the two task blocks, each participant saw each landscape once, presented in random order. Both landscapes had a mean value of 30 points and a maximum of approximately 60 points, and the rewards r_t for choosing an option i_t were computed as the sum of the mean reward m_{i_t} and an integer chosen uniformly from the range $[-5, 5]$. In the training blocks, the landscape had a mean value of zero everywhere except for a single square with a value of 100 points in the center. The participants were given no specific information about the value or the structure of the reward landscapes.

To incentivize the participants to make choices to maximize their cumulative reward, the participants were told that they were being paid based on the total reward they collected during the tasks. As noted above, due to the multiple manipulations, not every participant performed a standard bandit task block. Data were collected from a total of 417 participants: 326 of these participants performed one standard bandit task block each, and the remaining 91 participants performed no standard bandit task blocks.

5.2 Phenotypes of observed performance

For each 90 choice standard bandit task block, we computed observed regret by subtracting the maximum mean cumulative reward from the participant's cumulative reward, i.e.,

$$\mathcal{R}(t) = m_{i^*} t - \sum_{\tau=1}^{t} r_\tau.$$

The definition of $\mathcal{R}(t)$ uses received reward rather than expected reward, so it is not identical to cumulative expected regret. However, due to the large number of individual rewards received and the small variance in rewards, the difference between the two quantities is small.

We study human performance by considering the functional form of $\mathcal{R}(t)$. Optimal performance in terms of regret corresponds to $\mathcal{R}(t) = \mathcal{C} \log t$, where $\mathcal{C}$ is the sum over i of the factors in (2.6). The worst-case performance, corresponding to repeatedly choosing the lowest-value option, corresponds to the form $\mathcal{R}(t) = \mathcal{K}t$, where $\mathcal{K} > 0$ is a constant. Other bounds in the bandit literature, notably in the continuum-armed bandit problem, (e.g. [116]) are known to have the form $\mathcal{R}(t) = \mathcal{K}\sqrt{t}$.

To classify types of observed human performance in bandit tasks, we fit models representing these three forms to each individual participant's observed regret from each task. Specifically, we fit the three models

$$\mathcal{R}(t) = a + bt \tag{5.1}$$

$$\mathcal{R}(t) = at^b \tag{5.2}$$

$$\mathcal{R}(t) = a + b\log(t) \tag{5.3}$$

to the data from each task and classified the behavior according to which of the models (5.1)–(5.3) best fit the data in terms of squared residuals. Model selection using this procedure is tenable given that the complexity or number of degrees of freedom of the three models is the same.

Of the 326 participants who performed a standard bandit task block, 59.2% were classified as exhibiting linear regret (model (5.1)), 19.3% power regret (5.2), and 21.5% logarithmic regret (5.3). This suggests that 40.8% of the participants performed better than a standard algorithm based on frequentist statistics would have and 21.5% achieved effectively optimal performance. We observed no significant correlation between performance and timing, landscape, or order (first or second) of playing the standard bandit task block.

Averaging across all tasks, mean performance was best fit by a power model with exponent $b \approx 0.9$, so participants on average achieved sub-linear regret, i.e., better than linear regret. The nontrivial number of positive performances are noteworthy given that $T < N$, i.e., a relatively short time horizon which makes the task challenging. By comparison, an algorithm based on a frequentist estimator, such as UCB-Normal, would have to initialize its estimates by sampling each arm once. Consequently, such an algorithm would, on average, achieve linear regret on a task with a short horizon $T < N$.

Averaging over subjects in each phenotype, conditional on the best-fit model for each subject, separates the performance of the participants into the three categories of regret performance as can be observed in Figure 5.3. The difference between linear and power-law performance is not statistically significant until near the task horizon at $t = 90$, but log-law performance is statistically different from the other two from $t \approx 30$, as seen using the con-

fidence intervals in the figure. We therefore interpret the linear and power-law performance phenotypes as representing participants with low performance and the log-law phenotype as representing participants with high performance. Interestingly, the three models are indistinguishable for time less than sufficiently small $t \lesssim 30$. This may represent a fundamental limit to performance that depends on the complexity of the reward surface: if the surface is smooth, skilled participants can quickly find good options, corresponding to a small value of the constant $\mathcal{K}$, and thus their performance will quickly be distinguished from that of less skilled participants. However, if the surface is rough, identifying good options is harder and will therefore require more samples, i.e., a large value of $\mathcal{K}$, even for skilled participants.

5.3 Comparison with UCL

Having identified the three phenotypes of observed human performance in the above section, we show that the stochastic UCL algorithm (Algorithm 2) can produce behavior corresponding to the linear-law and log-law phenotypes by varying a minimal number of parameters. Parameters are used to encode the prior beliefs and the decision noise of the participant. A minimal set of parameters is given by the four scalars μ_0, σ_0, λ and υ, defined as follows.

(i) **Prior mean** The model assumes prior beliefs about the mean rewards to be a Gaussian distribution with mean $\boldsymbol{\mu}_0$ and covariance Σ_0. It is reasonable to assume that participants set $\boldsymbol{\mu}_0$ to the uniform prior $\boldsymbol{\mu}_0 = \mu_0 \mathbf{1}_N$, where $\mathbf{1}_N \in \mathbb{R}^N$ is the vector with every entry equal to 1. Thus, $\mu_0 \in \mathbb{R}$ is a single parameter that encodes the participants' beliefs about the mean value of rewards.

(ii,iii) **Prior covariance** For a spatially-embedded task, it is reasonable to assume that arms that are spatially close will have similar mean rewards. Following [68] we choose the elements of Σ_0 to have the form

$$\Sigma_{ij} = \sigma_0^2 \exp(-|\mathbf{x}_i - \mathbf{x}_j|/\lambda), \tag{5.4}$$

where $\mathbf{x}_i$ is the location of arm i and $\lambda \geq 0$ is the correlation length scale parameter that encodes the spatial smoothness of the reward surface. The case $\lambda = 0$ represents complete independence of rewards, i.e., a very rough surface, while as λ increases the agent believes the surface to be more smooth. The parameter $\sigma_0 \geq 0$ can be interpreted as a confidence parameter, with $\sigma_0 = 0$ representing absolute confidence in the beliefs about the mean $\boldsymbol{\mu}_0$, and $\sigma_0 = +\infty$ representing complete lack of confidence.

(iv) **Decision noise** In Theorem 4.8 we show that for an appropriately chosen cooling schedule, the stochastic UCL algorithm with softmax action selection achieves logarithmic regret.

However, the assumption that human participants employ this particular cooling schedule is unreasonably strong. It is of great interest in future experimental work to investigate what kind of cooling schedule best models human behavior. The Bayes-optimal cooling schedule can be computed using variational Bayes methods [38]; however, for simplicity, we model the participants' decision noise by using softmax action selection with a constant temperature $\upsilon \geq 0$. This yields a single parameter representing the stochasticity of the decision-making: in the limit $\upsilon \to 0^+$, the model reduces to the deterministic UCL algorithm, while with increasing υ the decision-making is increasingly stochastic.

With this set of parameters, the prior quality ζ from Remark 4.6 reduces to $\zeta = (\max_i |m_i - \mu_0|)/\sigma_0$. Uninformative priors correspond to very large values of σ_0. Good priors, corresponding to small values of ζ, have μ_0 close to $m_{i^*} = \max_i m_i$ or little confidence in the value of μ_0, represented by large values of σ_0.

We compare the model to observed behavioral data by fixing parameter values and having the model make choices in simulated games. We then categorize the behavior observed in simulation using the same fitting procedure used for human subjects. By adjusting the parameters, we can replicate both linear and logarithmic observed regret behaviors as seen in the human data.

Figure 5.4 shows examples of simulated observed regret $\mathcal{R}(t)$ that capture linear and logarithmic regret, respectively. In both examples, Landscape B was used for the mean rewards. The example with linear regret shows a case where the agent has fairly uninformative and fully uncorrelated prior beliefs (i.e., $\lambda = 0$). The prior mean $\mu_0 = 30$ is set equal to the true surface mean, but with $\sigma_0^2 = 1000$, so that the agent is not very certain of this value. Moderate decision noise is incorporated by setting $\upsilon = 4$. The values of the prior encourage the agent to explore most of the $N = 100$ options in the $T = 90$ choices, yielding regret that is linear in time. As emphasized in Remark 4.3, the deterministic UCL algorithm (and any agent employing the algorithm) with an uninformative prior cannot in general achieve sub-linear cumulative expected regret in a task with such a short horizon. The addition of decision noise to this algorithm will tend to increase regret, making it harder for the agent to achieve sub-linear regret.

In contrast, the example with logarithmic regret shows how an informative prior with an appropriate correlation structure can significantly improve the agent's performance. The prior mean $\mu_0 = 200$ encourages more exploration than the previous value of 30, but the smaller value of $\sigma_0^2 = 10$ means the agent is more confident in its belief and will explore less. The correlation structure induced by setting the length scale $\lambda = 4$ is a good model for the reward surface, allowing the agent to more quickly reject areas of low rewards. Furthermore, a lower softmax temperature $\upsilon = 1$ means that the agent's decisions are made

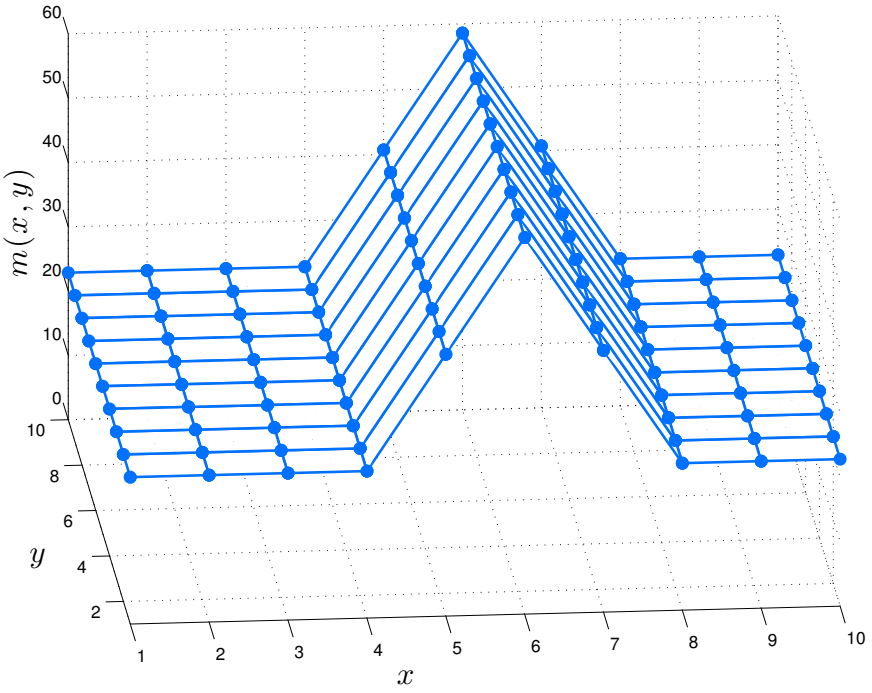
more deterministically. Together, these differences lead to the agent's logarithmic regret curve; this agent suffers less than a third of the total regret during the task as compared to the agent with the poorer prior and linear regret.

The simulations in Figure 5.4 suggest that there are various regions of parameter space that result in qualitatively different regret behaviors, and that there may be phase transitions between these different regions. Characterization of these regions and phase transitions is an open question, but it is clear that the agent's prior plays an important role. We have begun to characterize the effect of the prior on regret and will discuss the findings in future work [119].

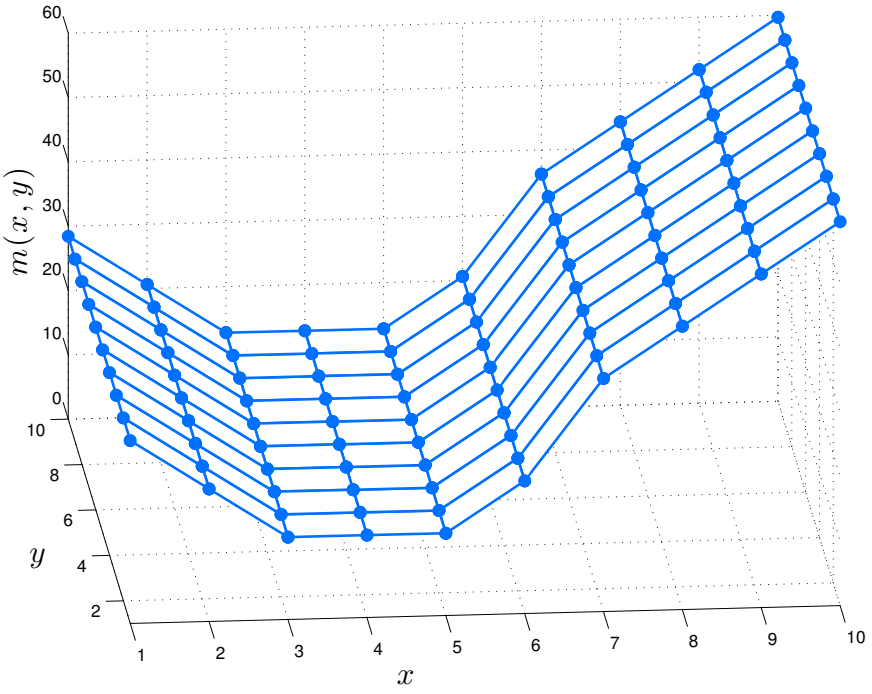
5.4 Discussion

In this chapter, we have studied human subject data from a spatial multi-armed bandit task. We showed that human performance in this task falls into one of several phenotypes that can be interpreted in terms of performance bounds from the multi-armed bandit literature. A significant fraction of the subjects exhibited performance that is better than the average performance achievable using a multi-armed bandit algorithm based on frequentist statistics, which we interpret as showing that some humans have high-quality priors for spatial search tasks.

Through simulation, we showed that the stochastic UCL algorithm could be used as a model of human choice behavior and capture the various phenotypes of human performance using a minimal set of four parameters that encode the human's prior and decision noise level. If one can fit these four parameters to empirical human choice data, one can extract the prior used to make the decisions. By extracting the prior from a human with a high-quality prior, we can improve the performance of a human-machine system over that of a machine with an uninformative prior.



(a)



(b)

Figure 5.2: The two task reward landscapes: (a) Landscape A, (b) Landscape B. The two-dimensional reward surfaces followed the profile along one dimension (here the x direction) and were flat along the other (here the y direction). The Landscape A profile is designed to be simple in the sense that the surface is concave and there is only one global maximum ($x = 6$), while the Landscape B profile is more complicated since it features two local maxima ($x = 1$ and 10), only one of which ($x = 10$) is the global maximum. Previously published as Figure 6 of [99].

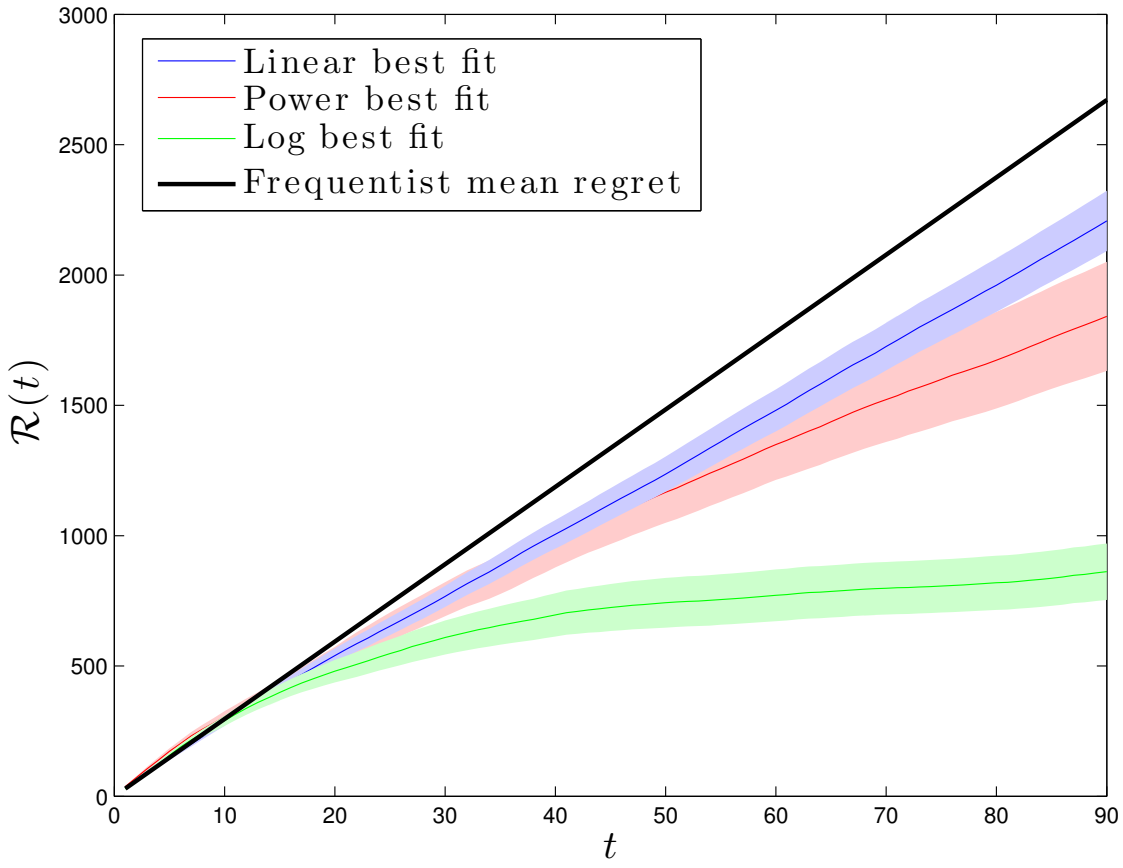


Figure 5.3: Mean observed regret $\mathcal{R}(t)$ conditional on the best-fit model (5.1)–(5.3), along with bands representing 95% confidence intervals based on the standard error of the mean. The width of the confidence intervals indicates the spread in the regret curves among different subjects in each phenotype. The black curve shows the linear expected regret that would be achieved by a frequentist algorithm, which would make its first $N = 100$ choices at random. Subjects who incur linear regret do so at a slower average rate than a frequentist algorithm would, indicating that their choices are not made purely at random. Note how the difference between linear and power-law regret is not statistically significant until near the task horizon $T = 90$, while logarithmic regret is significantly less than that of the linear and power-law cases. Adapted from Figure 7 of [99].

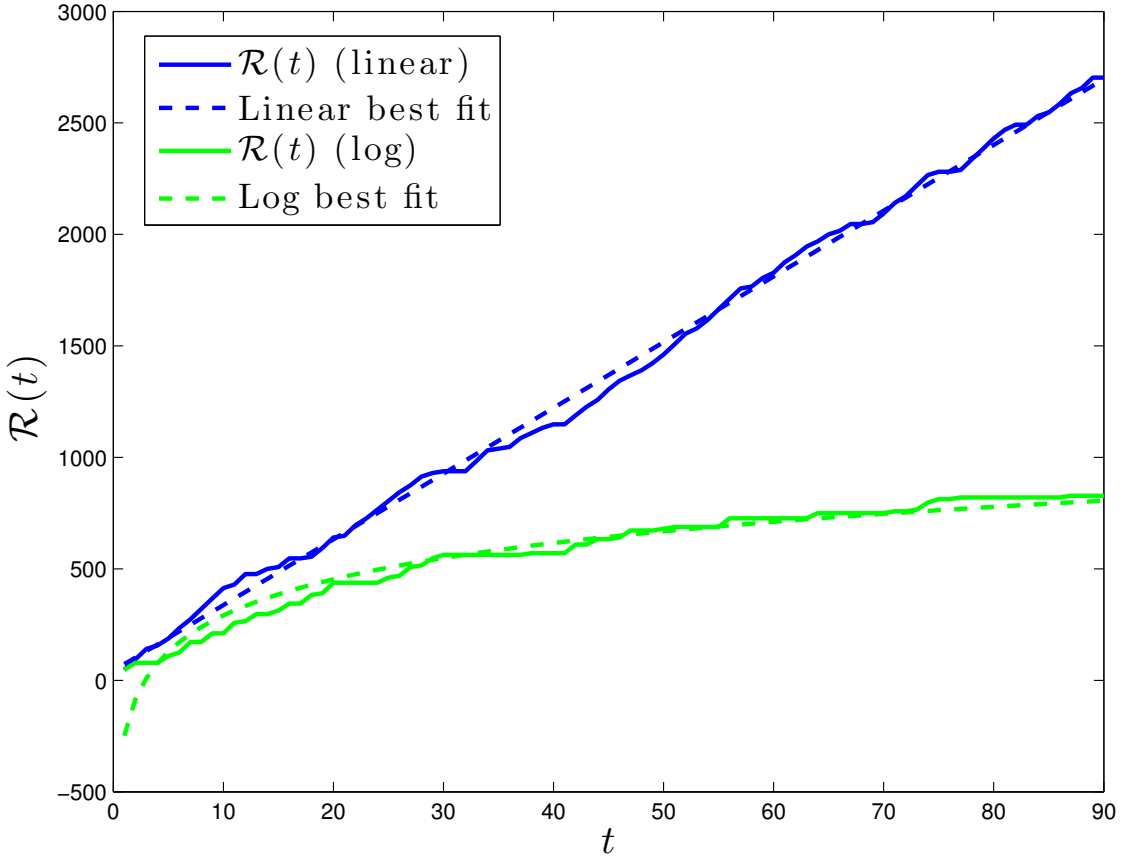


Figure 5.4: Observed regret $\mathcal{R}(t)$ from simulations (solid lines) that demonstrate linear regret (5.1), blue curves, and log regret (5.3), green curves. The best fits to the simulations are shown (dashed lines). The simulated task parameters were identical to those of the human participant task with Landscape B from Figure 5.2(b). In the example with linear regret, the agent's prior on rewards was the uncorrelated prior $\mu_0 = 30$, $\sigma_0^2 = 1000, \lambda = 0$. Decision noise was incorporated using softmax selection with a constant temperature $\upsilon = 4$. In the example with log regret, the agent's prior on rewards was the correlated prior with uniform $\mu_0 = 200$ and Σ_0 an exponential prior (5.4) with parameters $\sigma_0^2 = 10, \lambda = 4$. The decision noise parameter was set to $\upsilon = 1$. Previously published as Figure 8 of [99].

Chapter 6

Parameter estimation for softmax decision-making models[1]

"If your experiment needs statistics, you ought to have done a better experiment."
(Ernest Rutherford)

In this chapter, we consider the stochastic UCL algorithm as a model of human behavior and study the problem of estimating the model parameters from observed choice data. By estimating the model parameters, one can extract the human subject's prior over the rewards. As shown in the previous chapter, many humans have high-quality priors that can result in better performance than is possible with an uninformative prior, so estimating the prior will be valuable to constructing integrated human-machine systems.

We study the parameter estimation problem using a likelihood-based approach. The stochastic UCL model trivially defines a likelihood function, which quantifies how likely the observed data would be under a given model as a function of the parameters. Maximizing the likelihood function (perhaps with a penalty for parameter values thought to be unlikely) produces the estimate of parameter values that best explains the data for the given model. Unfortunately the stochastic UCL model likelihood function is poorly behaved in general, meaning that finding the optimum parameter values is difficult and the convergence properties of the resulting estimator are difficult to quantify.

In this chapter, we develop an estimator for the stochastic UCL model parameters using approximations to the likelihood function. The development proceeds as follows: we first consider decision-making models that use softmax action selection with an objective function that is a linear function of the unknown model parameters. For such models, we prove conditions under which the likelihood function is concave and show that in this case the maximum likelihood estimator converges to the true parameter value. The concavity of the

[1]This chapter is adapted from [94], with most text taken verbatim.

likelihood function permits the estimator to be computed using standard convex optimization tools. We also develop an iterative parameter estimation algorithm that could allow better performance in some cases. We then show that this algorithm can be applied to the stochastic UCL model by linearizing the heuristic function about a nominal point in parameter space. The resulting estimator is relevant for models involving softmax action selection, which includes a wide variety of models in neuroscience and machine learning.

The remainder of the chapter is organized as follows. In Section 6.1 we introduce softmax decision-making models and briefly review the relevant literature. In Section 6.2 we review the literature on Generalized Linear Models (GLMs) and show that softmax decision-making models with linear objective functions are a special case of this general class of statistical models. In Section 6.3 we adopt a maximum likelihood estimation framework and define the estimation problem for softmax decision-making models in terms of optimizing the model likelihood function. We review two relevant results from the literature: 1) an approach to solving the resulting optimization problem, and 2) standard results concerning the convergence of the maximum likelihood estimator. In Section 6.4 we provide several examples of softmax decision models that appear in the literature. In Section 6.5 we analytically compute the gradient and Hessian matrix of the likelihood function and use these to develop an iterative algorithm to solve the likelihood maximization problem. We study the Hessian matrix in detail to derive conditions under which the iterative algorithm converges to the correct parameter values. These conditions also imply that the likelihood function is concave. In Section 6.6 we demonstrate the convergence results by presenting results from several examples where we applied the estimator to simulated data. In Section 6.7 we show that this estimator can be applied to the stochastic UCL model by linearizing the likelihood function about a nominal point in parameter space. In Section 6.8 we discuss the implications for human-machine systems and conclude.

6.1 Introduction

In a variety of decision-making scenarios an agent selects one among a discrete set of options $i \in \{1, \ldots, m\}$. In the literature, decision-making models are derived and applied to study and predict the strategies that agents use to make their selections and to evaluate decision-making performance. One common approach is to derive a decision-making model as the solution of an optimization problem. An objective function Q_i is defined for each option i, and the model agent selects the option i^* that maximizes the objective function:

$$i^* = \arg\max_i Q_i.$$

The maximum operation is deterministic and non-differentiable, so for many applications it is replaced by the so-called softmax operation, under which option i is chosen with probability

$$\mathbb{P}(i) = \frac{\exp(Q_i)}{\sum_{j=1}^{m} \exp(Q_j)}.$$

The softmax operation is a stochastic, biologically-plausible approximation of the maximum operation [124]. Furthermore, it is differentiable with respect to its arguments Q, which makes it more analytically tractable.

In contexts such as inverse reinforcement learning [103, 87] and neuroscience [82], a central goal is to understand the decision-making process by finding the objective function values $\{Q_i\}$ that explain observed decisions. In this chapter, we consider this problem in the case that the objective function Q is linear in a set of n_{obj} known objective variables $\mathbf{x}$, i.e.,

$$Q_i = \boldsymbol{\theta}^T \mathbf{x}_i, \ \boldsymbol{\theta}, \mathbf{x}_i \in \mathbb{R}^{n_{obj}}. \tag{6.1}$$

Our goal is to learn the vector of parameters $\boldsymbol{\theta}$, which is assumed to be constant across options and decisions.

The problem of learning the objective function that can explain observed decision-making behavior is relevant for several different disciplines. In econometrics and signal processing, it is a parameter estimation problem. In system identification, it can be considered as a grey-box modeling problem. As a motivating example, consider the case of $m = 2$ options and $n_{obj} = 1$ known variables, such that both $\boldsymbol{\theta} = \theta$ and $\mathbf{x}_i = x_i$ are scalar. Then the probability of picking option 1 is

$$\mathbb{P}(\text{pick option 1}) = \frac{1}{1 + \exp(-\theta(x_1 - x_2))}. \tag{6.2}$$

Figure 6.1 plots the probability (6.2) as a function of the difference in value of the two options $\Delta x = x_1 - x_2$. When the values of the two options are identical, the probability is equal to 0.5 and it increases monotonically with increasing Δx. The rate of the increase is controlled by θ, which sets the slope of the function at $\Delta x = 0$. Large values of θ increase the slope and make the choice represented by (6.2) discriminate between x_1 and x_2 with more sensitivity, while small values of θ decrease the slope and make the choice less sensitive to Δx. Models of this form have been used to study a variety of decision-making tasks [64, 105, 34, 84, 121], where finding the value of θ that explains a given set of decisions is an important problem.

Our problem has similarities to ones previously studied in the literature, in particular multinomial logistic regression [14, 59]. In multinomial logistic regression, one is given a

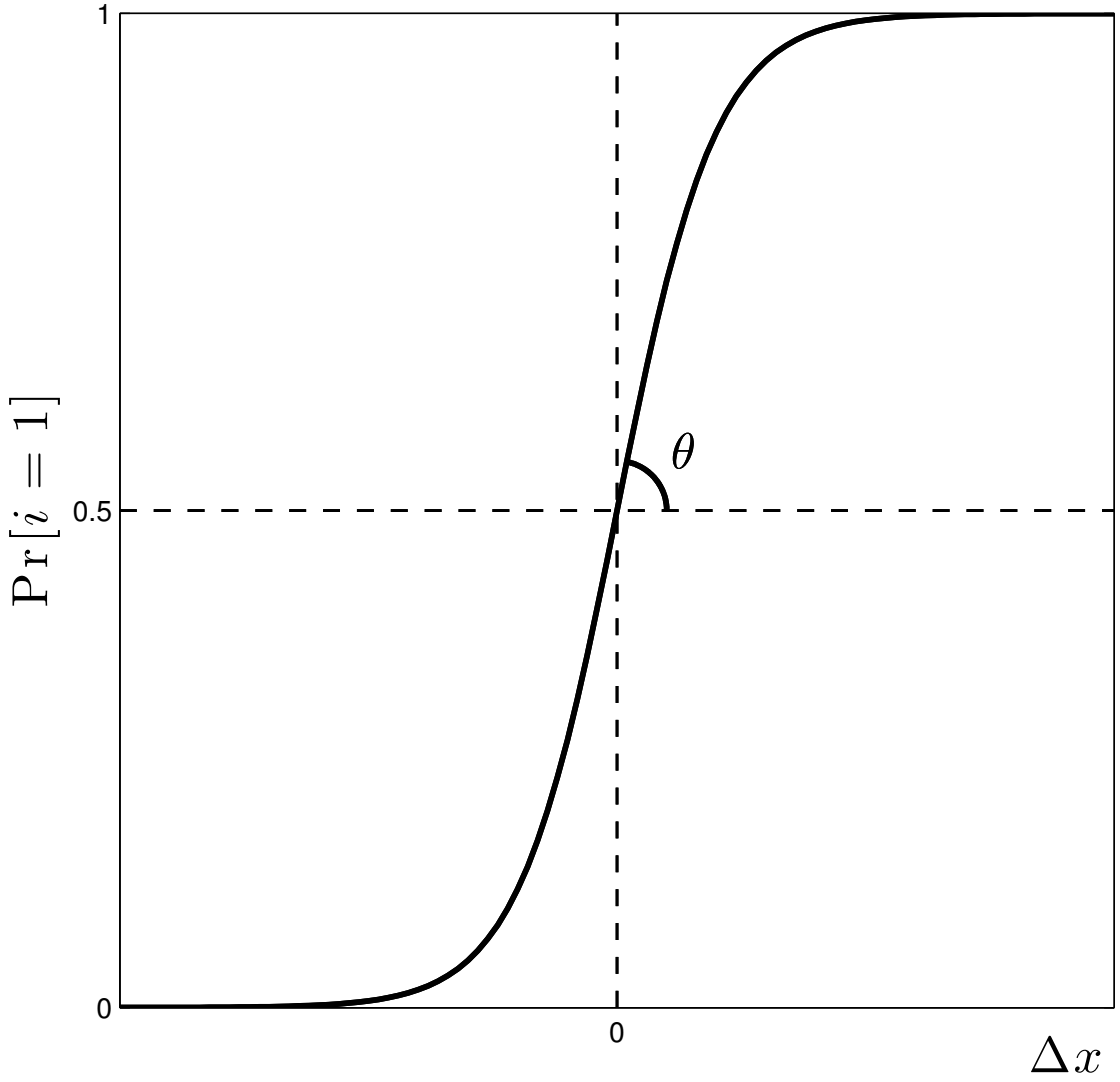


Figure 6.1: The probability (6.2) from the model (6.1) with $m = 2$ options and a scalar ($n_{obj} = 1$) parameter θ. The probability of picking option 1 is a logistic function of $\Delta x = x_1 - x_2$ and the sensitivity to Δx is controlled by θ, which sets the slope at $\Delta x = 0$.

vector of explanatory variables $\mathbf{x}$ that may belong to one of m classes $i \in \{1, \ldots, m\}$, and the goal is to predict the class to which the observed variables belong. The multinomial logistic regression model defines a weight vector $\mathbf{w}_i$ for each class and uses the softmax operation with the objective function $Q_i(\mathbf{x}) = \mathbf{w}_i^T \mathbf{x}$. The parameter estimation problem for the multinomial logistic regression model involves learning the values of all the weight vectors $\mathbf{w}_i, i \in \{1, \ldots, m\}$, i.e., $m \times n_{obj}$ individual numbers. Both our model and the multinomial logistic regression model are instances of Generalized Linear Models, or GLMs [85]. However, our problem is dual to the standard multinomial logistic regression problem in that the dependence of the weight vector and the explanatory variables on the class has been reversed.

The assumption of a linear form for the objective function (6.1) may appear restrictive, but many relevant models can be reduced to this form, at least locally or in certain limits. In Chapter 4, we developed the stochastic UCL algorithm, and in Chapter 5 we showed that it can be used as model for human behavior in spatial search tasks. Stochastic UCL is a softmax decision model with an objective function Q_{UCL} that depends on several parameters which encode the human subject's prior and level of decision noise. We wish to estimate the values of these parameters both to enable more rigorous analysis of the human subject data presented in Chapter 5 and to facilitate the integration of the stochastic UCL model in engineered systems. In Section 6.7 below we show that Q_{UCL} can be transformed into a linear function of the form (6.1) by linearizing about a point in parameter space.

Previous work, e.g. Krishnapuram *et al.* [59], has developed fast algorithms for learning the parameters of multinomial logistic regression models, but the dual structure of the model (6.1) precludes the use of Krishnapuram *et al.*'s algorithm here. Krishnapuram *et al.*'s bound optimization framework is applicable to the model (6.1), but additional work is required to derive the analogous algorithm and to analyze its convergence behavior. We develop an algorithm for the parameter estimation problem for the general case of model (6.1) and prove conditions under which the algorithm converges to the true model parameters. We then apply this algorithm to the problem of estimating parameters of the UCL model.

There are three major contributions of the work reported in this chapter. The first two concern the case of a linear objective function of the form (6.1). First, we derive conditions under which the likelihood function of a model with such an objective function is concave (Lemma 6.10). In the case these conditions are satisfied, we develop and prove the convergence of a fast iterative algorithm for performing maximum likelihood parameter estimation of softmax decision-making models with linear objective functions. Second, in proving the concavity of the likelihood function, we construct several new matrix operations and derive some of their important properties. The first is a binary matrix product that

generalizes the Hadamard product to the case where one matrix has a block structure, while the second is a block-wise matrix contraction operator. For the block Hadamard product, which is a special case of the Khatri-Rao product [52, 72], we prove a theorem analogous to the Schur product theorem for the standard Hadamard product. The third major contribution concerns the extension to softmax decision-making models with nonlinear objective functions. In Section 6.7 we show in the case of UCL that the approach developed for linear objective functions can be applied by linearizing about a nominal point in parameter space, which produces an estimator for the UCL model. Linearization makes this approach applicable to more general models with nonlinear objective functions.

6.2 Generalized Linear Models

Linear models of the form

$$\mathbb{E}[y] = \mu = \mathbf{w}^T\mathbf{x}; \quad y \sim \mathcal{N}(\mu, \sigma^2), \tag{6.3}$$

are the basis of most analyses of continuous data. In such a model, the observed data consists of the explanatory variables $\mathbf{x}$ and the response variable y. For example, the explanatory variables might be a child's age and height and the response variable their weight. In this case, the model predicts the child's weight given their age and height. The model assumes that the expected value of the response y is a linear combination of the explanatory variables $\mathbf{x}$ with the unknown parameters $\mathbf{w}$. Such a model is appropriate when the response variable follows a normal distribution $\mathcal{N}(\mu, \sigma^2)$.

However, this sort of model has two important limitations: 1) the response variables y often follow a distribution other than the normal distribution, for example they may be discrete rather than continuous, and 2) the relationship between the response variables y and the explanatory variables $\mathbf{x}$ need not be linear.

To help deal with these issues, a variety of researchers considered generalizations of the linear model (6.3), which Nelder and Wedderburn [85] unified in the *generalized linear model*, or GLM. In the GLM, there is a nonlinear function relating the expected response $\mathbb{E}[y] = \mu$ to the linear predictor $\mathbf{w}^T\mathbf{x}$:

$$f(\mu) = \mathbf{w}^T\mathbf{x}.$$

The function $f(\cdot)$ is termed the *link* function. For example, consider the example with $m = 2$ options from Figure 6.1. In this case, the explanatory variable is the difference in values Δx and the response variable y takes value 1 if option 1 is chosen and zero otherwise. Therefore,

the response variable y is binomial with success probability given by (6.2), i.e.,

$$y = \begin{cases} 1, & \text{with probability } \mu, \\ 0, & \text{else.} \end{cases}$$

Its expected value is

$$\mathbb{E}[y] = \mu = \mathbb{P}(\text{choose option 1}) = \frac{1}{1 + \exp(-\mathbf{w}^T \mathbf{x})},$$

where $\mathbf{w} = \theta$ and $\mathbf{x} = \Delta x$.

Inverting this equation, we get the link function

$$f(\mu) = \mathbf{w}^T \mathbf{x} = \log\left(\frac{\mu}{1-\mu}\right),$$

which is known as the *logit* function. In the case of $m > 2$ options, the response data is referred to as multinomial and the response variable can be encoded as a vector $\mathbf{y}$ whose i^{th} component y_i takes value 1 if option i is chosen and zero otherwise:

$$(\mathbf{y})_i = y_i = \begin{cases} 1, & \text{option } i \text{ is chosen,} \\ 0, & \text{else.} \end{cases}$$

Letting $\mu_i = \mathbb{E}[y_i]$, the following link function holds:

$$f(\mu_i) = \mathbf{w}_i^T \mathbf{x} = \log\left(\frac{\mu_i}{1-\mu_i}\right),$$

where $\mathbf{w}_i$ are the parameters associated with option i. This model is known as *multinomial logistic regression*. In Section 6.2.2 we will show that softmax decision-making models with linear objective functions of the form (6.1) are a special case of the multinomial logistic regression model where the parameters $\mathbf{w}_i$ have a certain sparse structure. This allows us to develop a parameter estimation algorithm by extending the framework of Krishnapuram *et al.* [59].

6.2.1 Multinomial logistic regression

In the spirit of [59], we set the following notation. We assume we have n observations, and for each observation we have data consisting of explanatory variables and a response, which falls into one of m categories. Specifically, for each observation $k \in \{1, \ldots, n\}$ we

have data $(\mathbf{x}^k, \mathbf{y}^k)$, where $\mathbf{x}^k = \begin{bmatrix} x_1^k & \cdots & x_d^k \end{bmatrix}^T \in \mathbb{R}^d$ are the d explanatory variables and $\mathbf{y}^k = \begin{bmatrix} y_1^k & \cdots & y_m^k \end{bmatrix}^T$ represents the response variable. The element $y_i^k = 1$ if the observation corresponds to category i and zero otherwise. For each category $i \in \{1, \ldots, m\}$ we define a corresponding unknown weight vector $\mathbf{w}_i \in \mathbb{R}^d$, and define $\mathbf{w} \in \mathbb{R}^{d(m-1)}$ as the concatenation of the individual categories' weight vectors:

$$\mathbf{w} = \begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \\ \vdots \\ \mathbf{w}_{m-1} \end{bmatrix},$$

where without loss of generality, as explained below, we set $\mathbf{w}_m = \mathbf{0}$.

Under a multinomial logistic regression model, the probability that $\mathbf{x}^k$ corresponds to category i is written as

$$\mathbb{P}\left(y_i^k = 1 | \mathbf{x}^k, \mathbf{w}\right) = \frac{\exp\left(\mathbf{w}_i^T \mathbf{x}^k\right)}{\sum_{j=1}^m \exp\left(\mathbf{w}_j^T \mathbf{x}^k\right)}, \tag{6.4}$$

for $i \in \{1, \ldots, m\}$, where $\mathbf{w}_i$ is the weight vector corresponding to category i. In this model, the weight vector changes from category to category and the explanatory variables $\mathbf{x}^k$ are held fixed over all categories. Because of the normalization condition

$$\sum_{i=1}^m \mathbb{P}\left(y_i^k = 1 | \mathbf{x}^k, \mathbf{w}\right) = 1,$$

the weight vector for one of the categories need not be estimated. Without loss of generality, we thus set $\mathbf{w}_m = 0$ and the only parameters to be learned are the weight vectors $\mathbf{w}_i$ for $i \in \{1, \ldots, m-1\}$. In the remainder of the chapter, we use $\mathbf{w}$ as defined above to denote the $(d(m-1))$-dimensional vector of parameters to be learned.

6.2.2 Softmax decision models

We now make the connection between multinomial logistic regression and the softmax decision-making model (6.1). We let $d = m \cdot n_{obj}$, so there are n_{obj} explanatory variables for each of the m categories. Consider a single observation k with data $(\mathbf{x}^k, \mathbf{y}^k)$, where $\mathbf{x}^k \in \mathbb{R}^d$ is partitioned into m blocks, each of length n_{obj}:

$$\mathbf{x}^k = [\mathbf{x}_1^k; \mathbf{x}_2^k; \cdots \mathbf{x}_m^k].$$

Motivated by models of decision making like stochastic UCL, we consider a variant of the multinomial logistic regression model (6.4) with the following structure:

$$\mathbb{P}\left(y_i^k = 1 | \mathbf{x}^k, \boldsymbol{\theta}\right) = \frac{\exp\left(\boldsymbol{\theta}^T \mathbf{x}_i^k\right)}{\sum_{j=1}^m \exp\left(\boldsymbol{\theta}^T \mathbf{x}_j^k\right)} \tag{6.5}$$

for $i \in \{1, \ldots, m\}$, where $\boldsymbol{\theta} \in \mathbb{R}^o$ is a weight vector that is the same for all categories and $\mathbf{x}_i^k \in \mathbb{R}^{n_{obj}}$ is the subset of the explanatory variables that correspond to category i. This is the softmax decision-making model with linear objective function (6.1) introduced above. When referring to this variant of the model, we use the word *option* instead of the word category to emphasize the connection to decision making.

The model (6.5) can be related to the model (6.4) more commonly studied in the literature as follows. Denote the unique weights by $\boldsymbol{\theta} \in \mathbb{R}^{n_{obj}}$ and the vector $\mathbf{x}^k$ as above. Then the weight vectors $\mathbf{w}_i$ are given by $\mathbf{w}_i = \phi_i \otimes \boldsymbol{\theta}$, where $\phi_i \in \mathbb{R}^m$ is the indicator vector with $(\phi_i)_j = \delta_{ij}$ and $\otimes$ is the Kronecker product of two matrices. For model (6.5) the estimation procedure needs only learn the n_{obj}-dimensional parameter $\boldsymbol{\theta}$. Note that the redundancy in parameters that led to setting $\mathbf{w}_m = 0$ has been transformed into a redundancy in explanatory variables, which could be dealt with by, e.g., subtracting $\mathbf{x}_m^k$ from each set of explanatory variables for each observation k. In the following, we will assume this transformation has been made. Concretely, if the original data is given by $\tilde{\mathbf{x}}^k = [\tilde{\mathbf{x}}_1^k; \tilde{\mathbf{x}}_2^k; \cdots \tilde{\mathbf{x}}_m^k]$, then $\mathbf{x}^k$ is the transformed data $\mathbf{x}^k = [\tilde{\mathbf{x}}_1^k - \tilde{\mathbf{x}}_m^k; \tilde{\mathbf{x}}_2^k - \tilde{\mathbf{x}}_m^k; \cdots \tilde{\mathbf{x}}_{m-1}^k - \tilde{\mathbf{x}}_m^k; \mathbf{0}]$.

The new model (6.5) is dual to the standard model in the sense that the dependence on categories i is transferred from the weights $\mathbf{w}_i$ to the explanatory variables $\mathbf{x}_i$. This duality means the algorithms developed in [59] are not applicable to such a model, though the general framework applies. In Section 6.3 we pose the parameter estimation problem for the model (6.5) and review the bound optimization framework that [59] used to develop parameter estimation algorithms for the multinomial logistic regression model (6.4). In Section 6.5 we apply this framework to develop a new algorithm for learning the vector parameters $\boldsymbol{\theta}$ of the softmax model (6.5).

6.3 Parameter estimation for softmax decision-making models

In this section, we define the parameter estimation problem for multinomial logistic regression and review relevant results from the literature. In particular we review the bound optimization approach, which was used in [59], and standard convergence results for maxi-

mum likelihood estimators. The bound optimization approach provides an efficient way to solve the parameter estimation problem, and several results from the econometrics literature provide conditions under which the estimator is guaranteed to converge asymptotically. The development in Sections 6.3.1–6.3.3 largely follows Sections 2 and 3 of [59], while Section 6.3.4 summarizes results from the standard econometrics reference [86].

6.3.1 The parameter estimation problem

In the parameter estimation problem for the softmax decision-making model (6.5), we wish to estimate the values of $\boldsymbol{\theta}$ based on the observed data $(\mathbf{x}^k, \mathbf{y}^k)$. A standard way to perform parameter estimation is using the maximum likelihood method [50, Chapter 7]. To perform maximum likelihood (ML) estimation of $\boldsymbol{\theta}$, one maximizes the log-likelihood function,

$$\ell(\boldsymbol{\theta}) = \sum_{k=1}^{n} \log \mathbb{P}\left(\mathbf{y}^k | \mathbf{x}^k, \boldsymbol{\theta}\right) \tag{6.6}$$

$$= \sum_{k=1}^{n} \left[\sum_{i=1}^{m} y_i^k \boldsymbol{\theta}_i^T \mathbf{x}^k - \log \sum_{i=1}^{m} \exp\left(\boldsymbol{\theta}_i^T \mathbf{x}^k\right) \right], \tag{6.7}$$

yielding the ML estimate

$$\hat{\boldsymbol{\theta}}_{ML} = \arg\max_{\boldsymbol{\theta}} \ell(\boldsymbol{\theta}). \tag{6.8}$$

This estimate can be interpreted as the parameter value that makes the observed data most likely under the given model. In the following, we occasionally write $\ell(\boldsymbol{\theta}; \mathbf{x})$ to emphasize that ℓ is a function of $\boldsymbol{\theta}$ that depends on the data $\mathbf{x}$. The data can be thought of as fixed parameters of the function ℓ.

A prior on $\boldsymbol{\theta}$ can be incorporated by adopting a maximum a posteriori (MAP) estimate [50, Chapter 11],

$$\hat{\boldsymbol{\theta}}_{MAP} = \arg\max_{\boldsymbol{\theta}} L(\boldsymbol{\theta}) = \arg\max_{\boldsymbol{\theta}} [\ell(\boldsymbol{\theta}) + \log p(\boldsymbol{\theta})], \tag{6.9}$$

with $p(\boldsymbol{\theta})$ being the prior on $\boldsymbol{\theta}$. The MAP estimate penalizes ML estimates that are considered unlikely under the prior.

6.3.2 Bound optimization algorithms

The optimization problem (6.9) can be solved by a variety of methods, for example Iteratively Reweighted Least Squares [85]. However, bound optimization algorithms can provide a faster solution. In [59], the authors develop fast bound optimization algorithms for MAP

parameter estimation of the standard multinomial logistic regression model (6.4). In Section 6.5 we develop a similar algorithm for parameter estimation of the model (6.5) with Kronecker product structure. In this section, we review the basic theory of bound optimization algorithms following [59] and [63].

Bound optimization algorithms solve the optimization problem (6.9) by iteratively maximizing a *surrogate function* V [63]:[2]

$$\hat{\boldsymbol{\theta}}^{(t+1)} = \arg\max_{\boldsymbol{\theta}} V\left(\boldsymbol{\theta}|\hat{\boldsymbol{\theta}}^{(t)}\right). \tag{6.10}$$

Then $L(\hat{\boldsymbol{\theta}}^{(t+1)}) \geq L(\hat{\boldsymbol{\theta}}^{(t)})$ if V satisfies the following *key condition*: $L(\boldsymbol{\theta}) - V(\boldsymbol{\theta}|\hat{\boldsymbol{\theta}}^{(t)})$ attains its minimum when $\boldsymbol{\theta} = \hat{\boldsymbol{\theta}}^{(t)}$. The proof is as follows:

$$\begin{aligned}
L(\hat{\boldsymbol{\theta}}^{(t+1)}) &= L(\hat{\boldsymbol{\theta}}^{(t+1)}) - V(\hat{\boldsymbol{\theta}}^{(t+1)}|\hat{\boldsymbol{\theta}}^{(t)}) + V(\hat{\boldsymbol{\theta}}^{(t+1)}|\hat{\boldsymbol{\theta}}^{(t)}) \\
&\geq L(\hat{\boldsymbol{\theta}}^{(t)}) - V(\hat{\boldsymbol{\theta}}^{(t)}|\hat{\boldsymbol{\theta}}^{(t)}) + V(\hat{\boldsymbol{\theta}}^{(t+1)}|\hat{\boldsymbol{\theta}}^{(t)}) \\
&\geq L(\hat{\boldsymbol{\theta}}^{(t)}) - V(\hat{\boldsymbol{\theta}}^{(t)}|\hat{\boldsymbol{\theta}}^{(t)}) + V(\hat{\boldsymbol{\theta}}^{(t)}|\hat{\boldsymbol{\theta}}^{(t)}) \\
&= L(\hat{\boldsymbol{\theta}}^{(t)}).
\end{aligned}$$

The standard Expectation Maximization algorithm [79] for maximum likelihood estimation with missing data is a special case of this approach, where the key condition comes from using Jensen's inequality on the likelihood function. The bound optimization approach allows us to derive surrogate functions using purely analytical means, without recourse to the concept of missing data. The iterative update procedure guarantees that $\hat{\boldsymbol{\theta}}^{(t)}$ converges to a local maximum of $L(\boldsymbol{\theta})$. If $L(\boldsymbol{\theta})$ is concave, this local maximum is in fact the desired global maximum. When $L(\boldsymbol{\theta})$ is concave, an analytical method for obtaining a surrogate function $V(\boldsymbol{\theta}|\hat{\boldsymbol{\theta}})$ is by using a bound on the Hessian $\mathbf{H}(\boldsymbol{\theta})$ of $L(\boldsymbol{\theta})$.

For two square matrices A, B of the same size, let $A \succeq B$ denote that $A - B$ is positive semidefinite. If the Hessian $\mathbf{H}$ of $L(\boldsymbol{\theta})$ is lower bounded, i.e., if there exists a negative definite matrix $\boldsymbol{B}$ such that $\mathbf{H}(\boldsymbol{\theta}) \succeq \boldsymbol{B}$ for all values of $\boldsymbol{\theta}$, then it is easy to prove that, for any $\boldsymbol{\theta}'$,

$$L(\boldsymbol{\theta}) \geq L(\boldsymbol{\theta}') + (\boldsymbol{\theta} - \boldsymbol{\theta}')^T g(\boldsymbol{\theta}') + \frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\theta}')^T \boldsymbol{B}(\boldsymbol{\theta} - \boldsymbol{\theta}'),$$

where $\mathbf{g}(\boldsymbol{\theta}')$ denotes the gradient of $L(\boldsymbol{\theta})$ computed at $\boldsymbol{\theta}'$. Define $V(\boldsymbol{\theta}|\boldsymbol{\theta}') = L(\boldsymbol{\theta}') + (\boldsymbol{\theta} - \boldsymbol{\theta}')^T \mathbf{g}(\boldsymbol{\theta}') + \frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\theta}')^T \boldsymbol{B}(\boldsymbol{\theta} - \boldsymbol{\theta}')$. Then we have $L(\boldsymbol{\theta}) - V(\boldsymbol{\theta}|\boldsymbol{\theta}') \geq 0$, with equality if and

[2] References [59] and [63] denote the surrogate function by Q, but here we use V to avoid confusion with the objective function Q_i.

only if $\boldsymbol{\theta} = \boldsymbol{\theta}'$. Therefore, $V(\boldsymbol{\theta}|\boldsymbol{\theta}')$ is a valid surrogate function and we can let

$$V(\boldsymbol{\theta}|\hat{\boldsymbol{\theta}}^{(t)}) = \boldsymbol{\theta}^T \mathbf{g}(\hat{\boldsymbol{\theta}}^{(t)}) - \boldsymbol{\theta}^T \boldsymbol{B}\hat{\boldsymbol{\theta}}^{(t)} + \frac{1}{2}\boldsymbol{\theta}^T \boldsymbol{B}\boldsymbol{\theta}, \tag{6.11}$$

where we have omitted terms that are constants as a function of $\boldsymbol{\theta}$ since they have no effect on the maximization step.

In the case of ML estimation, $L(\boldsymbol{\theta})$ is simply the likelihood function $\ell(\boldsymbol{\theta})$ (6.6). Maximizing $V(\boldsymbol{\theta}|\hat{\boldsymbol{\theta}}^{(t)})$, we get the iterative update equation

$$\hat{\boldsymbol{\theta}}^{(t+1)} = \hat{\boldsymbol{\theta}}^{(t)} - \boldsymbol{B}^{-1}\mathbf{g}(\hat{\boldsymbol{\theta}}^{(t)}), \tag{6.12}$$

where $\mathbf{g}(\boldsymbol{\theta})$ is simply the gradient of the likelihood function evaluated at $\boldsymbol{\theta}$.

6.3.3 Prior for MAP estimation

To apply a Bayesian approach to the estimation problem, we consider a prior on the parameter $\boldsymbol{\theta}$ and perform maximum a posteriori estimation. Let the prior on $\boldsymbol{\theta}$ be Gaussian with mean $\bar{\boldsymbol{\theta}}$ and precision λ, i.e.,

$$p(\boldsymbol{\theta}) \propto \exp\left(\frac{\lambda}{2}\|\boldsymbol{\theta} - \bar{\boldsymbol{\theta}}\|_2^2\right).$$

The mean $\bar{\boldsymbol{\theta}}$ encodes the average value of the prior belief, and the parameter $\lambda > 0$ encodes the strength of that belief, i.e., the scaling of the penalty for deviating from the prior mean belief.

Then the objective function is

$$L(\boldsymbol{\theta}) = \ell(\boldsymbol{\theta}) - \frac{\lambda}{2}\|\boldsymbol{\theta} - \bar{\boldsymbol{\theta}}\|_2^2, \tag{6.13}$$

where $\|\cdot\|_2^2$ is the squared Euclidean norm. This requires the straightforward modification of the update equation (6.12):

$$\hat{\boldsymbol{\theta}}^{(t+1)} = \left(\boldsymbol{B} - \frac{\lambda}{2}I\right)^{-1}\left(\boldsymbol{B}\hat{\boldsymbol{\theta}}^{(t)} - \frac{\lambda}{2}\bar{\boldsymbol{\theta}} - \mathbf{g}(\hat{\boldsymbol{\theta}}^{(t)})\right), \tag{6.14}$$

where I is the identity matrix. The factors $\left(\boldsymbol{B} - \frac{\lambda}{2}I\right)^{-1}\boldsymbol{B}$ and $\left(\boldsymbol{B} - \frac{\lambda}{2}I\right)^{-1}$ can be precomputed once and stored, so each iterative update is computationally inexpensive and fast.

While we have only considered a Gaussian prior here, these methods can be extended to other priors, for example the sparsity-inducing Laplacian prior

$$p(\boldsymbol{\theta}) \propto \exp\left(-\lambda \|\boldsymbol{\theta}\|_1\right),$$

where $\|\boldsymbol{\theta}\|_1 = \sum_i |w_i|$ denotes the l_1 norm and λ again denotes the strength of the prior. See [59] for more details of how to extend the bound optimization approach to multinomial logistic regression with a Laplacian prior.

6.3.4 Asymptotic behavior of the ML estimator

The ML estimator $\hat{\boldsymbol{\theta}}_{ML}$ solves the estimation problem in the frequentist framework, which posits that there is a true value $\boldsymbol{\theta}_0$ of the parameters that we attempt to recover from analyzing the given data. In this framework, natural questions to be asked are 1) does $\hat{\boldsymbol{\theta}}_{ML} \to \boldsymbol{\theta}_0$ as the number of observations n grows, and 2) how dispersed is the difference $\hat{\boldsymbol{\theta}}_{ML} - \boldsymbol{\theta}_0$? These questions have been studied in the literature, for which [86] is a standard reference. The remainder of this section summarizes the relevant results from [86]. The answers to these two questions depend on two properties of the model, identification and concavity, defined as follows.

Definition 6.1 (Identification)**.** *A statistical model with likelihood function $\ell : \mathbb{R}^q \to \mathbb{R}$ and observed data $\mathbf{x}$ is said to be* identified *if, for all $\boldsymbol{\theta}, \boldsymbol{\theta}_0 \in \mathbb{R}^q$,*

$$\boldsymbol{\theta} \neq \boldsymbol{\theta}_0 \Rightarrow \ell(\boldsymbol{\theta}_0; \mathbf{x}) \neq \ell(\boldsymbol{\theta}; \mathbf{x}).$$

Definition 6.2 (Concavity)**.** *A statistical model with likelihood function $\ell : \mathbb{R}^q \to \mathbb{R}$ is said to be* concave *if $\ell(\boldsymbol{\theta}; \mathbf{x})$ is strictly concave in $\boldsymbol{\theta}$.*

If a model is identified and concave (see [86, Theorem 2.7] for details), the answer to question 1) is yes. These two conditions imply that the true value $\boldsymbol{\theta}_0$ of the parameter is the unique maximum of the log-likelihood $\ell(\boldsymbol{\theta})$.

Concavity is a property only of the functional form of $\ell(\boldsymbol{\theta}; \mathbf{x})$, but identification may depend on the observed data $\mathbf{x}$. As an example of how a model may fail to be identified due to data, consider the model (6.5) with $\mathbf{x}_i$ being the zero vector for each i. In this case, $\mathbb{P}(y_i = 1 | \mathbf{x}, \theta) = 1/m$ for each i independent of $\boldsymbol{\theta}$ and the estimation procedure will be unable to distinguish among the possible parameter values.

In the following sections, we show that the functional form (6.6) of $\ell(\boldsymbol{\theta}; \mathbf{x})$ ensures weak concavity and provide conditions on the data $\mathbf{x}$ that ensure identification. These condi-

tions also ensure that $\ell(\boldsymbol{\theta}; \mathbf{x})$ is strictly concave and are useful guidelines for the design of experiments for estimating $\boldsymbol{\theta}$.

The answer to question 2) is that, under mild regularity conditions, the distribution of $\hat{\boldsymbol{\theta}}_{ML}$ approaches a normal distribution as the number of samples n grows. In particular, the following limit holds:

$$\hat{\boldsymbol{\theta}}_{ML} \xrightarrow{d} \mathcal{N}(\boldsymbol{\theta}_0, J^{-1}/n), \tag{6.15}$$

where $\xrightarrow{d}$ signifies a limit in distribution as $n \to \infty$ and $\mathbf{J} = -\mathbb{E}\left[\mathbf{H}(\boldsymbol{\theta}_0)\right]$ is the negative of the expected value of the Hessian. See [41, Chapter 9] for more details about the concept of a limit in distribution and see [86, Theorem 3.3] for full details of the conditions under which (6.15) holds. In practice one uses $\hat{\mathbf{J}} = -\mathbf{H}(\hat{\boldsymbol{\theta}}_{ML})/n$ as an estimate of $\mathbf{J}$. This permits construction of standard frequentist analysis tools, such as confidence intervals for the parameter estimates and hypothesis tests. The estimate $\hat{\boldsymbol{\theta}}_{ML}$ is efficient [50, Theorem 7.3] in the sense that it obeys the Cramér-Rao lower bound on the variance of estimators $\hat{\boldsymbol{\theta}}$, so no other unbiased estimator can have lower variance than $\hat{\boldsymbol{\theta}}_{ML}$.

6.4 Several examples of softmax decision-making models with linear objective functions

In this section, we provide several concrete examples of the softmax decision model (6.5). The goal is to make the connection between this functional form and others that appear in the literature.

Example 1 (Softmax with unknown temperature)**.** *A standard decision model in reinforcement learning [124] is the so-called softmax action selection rule, which selects an option i with probability*

$$\mathbb{P}(i) = \frac{\exp(V_i/\tau)}{\sum_{j=1}^{n} \exp(V_j/\tau)},$$

where V_i is the value associated with option i and τ is a positive parameter known as the temperature*. This rule selects options stochastically, preferentially selecting those with higher values. The degree of stochasticity is controlled by the temperature τ: in the limit $\tau \to 0^+$, the rule reduces to the standard maximum and deterministically selects the option with the highest value of V_i, while in the limit $\tau \to +\infty$, all options are equally probable and the rule selects options according to a uniform distribution.*

This model is clearly similar to (6.5) *with $n_{obj} = 1$. Specifically, assume that the temperature τ is constant but unknown, and the values V_i are known. Then the two models are*

identical if we identify

$$\theta = 1/\tau,\ \mathbf{x}_i = V_i.$$

In the reinforcement learning literature, the quantity $1/\tau$ *is sometimes known as the* inverse temperature *and referred to by the symbol* β*. Applying our methods allows one to estimate* $\theta = 1/\tau$ *from observed choice data.*

Example 2 (Softmax with known cooling schedule form)**.** *A slightly more complicated model might let the softmax temperature* τ *follow a known functional form, called a* cooling schedule*, that depends on an unknown parameter. For example, in simulated annealing, Mitra et al. [77] showed that good cooling schedules follow a logarithmic functional form:*

$$\tau(t) = \frac{\nu}{\log t},$$

where t *is the decision index and* $\nu > 0$ *is a parameter.*

In Chapter 4 we consider tuning rules to dynamically adjust ν *and show that there exists a tuning rule that results in stochastic UCL achieving logarithmic regret. If, however,* ν *is constant but unknown, this model can be represented in the form of* (6.5) *with* $n_{obj} = 1$ *by identifying*

$$\theta = 1/\nu,\ \mathbf{x}_i = V_i \log t.$$

Example 3 (Softmax Q-learning with unknown temperature and learning rate)**.** *According to a simple* Q*-learning model [128], for each choice* t *the agent assigns an expected value* V_i^t *to each option* i*. The values are initialized to 0 at* $t = 1$ *and then for each trial, the agent picks option* i_t*, receives reward* r_t*, and updates the value of the chosen option* i_t *according to*

$$V_{i_t}^{t+1} = V_{i_t}^t + \alpha\delta_t,$$

where $\alpha \in [0, 1]$ *is a free parameter called the* learning rate *and* $\delta_t = r_t - V_{i_t}^t$ *is the* prediction error*.*

A common model in reinforcement learning [33] is for the agent to make decisions using a softmax rule on the value function V_i^t*, so the probability of selecting an option* i *at time* t *is*

$$\mathbb{P}\left(i_t = i\right) = \frac{\exp\left(V_i^t/\tau\right)}{\sum_{j=1}^n \exp\left(V_j^t/\tau\right)} = \frac{\exp\left(V_i^{t-1}/\tau + \alpha\delta_{t-1}/\tau\right)}{\sum_{j=1}^n \exp\left(V_j^{t-1}/\tau + \alpha\delta_{t-1}/\tau\right)}.$$

Similar models are used in the analysis of fMRI data, e.g. [132]. If V_i^{t-1}, V_i^{t-2}*, and* r_t *are known while* τ *and* α *are unknown, this is in the form of* (6.5) *with* $n_{obj} = 2$ *and identifying*

$$\boldsymbol{\theta} = \begin{bmatrix} 1/\tau; & \alpha/\tau \end{bmatrix},\ \mathbf{x}_i = \begin{bmatrix} V_i^{t-1}; & \delta_{t-1} \end{bmatrix}.$$

If only the initial value $V_i^{t=1} = 0$ is known, then the value function V_i^t becomes a nonlinear function of the parameters α, τ and the model is not of the form (6.5)*, although it may be possible to find a transformation that puts it in such a form.*

6.5 A fast iterative algorithm for softmax decision models with linear objective functions

In this section we develop fast iterative algorithms for multinomial logistic regression with Kronecker product structure (6.5) by specifying the form of the likelihood gradient $\mathbf{g}(\boldsymbol{\theta})$ and the bound $\boldsymbol{B}$ on the Hessian in the update equations (6.12) and (6.14). This study of the Hessian leads us to prove Theorem 6.11, which provides conditions under which the likelihood function $\ell(\boldsymbol{\theta})$ is concave, which implies the convergence of the ML estimator and reduces the maximization problem (6.8) to a convex optimization problem. When (6.8) is a convex optimization problem, it can be solved by a variety of standard methods, such as the various variants of Newton's method. See [17] for a standard textbook on convex optimization theory and methods. In Section 6.5.4 we develop an alternative iterative algorithm for solving (6.8) that may be faster than standard convex optimization methods in some cases.

6.5.1 Likelihood function

Considering the log-likelihood function (6.6) of the model (6.5) for just one observation $(\mathbf{y}^k, \mathbf{x}^k)$, we have

$$\begin{aligned} \ell(\boldsymbol{\theta}) &= \sum_{i=1}^{m} y_i^k \boldsymbol{\theta}_i^T \mathbf{x}^k - \log \sum_{i=1}^{m} \exp\left(\boldsymbol{\theta}_i^T \mathbf{x}^k\right) \\ &= \sum_{i=1}^{m} y_i^k \left(\phi_i \otimes \boldsymbol{\theta}\right)^T \mathbf{x}^k \\ &\quad - \log \sum_{i=1}^{m} \exp\left(\left(\phi_i \otimes \boldsymbol{\theta}\right)^T \mathbf{x}^k\right) \end{aligned} \tag{6.16}$$

The gradient of $\ell(\boldsymbol{\theta})$ is easily computed:

$$\mathbf{g}(\boldsymbol{\theta}) = \nabla \ell(\boldsymbol{\theta}) = \sum_{i=1}^{m} \left(y_i^k - p_i^k\right) \left(\phi_i \otimes I_{n_{obj}}\right) \mathbf{x}^k, \tag{6.17}$$

where p_i^k is defined as

$$p_i^k = p_i^k(\boldsymbol{\theta}) = \mathbb{P}\left(y_i^k = 1 | \mathbf{x}^k, \boldsymbol{\theta}\right) \tag{6.18}$$

and $I_{n_{obj}}$ is the n_{obj}-dimensional identity matrix.

6.5.2 Two operations for block matrices

The algorithm requires a lower bound $\boldsymbol{B}$ on the Hessian of the likelihood function, and convergence of the algorithm requires that the Hessian be negative-definite, i.e. that the Hessian be upper bounded by the zero matrix. To study the Hessian, we require some additional notation. We define two operations on block matrices: one a generalization of the Hadamard product (often called the Schur product, [46, Section 7.5]), and another a block contraction, and prove several properties of these operations.

We begin with the following generalization of the Hadamard product to block matrices:

Definition 6.3. *Let n, m, p and q be positive integers. Let A be an $n \times m$ real-valued matrix, and let B be an $np \times mq$ real-valued block matrix, where each block is of size $p \times q$. Denote the i, j element of A as a_{ij} and the i, j block of B as B_{ij}. Then the block Hadamard product $A \odot B$ is defined as the $np \times mq$ block matrix whose i, j block is $a_{ij}B_{ij}$. That is,*

$$(A \odot B)_{ij} = a_{ij}B_{ij}.$$

For two matrices A and B of equal size, the Hadamard (or element-wise) product $A \circ B$ is defined as the element-by-element product, i.e., $(A \circ B)_{ij} = a_{ij}b_{ij}$. The block Hadamard product can be thought of as an analogy of the Hadamard product in the case where one of the two matrices is a block matrix and the other is of conformable size. The block Hadamard product is a special case of the Khatri-Rao product $A * B$ [52, 72], defined as follows in the case where both A and B are block matrices:

$$(A * B)_{ij} = A_{ij} \otimes B_{ij},$$

where the i, j block of the product is the $m_i p_i \times n_j q_j$-sized Kronecker product of the corresponding blocks of A and B.

The Schur product theorem [106],[46, Theorem 7.5.3] states that the Hadamard product of two positive definite (semidefinite) matrices is positive definite (semidefinite). Liu [71] proved an analogous result for the Khatri-Rao product:

Theorem 6.4 ([71], Theorem 5)**.** *Let $M \succeq P \succeq 0$, $N \succeq Q \succeq 0$, and M, P, N and Q be compatibly partitioned matrices. Then*

$$M * N \succeq P * Q \succeq 0. \tag{6.19}$$

This result is somewhat more general than the Schur product theorem because it not only shows that the product is positive semidefinite, but also that the product preserves ordering, as shown by the first inequality in (6.19). Since the block Hadamard product is a special case of the Khatri-Rao product, it obeys Theorem 6.4. In particular, the following holds.

Corollary 6.5. *Let A be a positive definite (semidefinite) matrix of size n, and let B be a positive definite (semidefinite) block matrix of size mn, where each block is square and of size m. Then their block Hadamard product $A \odot B$ is positive definite (semidefinite), i.e., $A \odot B \succeq 0$.*

This result can be proved by applying Theorem 6.4. We develop a second, more direct proof using an argument analogous to the proof of the Schur product theorem as follows.

Proof. Let $\mathbf{x}, \mathbf{y} \in \mathbb{R}^{mn}$ be considered as block vectors with each block of length m and denote block i of each vector as $\mathbf{x}_i, \mathbf{y}_i$, respectively. Consider the quantity

$$\mathbf{x}^T (A \odot B)\mathbf{y} = \sum_{i,j=1}^{n} \mathbf{x}_i^T a_{ij} B_{ij} \mathbf{y}_j = \sum_{i,j=1}^{n} a_{ij} \mathbf{x}_i^T B_{ij} \mathbf{y}_j.$$

This can be rewritten as

$$\sum_{i,j=1}^{n} a_{ij} (\operatorname{diag}(\mathbf{x}^T) B \operatorname{diag}(\mathbf{y}))_{ij},$$

where $\operatorname{diag}(\mathbf{x}^T)$ is the $n \times nm$ block diagonal matrix with $\mathbf{x}_i^T$ as the i, i block, and $\operatorname{diag}(\mathbf{y})$ is defined similarly. Using the formula for the trace of a product, this can in turn be rewritten as

$$\begin{aligned} &\operatorname{tr}\left(A(\operatorname{diag}(\mathbf{x}^T) B \operatorname{diag}(\mathbf{y})\right) \\ =&\operatorname{tr}\left(A^{1/2} A^{1/2} (\operatorname{diag}(\mathbf{x}^T) B^{1/2} B^{1/2} \operatorname{diag}(\mathbf{y})\right) \\ =&\operatorname{tr}\left(A^{1/2} (\operatorname{diag}(\mathbf{x}^T) B^{1/2} B^{1/2} \operatorname{diag}(\mathbf{y}) A^{1/2}\right), \end{aligned} \tag{6.20}$$

where the first equality is valid since the positive (semi)definiteness of A and B allows us to define the square roots, and the second equality is due to the cyclic property of the trace.

If $\mathbf{x} = \mathbf{y}$, (6.20) can be rewritten as $\operatorname{tr}\left(C^T C\right)$, where $C = A^{1/2} \operatorname{diag}(\mathbf{x}) B^{1/2}$. It is therefore the sum of C_{ij}^2 and thus non-negative, and we have

$$\mathbf{x}^T (A \odot B)\mathbf{x} \geq 0.$$

If A and B are strictly positive definite, then equality holds if and only if $\mathbf{x} = 0$. □

We define the following operation for block matrices:

Definition 6.6. *Let n, m, p and q be positive integers, and let A be the $np \times mq$ block matrix where each block is of size $p \times q$. Denote the i, j block of A by A_{ij}. Then* $\operatorname{sum}(A)$ *is the $p \times q$ matrix defined by*

$$\operatorname{sum}(A) = \sum_{i=1}^{n} \sum_{j=1}^{m} A_{ij}.$$

This is the block-wise sum (or contraction) of the matrix A.

Recall that for two $n \times n$ matrices A and B, $A \succeq B$ denotes that $A - B$ is positive semidefinite. A similar relationship holds for matrices related via the block Hadamard product and block-wise sum, as specified in the following lemma:

Lemma 6.7. *Let A and B be square matrices of size n such that $a_{ij} \leq b_{ij}\ \forall\ i, j \in \{1, \ldots, n\}$, i.e., each element of B is at least as large as the corresponding element of A. Let C be a square block matrix of size np, where each block is square of size p, and let C be nonnegative definite. Then,*

$$\operatorname{sum}(A \odot C) \preceq \operatorname{sum}(B \odot C).$$

Proof. Let $\mathbf{z} \in \mathbb{R}^p$ and $\tilde{\mathbf{z}} = [\mathbf{z}; \mathbf{z}; \cdots \mathbf{z}] \in \mathbb{R}^{np}$. Define $M = \operatorname{sum}(B \odot C) - \operatorname{sum}(A \odot C)$, and define $\Delta = \min_{i,j}(b_{ij} - a_{ij})$. Note that $\operatorname{sum}(A \odot C) \preceq \operatorname{sum}(B \odot C) \Leftrightarrow M \succeq 0$. By the definition of M, we have

$$M = \sum_{i,j} (b_{ij} - a_{ij}) C_{ij}.$$

Then, the following sequence of inequalities holds:

$$\begin{aligned}
\mathbf{z}^T M \mathbf{z} &= \sum_{i=1}^{n} \sum_{j=1}^{m} (b_{ij} - a_{ij}) \mathbf{z}^T C_{ij} \mathbf{z} \\
&\geq \Delta \sum_{i=1}^{n} \sum_{j=1}^{m} \mathbf{z}^T C_{ij} \mathbf{z} \\
&= \Delta \tilde{\mathbf{z}}^T C \tilde{\mathbf{z}} \geq 0,
\end{aligned}$$

where the first inequality follows from the definition of Δ and the second from the fact that C is nonnegative definite. Therefore $M \succeq 0$ and $\operatorname{sum}(A \odot C) \preceq \operatorname{sum}(B \odot C)$. □

6.5.3 Hessian of the log-likelihood function

In this section, we consider the Hessian $\mathbf{H}(\boldsymbol{\theta})$ of the likelihood function (6.6) for a single observation $(\mathbf{y}^k, \mathbf{x}^k)$. We establish the upper and lower bounds on the Hessian of the like-

lihood function required to implement the iterative updates (6.12) and (6.14). To study the Hessian, we use the matrix operations introduced in the previous section. Recall that $\mathbf{x}_i^k = \left(\phi_i^T \otimes I_{n_{obj}}\right) \mathbf{x}^k$ is the subset of explanatory variables that correspond to option i.

Recall from (6.18) that $p_i^k = p_i^k(\boldsymbol{\theta})$ is the probability of selecting option i given parameters $\boldsymbol{\theta}$ and define

$$\boldsymbol{p}^k = \boldsymbol{p}^k(\boldsymbol{\theta}) = [p_1^k(\boldsymbol{\theta}), p_2^k(\boldsymbol{\theta}), \ldots, p_m^k(\boldsymbol{\theta})] \in \mathbb{R}^m, \tag{6.21}$$

$$\boldsymbol{P}^k = \boldsymbol{P}^k(\boldsymbol{\theta}) = \operatorname{diag}(\boldsymbol{p}^k(\boldsymbol{\theta})) \in \mathbb{R}^{m \times m}, \tag{6.22}$$

so $\boldsymbol{P}^k$ is the diagonal matrix with $\boldsymbol{p}^k$ on the diagonal. The likelihood function $\ell(\boldsymbol{\theta})$ defined in (6.16) has the Hessian matrix

$$\mathbf{H}(\boldsymbol{\theta}) = -\sum_{i=1}^{m} \left(p_i^k \mathbf{x}_i^k \otimes \mathbf{x}_i^k - p_i^k \sum_{j=1}^{m} \mathbf{x}_j^k \otimes \mathbf{x}_j^k \right).$$

Defining $A^k = \boldsymbol{P}^k - \boldsymbol{p}^{kT} \boldsymbol{p}^k$ and using the two operations introduced in the previous section, this can be rewritten as

$$\mathbf{H}(\boldsymbol{\theta}) = -\operatorname{sum}\left(A^k \odot \mathbf{x}^{kT} \mathbf{x}^k\right). \tag{6.23}$$

Define $\mathbf{1}$ as the vector where each element is equal to 1, and let $B = \mathbf{1}^T \mathbf{1} \in \mathbb{R}^{n_{obj} \times n_{obj}}$ be a matrix with each element equal to 1. In order to apply the bound optimization algorithm from Section 6.3.2 we require the likelihood function to be concave, and the Hessian to be lower bounded by a matrix $\boldsymbol{B}$ that is independent of the parameter $\boldsymbol{\theta}$. Setting

$$\boldsymbol{B} = -\operatorname{sum}\left(\mathbf{x}^{kT} \mathbf{x}^k\right), \tag{6.24}$$

these conditions are ensured by the following main theorem:

Theorem 6.8. *The Hessian matrix* $\mathbf{H}(\boldsymbol{\theta})$ *for a single observation* $(\mathbf{y}^k, \mathbf{x}^k)$ *given by* (6.23) *satisfies*

$$\boldsymbol{B} \preceq \mathbf{H}(\boldsymbol{\theta}) \preceq 0,$$

where $\boldsymbol{B} = -\operatorname{sum}\left(\mathbf{x}^{kT} \mathbf{x}^k\right)$.

In order to prove Theorem 6.8 we will use the following lemma.

Lemma 6.9. *The matrix* $A^k = \boldsymbol{P}^k - \boldsymbol{p}^{kT} \boldsymbol{p}^k$, *where* $\boldsymbol{p}^k$ *and* $\boldsymbol{P}^k$ *are defined in* (6.21) *and* (6.22), *respectively, is symmetric and obeys the following bounds:*

$$0 \preceq A^k \preceq (I_m - \mathbf{1}^T \mathbf{1}/m).$$

Proof. The matrix A^k is symmetric because it is the difference of two symmetric matrices.

We begin by proving the lower bound. Note that A^k is the covariance matrix of a multinomial distribution with one sample and probabilities $\boldsymbol{p}^k$ [35, pp. 134–136]. Therefore, for any vector $\mathbf{c} \in \mathbb{R}^{n_{obj}}$, the quantity $\mathbf{c}^T A^k \mathbf{c}$ represents the variance of a linear combination of random variables drawn from the multinomial distribution. Furthermore, variance is non-negative [102], so therefore $\mathbf{c}^T A^k \mathbf{c} \geq 0$ and the lower bound holds.

For a proof of the upper bound, see Lemma 2.2 of [14]. □

With these pieces in place, we can now prove Theorem 6.8.

Proof of Theorem 6.8. We begin with the lower bound. Picking $B = \mathbf{1}^T \mathbf{1}$ and applying Lemma 6.7, we find that $\mathbf{H}(\boldsymbol{\theta})$ is bounded below by

$$\begin{aligned} \mathbf{H}(\boldsymbol{\theta}) \succeq -\operatorname{sum}\left(B \odot \mathbf{x}^{kT}\mathbf{x}^k\right) &= -\operatorname{sum}\left(\mathbf{1}^T\mathbf{1} \odot \mathbf{x}^{kT}\mathbf{x}^k\right) \\ &= -\operatorname{sum}\left(\mathbf{x}^{kT}\mathbf{x}^k\right) = \boldsymbol{B}. \end{aligned}$$

For the upper bound, note that A^k is positive-semidefinite by Lemma 6.9 and that $\mathbf{x}^{kT}\mathbf{x}^k$ is positive-semidefinite by construction. Then apply Corollary 6.5 to conclude that the product $A^k \odot \mathbf{x}^{kT}\mathbf{x}^k$ is positive-semidefinite, and therefore

$$\mathbf{y}^T \left(A^k \odot \mathbf{x}^{kT}\mathbf{x}^k\right) \mathbf{y} \geq 0, \ \forall \ \mathbf{y} \in \mathbb{R}^d.$$

In particular, let $\mathbf{y} = [\tilde{\mathbf{y}}; \tilde{\mathbf{y}}; \cdots \tilde{\mathbf{y}}]$ and note that

$$\begin{aligned} 0 &\leq \mathbf{y}^T \left(A^k \odot \mathbf{x}^{kT}\mathbf{x}^k\right) \mathbf{y} \\ &= \sum_{i,j} \tilde{\mathbf{y}}^T \left(A^k \odot \mathbf{x}^{kT}\mathbf{x}^k\right) \tilde{\mathbf{y}} \\ &= \tilde{\mathbf{y}}^T \operatorname{sum}\left(A^k \odot \mathbf{x}^{kT}\mathbf{x}^k\right) \tilde{\mathbf{y}}. \end{aligned}$$

Therefore $\operatorname{sum}\left(A^k \odot \mathbf{x}^{kT}\mathbf{x}^k\right)$ is positive-semidefinite and $\mathbf{H}(\boldsymbol{\theta}) = -\operatorname{sum}\left(A^k \odot \mathbf{x}^{kT}\mathbf{x}^k\right)$ is negative-semidefinite. □

Theorem 6.8 shows that the log-likelihood function ℓ for a single observation $(\mathbf{y}^k, \mathbf{x}^k)$ given by (6.16) is weakly concave and that its Hessian is lower-bounded by a matrix $\boldsymbol{B}$ that is independent of the data $\mathbf{x}^k$. If $\boldsymbol{B}$ has a non-trivial null space, Theorem 6.8 cannot guarantee that $\mathbf{H}(\boldsymbol{\theta})$ is strictly negative definite. In such a case, $\ell(\boldsymbol{\theta})$ may have multiple maxima and a maximum likelihood estimator is not guaranteed to converge. In Lemma 6.10 below, we derive conditions such that $\mathbf{H}(\boldsymbol{\theta})$ is strictly negative definite, which implies the convergence of a maximum likelihood estimator.

When there are multiple observations $(\mathbf{y}^k, \mathbf{x}^k), k \in \{1, \ldots, n\}$, the log-likelihood function $\ell(\boldsymbol{\theta})$, its gradient $\mathbf{g}(\boldsymbol{\theta})$, its Hessian $\mathbf{H}(\boldsymbol{\theta})$, and the bound $\boldsymbol{B}$ are computed by summing the single observation expressions (6.16), (6.17), (6.23), and (6.24), respectively, over the observations k. Using $\mathbf{g}(\boldsymbol{\theta})$ and $\boldsymbol{B}$ we can apply the bound optimization approach from Equations (6.12) and (6.14).

6.5.4 Iterative algorithm

We now bring together the results of the previous sections to construct an iterative algorithm for solving the ML estimation problem (6.8) for softmax decision-making models (6.5). This algorithm is analogous to the one developed in [59] for the multinomial logistic regression model (6.4). The surrogate function approach (6.12) applied to the ML estimation problem (6.8) gives the simple update equation

$$\hat{\boldsymbol{\theta}}_{ML}^{(t+1)} = \hat{\boldsymbol{\theta}}_{ML}^{(t)} - \boldsymbol{B}^{-1}\mathbf{g}\left(\hat{\theta}_{ML}^{(t)}\right), \tag{6.25}$$

where $\mathbf{g}(\boldsymbol{\theta})$ is the gradient of the likelihood function (6.17) and $\boldsymbol{B}$ is the bound matrix (6.24). We define the *bound ML algorithm* for softmax decision-making models with linear objective functions (6.1) as the procedure that begins with an initial guess $\hat{\boldsymbol{\theta}}_{ML}^{(0)}$ and applies (6.25) until a desired convergence tolerance is achieved. Define $\hat{\boldsymbol{\theta}}_{ML}^{\infty}$ as the limit

$$\hat{\boldsymbol{\theta}}_{ML}^{\infty} = \lim_{t \to +\infty} \hat{\boldsymbol{\theta}}_{ML}^{(t)}.$$

We know from Section 6.3.2 that the update procedure will monotonically converge to a maximum of the likelihood function ℓ. In the following section we will prove conditions under which the bound ML algorithm converges to the true parameter value $\boldsymbol{\theta}_0$.

For the MAP estimation problem (6.9), the Gaussian prior p on $\boldsymbol{\theta}$ has mean $\bar{\boldsymbol{\theta}}$ and precision λ, i.e.,

$$p(\boldsymbol{\theta}) \propto \exp\left(\frac{\lambda}{2}\|\boldsymbol{\theta} - \bar{\boldsymbol{\theta}}\|_2^2\right).$$

The update equation (6.14) becomes

$$\hat{\boldsymbol{\theta}}_{MAP}^{(t+1)} = \left(\boldsymbol{B} - \frac{\lambda}{2} I_{n_{obj}}\right)^{-1} \left(\boldsymbol{B}\hat{\boldsymbol{\theta}}_{MAP}^{(t)} - \frac{\lambda}{2}\bar{\boldsymbol{\theta}} - \mathbf{g}\left(\hat{\boldsymbol{\theta}}_{MAP}^{(t)}\right)\right). \tag{6.26}$$

We define the *bound MAP algorithm* for softmax decision-making models with linear objective functions (6.1) as the procedure that begins with an initial guess $\hat{\boldsymbol{\theta}}_{MAP}^{(0)}$ and applies (6.26) until a desired convergence tolerance is achieved. As with the bound ML algorithm,

we define $\hat{\boldsymbol{\theta}}^{\infty}_{MAP}$ as the limit

$$\hat{\boldsymbol{\theta}}^{\infty}_{MAP} = \lim_{t \to +\infty} \hat{\boldsymbol{\theta}}^{(t)}_{MAP}.$$

6.5.5 Asymptotic and finite-sample behavior

Recall from Section 6.3.4 that two properties that guarantee asymptotic convergence of the ML estimator are identification and concavity. In Section 6.5.4 above we showed that the functional form (6.16) ensured that ℓ is weakly concave in $\boldsymbol{\theta}$. Whether or not the model (6.5) is strictly concave can be a function of the data $\mathbf{x}^k, k \in \{1, \ldots, n\}$. Recall our example where $\mathbf{x}^k_i = 0$ for each i and k. In this case the probability $\mathbb{P}\left(y^k_i | \mathbf{x}^k, \boldsymbol{\theta}\right) = 1/m$ for each i and k independent of $\boldsymbol{\theta}$, the bound matrix (6.24) is the zero matrix $\boldsymbol{B} = \mathbf{0}$ and the likelihood function is flat, so neither identification nor concavity is satisfied.

However, a sufficient condition for identification is as follows: Define the $n_{obj} \times m$ matrix $\mathbf{X}^k$ by transforming the explanatory variable $\mathbf{x}^k$ of a single observation k:

$$\mathbf{X}^k = [\mathbf{x}^k_1 \mathbf{x}^k_2 \cdots \mathbf{x}^k_{m-1} \mathbf{0}]. \tag{6.27}$$

Note that $\mathbf{X}^k \mathbf{X}^{kT} = \text{sum}(\mathbf{x}^{kT} \mathbf{x}^k)$. Considering $\mathbf{X}^k$ as a random variable, the following lemma ensures identification.

Lemma 6.10. *If the second-moment matrix* $\mathbb{E}\left[\mathbf{X}^k \mathbf{X}^{kT}\right]$ *exists and is positive definite, where* $\mathbf{X}^k$ *is defined by* (6.27)*, then the softmax model* (6.5) *is identified.*

Proof. The probability of choosing an option i under the model (6.5) is a monotonic function of the objective value Q_i, so it suffices to show that there exists a one-to-one mapping between the parameter vector $\boldsymbol{\theta}$ and the objective values Q_i.

Let $\boldsymbol{\theta}, \boldsymbol{\theta}' \in \mathbb{R}^{n_{obj}}$ and define the vectors of objective function values $\mathbf{Q} = \boldsymbol{\theta}^T \mathbf{X}^k$ and $\mathbf{Q}' = \boldsymbol{\theta}'^T \mathbf{X}^k$. Define $\Delta\mathbf{Q} = \mathbf{Q} - \mathbf{Q}' = (\boldsymbol{\theta} - \boldsymbol{\theta}')^T \mathbf{X}^k \in \mathbb{R}^m$. Then the magnitude of $\Delta\mathbf{Q}$ satisfies $\mathbb{E}\left[\|\Delta\mathbf{Q}\|^2\right] = (\boldsymbol{\theta} - \boldsymbol{\theta}')^T \mathbb{E}\left[\mathbf{X}^k \mathbf{X}^{kT}\right] (\boldsymbol{\theta} - \boldsymbol{\theta}')$. Then by the assumption that $\mathbb{E}\left[\mathbf{X}^k \mathbf{X}^{kT}\right]$ is positive definite, $\mathbb{E}\left[\|\Delta\mathbf{Q}\|^2\right] = 0$ implies $(\boldsymbol{\theta} - \boldsymbol{\theta}') = 0$, so $\boldsymbol{\theta} = \boldsymbol{\theta}'$ and $\mathbf{Q} = \mathbf{Q}'$. Therefore the mapping between the parameters $\boldsymbol{\theta}$ and the objective values Q_i is one-to-one, which implies that $\ell(\boldsymbol{\theta} | \mathbf{x}^k, \mathbf{y}^k) \neq \ell(\boldsymbol{\theta}' | \mathbf{x}^k, \mathbf{y}^k)$ and the softmax model (6.5) is identified. □

The condition of Lemma 6.10 is given in terms of an expectation, but in practice one has a given sample of data. In this case the expectation can be replaced by the sample average. Specifically, define $\mathbf{X}^k$ for each observation $k \in \{1, \ldots, n\}$ by transforming $\mathbf{x}^k$ according to

(6.27). Then $\mathbb{E}\left[\mathbf{X}^k\mathbf{X}^{kT}\right]$ is estimated by

$$\mathbb{E}\left[\mathbf{X}^k\mathbf{X}^{kT}\right] \approx \frac{1}{n}\sum_{k=1}^{n}\mathbf{X}^k\mathbf{X}^{kT}.$$

If this sample average is positive definite, then the model is identified. For the sample average to be positive definite it must be full rank $= n_{obj}$, and each observation k can add at most m to the rank, where m is the number of options. Therefore, the following inequality must be satisfied for the model to be identified:

$$mn \geq n_{obj}.$$

This gives a lower bound $n \geq \lceil n_{obj}/m \rceil$ on the minimum number of observations required for identification. For most applications, the number of options m will be larger than the number of parameters n_{obj}, so the lower bound is trivial, but for cases with large numbers of parameters the bound can be useful for experimental design.

The following theorem summarizes the conditions under which the ML estimator (6.8) converges.

Theorem 6.11 (Convergence of the ML estimator)**.** *If the second-moment matrix*

$$\frac{1}{n}\sum_{k=1}^{n}\mathbf{X}^k\mathbf{X}^{kT}$$

exists and is positive definite, where $\mathbf{X}$ *is defined by* (6.27)*, then the ML estimator* $\hat{\boldsymbol{\theta}}_{ML}$ *for* (6.5) *is asymptotically approximately distributed as*

$$\hat{\boldsymbol{\theta}}_{ML} \sim \mathcal{N}(\boldsymbol{\theta}_0, \hat{\mathbf{J}}^{-1}/n), \tag{6.28}$$

where $\hat{\mathbf{J}} = -\mathbf{H}(\hat{\boldsymbol{\theta}}_{ML})/n$ *is the Hessian "per observation" of the likelihood function evaluated at the estimated parameter value.*

Proof. By Theorem 6.8, the likelihood function is weakly concave, and by Lemma 6.10, the sample second-moment matrix being positive definite implies that the likelihood function is strictly concave and that the model (6.5) is both identified and concave. Therefore, by the results summarized in Section 6.3.4, $\hat{\boldsymbol{\theta}}_{ML}$ is asymptotically normally distributed and (6.28) holds. □

By Theorem 6.11, the ML estimator $\hat{\boldsymbol{\theta}}_{ML}$ is the unique maximum of ℓ, and will be asymptotically normally distributed around its true value $\boldsymbol{\theta}_0$. Since the bound ML algorithm

(6.25) monotonically increases the likelihood value $\ell\left(\hat{\boldsymbol{\theta}}_{ML}^{(t)}\right)$ of its iterative estimates, the algorithm asymptotically converges to $\hat{\boldsymbol{\theta}}_{ML}$. In finite samples where the model is identified, Equation (6.15) gives the approximate distribution of $\hat{\boldsymbol{\theta}}_{ML}$. This distribution can be used to formulate frequentist confidence intervals for the estimated parameter $\hat{\boldsymbol{\theta}}_{ML}$.

6.6 Numerical examples

In this section we present several numerical examples to demonstrate the theory and the bound ML algorithm (6.25) developed in the previous sections. In the process we also discuss several issues relating to implementation of the algorithm.

6.6.1 Scalar parameter

First, we consider an instance of problem (6.5) with $m = 10$ options and where the parameter $\boldsymbol{\theta}$ is a scalar with $\theta_0 = 4$. This corresponds to Example 1 above, where a decision-maker is making choices using a softmax model with unknown constant temperature θ, which we wish to estimate. Equivalently, as in Example 2, the temperature could be varying with decision number k according to a known function with a single unknown parameter, e.g., $\tau_k = \theta / \log k$. In this case the $\log k$ term can be absorbed into the explanatory variables and we proceed as before.

Figure 6.2 shows how the estimator converges in distribution to the normal distribution (6.28). The explanatory variables $\mathbf{x}^k \sim \mathcal{N}(0, 1)$ were drawn from a Gaussian distribution with mean zero and unit variance, and the response variables $\mathbf{y}^k$ drawn according to probability distribution (6.5) with $\theta_0 = 4$. The estimates in the figure were computed by solving the optimization problem (6.8) using a BFGS quasi-Newton algorithm [19, 37, 42, 108] (Matlab R2013a function `fminunc`). This standard algorithm tends to converge more quickly than the iterative algorithm (6.25). This is likely because the bound matrix $\boldsymbol{B}$ used in iterative algorithm (6.25) does not tightly bound the Hessian $H(\theta)$. Both algorithms use forms of Newton's method, but the approximate Hessian computed by the BFGS algorithm is more accurate than the bound matrix $\boldsymbol{B}$ used by the iterative algorithm, leading to faster convergence. If the problem were high dimensional (i.e., had $n_{obj} \gg 1$), then the relative efficiency of the two algorithms may be different.

Note that the analysis of the preceding sections is still relevant when one uses the BFGS algorithm, as it guarantees concavity of the objective function and therefore that the BFGS algorithm will converge to the correct solution. The analysis is demonstrated by the convergence behavior seen in Figure 6.2, which plots estimates from an ensemble of 100 simulated

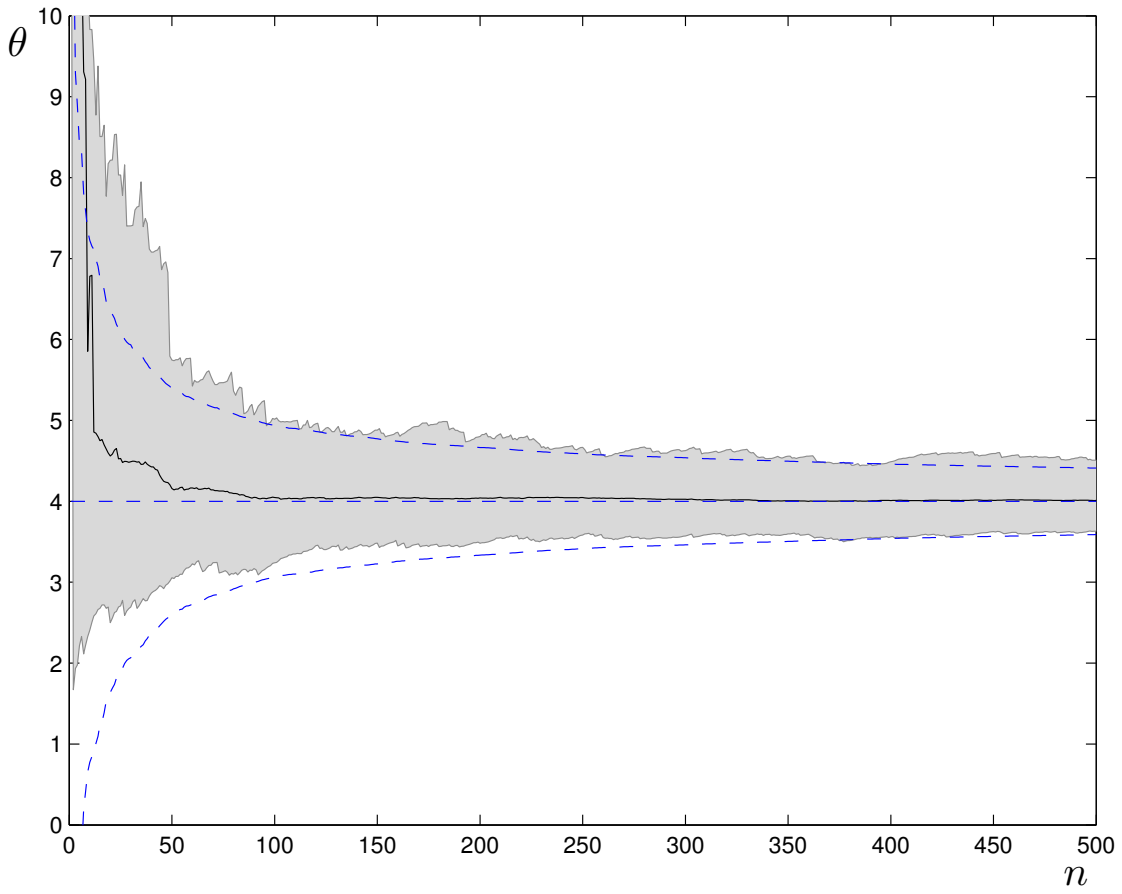


Figure 6.2: Depiction of the estimator's convergence to the asymptotic normal distribution (6.28) as the number of observations n grows. The dashed lines show the true value of the parameter $\theta_0 = 4$ and the accompanying 95% confidence intervals implied by the asymptotic normal distribution (6.15). For each value of n, an ensemble of 100 parameter estimates was formed by repeatedly simulating the data $\mathbf{y}$ while holding the explanatory variables $\mathbf{x}$ fixed, and using the estimator to compute the value of the parameter. The black line shows the mean parameter estimate and the shaded region the empirical 95% confidence interval.

data sets. It can easily be shown that the conditions of Theorem 6.11 are satisfied, so we expect the estimate $\hat{\theta}_{ML}$ to obey the asymptotic normal distribution (6.28). As seen in the figure, not only does the mean parameter estimate (solid black line) converge to the true value θ_0, but the empirical 95% confidence intervals on $\hat{\theta}_{ML}$ converge to the 95% confidence intervals implied by the asymptotic normal distribution (6.28). The convergence of the confidence intervals is evidence that $\hat{\theta}_{ML}$ obeys the asymptotic distribution.

6.6.2 Vector parameter

Second, we consider an instance of the model (6.5) with $m = 100$ options and a vector parameter $\boldsymbol{\theta}$ with $n_{obj} = 3$ elements. The model was simulated 100 times with the explanatory variables $\mathbf{x}^k \sim \mathcal{N}(0, I_3)$ drawn according to independent Gaussian distributions, and the response variables $\mathbf{y}^k$ drawn according to the model (6.5) conditional on $\mathbf{x}^k$ and $\boldsymbol{\theta}_0$.

As in the previous section, it can easily be shown that the conditions of Theorem 6.11 are satisfied, so we expect the estimate $\hat{\boldsymbol{\theta}}_{ML}$ to obey the asymptotic normal distribution (6.28). Figure 6.3 shows that the estimator indeed converges to the true value $\boldsymbol{\theta}_0 = [1, 2, 3]^T$ as the number of samples n increases. The dashed lines in the figure show the true value of each element $(\boldsymbol{\theta}_0)_i$, while the solid lines show the value of each element of the mean parameter estimate $(\hat{\boldsymbol{\theta}}_n)_i$. The shaded regions represent the empirical 95% confidence interval around that mean value, computed from an ensemble of 100 parameter estimates. For clarity, we omit the confidence intervals implied by the asymptotic normal distribution (6.15) from the figure, but the behavior is similar to that shown in Figure 6.2.

6.7 Application to the stochastic UCL decision-making model via linearization

The development up to this point has assumed that the objective function takes the linear form (6.1). However, many relevant objective functions are nonlinear functions of the unknown parameters $\boldsymbol{\theta}$, so the nonlinear function must be converted to a linear form in order to apply the algorithms developed above. A standard way of performing this conversion is by linearization around some nominal point. In this section, we consider the nonlinear objective function from the stochastic UCL algorithm and show that its parameters can be estimated using the bound ML algorithm by linearizing about a nominal point in parameter space.

As a model of human behavior, the stochastic UCL algorithm assumes that the agent's prior distribution of $\mathbf{m}$ (i.e. the agent's initial beliefs about the mean reward values $\mathbf{m}$ and their covariance) is multivariate Gaussian with mean $\boldsymbol{\mu_0}$ and covariance Σ_0:

$$\mathbf{m} \sim \mathcal{N}(\boldsymbol{\mu_0}, \Sigma_0),$$

where $\boldsymbol{\mu_0} \in \mathbb{R}^N$ and $\Sigma_0 \in \mathbb{R}^{N \times N}$ is a positive-definite matrix.

In Chapter 5 we picked a minimal set of three parameters to specify $(\boldsymbol{\mu}_0, \Sigma_0)$ in the case of a spatial multi-armed bandit problem. For the mean we pick a uniform prior $\boldsymbol{\mu}_0 = \mu_0 \mathbf{1}_N$,

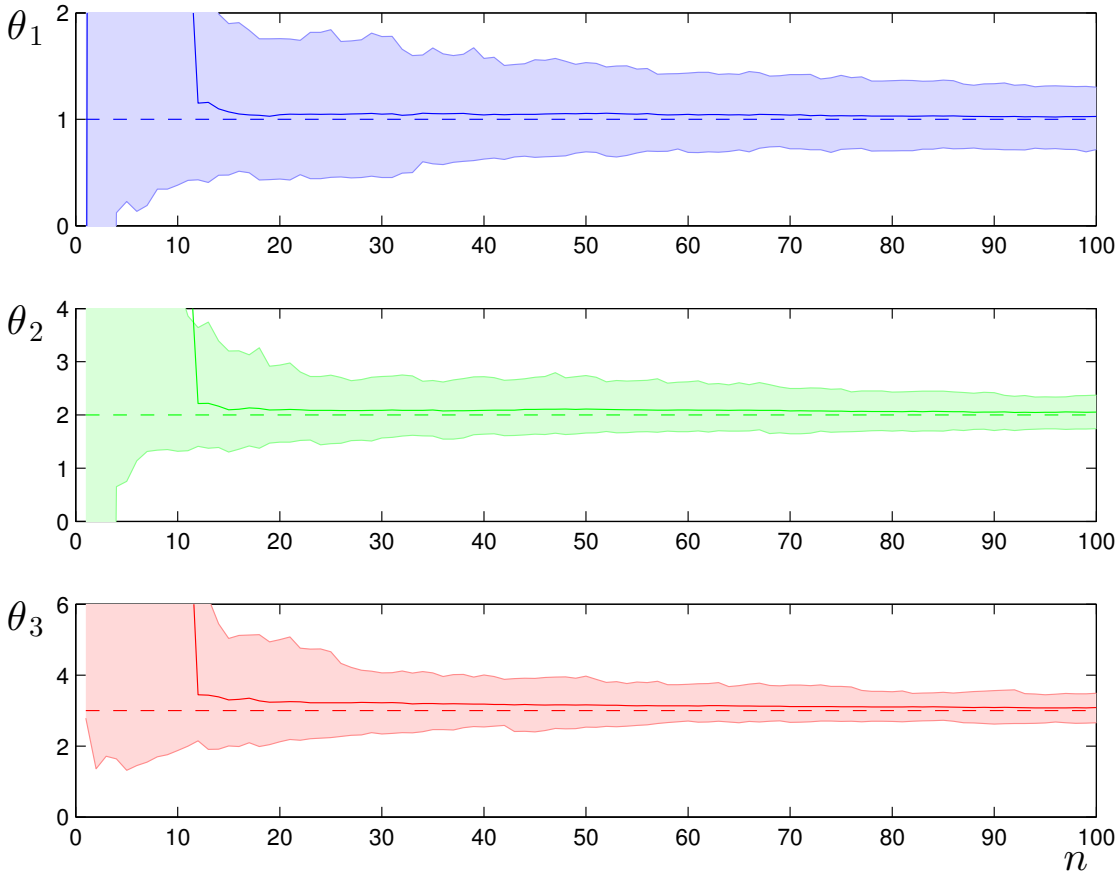


Figure 6.3: Depiction of the estimator's convergence in a vector parameter case as the number of observations n grows. The dashed lines show the true value of each element θ_i of the parameter $\boldsymbol{\theta}_0 = [1, 2, 3]^T$. For each value of n, an ensemble of 100 parameter estimates was formed by repeatedly simulating the data $\mathbf{y}$ while holding the explanatory variables $\mathbf{x}$ fixed, and using the estimator to compute the value of the parameter. The solid lines show the mean parameter estimate and the shaded regions the empirical 95% confidence interval.

where $\mathbf{1}_N \in \mathbb{R}^N$ is the vector with every entry equal to 1 and $\mu_0 \in \mathbb{R}$ is a single parameter that encodes the agent's belief about the mean value of the rewards. For the spatial multi-armed bandit problem, it is reasonable to assume that arms that are spatially close will have similar mean rewards. Therefore, for the covariance Σ_0 we set $\Sigma_0 = \sigma_0^2\Sigma$ where Σ is an exponential prior and each element has the form

$$\Sigma_{ij} = \exp(-|\mathbf{x}_i - \mathbf{x}_j|/\lambda), \tag{6.29}$$

where $\mathbf{x}_i$ is the location of arm i and $\lambda \geq 0$ is the correlation length scale. The parameter $\sigma_0 \geq 0$ can be interpreted as a confidence parameter, with $\sigma_0 = 0$ representing absolute confidence in the beliefs about the mean $\boldsymbol{\mu}_0$, and $\sigma_0 = +\infty$ representing complete lack of confidence.

With this prior, the posterior distribution is also Gaussian, so the Bayesian optimal inference algorithm is linear and can be written down as follows. At each time t, the agent selects option i_t and receives a reward r_t. Let $\mathbf{r}^t$ be the $t \times 1$ vector composed of the r_t. Let n_i^t be the number of times the agent has selected option i up to time t, and let $\mathbf{n}^t$ be the vector composed of the n_i^t. For each time t, define the precision matrix $\Lambda_t = \Sigma_t^{-1}$. Then the belief state at time t is [50, Theorem 10.3]

$$\Lambda_t = \frac{\operatorname{diag}(\mathbf{n}^t)}{\sigma_s^2} + \Lambda_0, \;\; \Sigma_t = \Lambda_t^{-1} \tag{6.30}$$

$$\boldsymbol{\mu}_t = \boldsymbol{\mu}_0 + \Sigma_0 H_t^T \left(H_t \Sigma_0 H_t^T + \sigma_s^2 I_t\right)^{-1} (\mathbf{r}^t - H_t \boldsymbol{\mu}_0), \tag{6.31}$$

where H_t is the $t \times N$ observation matrix with $H_t(t, j) = 1$ if $i_t = j$ and zero otherwise and recalling that I_t is the t-dimensional identity matrix. These equations perform the same update as (4.16), but are written using different notation to facilitate the linearization process.

Based on the belief state $(\boldsymbol{\mu}_t, \Sigma_t)$, the stochastic UCL algorithm chooses arm i_t with probability

$$\mathbb{P}\left(i_t = i | \tilde{Q}, \upsilon_t\right) = \frac{\exp(\tilde{Q}_i^t/\upsilon_t)}{\sum_{j=1}^N \exp(\tilde{Q}_i^t/\upsilon_t)}, \tag{6.32}$$

where $\tilde{Q}_i^t$ is the heuristic function value (4.1) for arm i at time t and υ_t is the temperature corresponding to the cooling schedule at time t. The cooling schedule is assumed to take the form $\upsilon_t = \nu/\log t$, so the probabilities (6.32) become

$$\mathbb{P}\left(i_t = i | \tilde{Q}, \nu\right) = \frac{\exp((\tilde{Q}_i^t \log t)/\nu)}{\sum_{j=1}^N \exp((\tilde{Q}_i^t \log t)/\nu)}. \tag{6.33}$$

The heuristic function value is

$$\tilde{Q}_i^t = \mu_i^t + \sigma_i^t \Phi^{-1}(1 - \alpha_t), \tag{6.34}$$

where $\mu_i^t = (\boldsymbol{\mu}_t)_i$ is the posterior mean reward of arm i at time t and $\sigma_i^t = \sqrt{(\Sigma_t)_{ii}}$ its associated standard deviation. The quantity $\Phi^{-1}(\cdot)$ is the inverse cumulative distribution function of the standard normal distribution and $\alpha_t = 1/\sqrt{2\pi e}t$ is a decreasing function of time.

The model (6.33) is a softmax decision model with unknown parameters $(\mu_0, \sigma_0, \lambda, \nu)$ but not yet in the form (6.1) since the quantity $(\tilde{Q}_i^t \log t)/\nu$ is a nonlinear function of the parameters. However, we can locally approximate this quantity with a linear function by linearizing it about a nominal prior. By estimating the parameter values of the linearized model, we can produce approximate estimates of the parameters of the original nonlinear model (6.33).

6.7.1 Linearization

We assume the value of ν is unknown but fixed and linearize the model (6.33) about a nominal prior. Let $\delta_0^2 = \sigma_s^2/\sigma_0^2$ be the relative precision of a reward measurement compared to the certainty of the prior. Fix a nominal prior with parameters $(\bar{\mu}_0, \bar{\delta}_0^2, \lambda)$, where λ takes its (assumed known) true value, and consider small deviations Δ_μ and Δ_δ in $\bar{\mu}_0$ and $\bar{\delta}_0^2$, respectively:

$$\mu_0 = \bar{\mu}_0 + \Delta_\mu, \;\; \delta_0^2 = \bar{\delta}_0^2 + \Delta_\delta.$$

In the case that the true value of λ is unknown, this method is easily generalized to include deviations in λ, but for simplicity of exposition we consider it fixed at a known value. Recall that the covariance prior is $\Sigma_0 = \sigma_0^2 \Sigma$, where Σ is defined by Equation (6.29), and denote its inverse by $\Lambda = \Sigma^{-1}$.

In terms of δ_0^2, Equation (6.30) becomes

$$\Lambda_t = \frac{1}{\sigma_s^2} \left(\mathrm{diag}(\mathbf{n}^t) + \bar{\delta}_0^2 \Lambda + \Delta_\delta \Lambda \right).$$

Therefore, to first order in Δ_δ, Σ_t is given by

$$\Sigma_t = \sigma_s^2 A_t^{-1} - \sigma_s^2 A_t^{-1} B A_t^{-1} \Delta_\delta + \mathcal{O}\left(\Delta_\delta^2\right), \tag{6.35}$$

where $A_t = \bar{\delta}_0^2 \Lambda + \text{diag}(\mathbf{n}^t)$ and $B = \Lambda = \Sigma^{-1}$. Expanding the square root, we get

$$\sigma_i^t = \sqrt{(\Sigma_t)_{ii}} = \sqrt{c_i^t} - \frac{d_i^t}{2\sqrt{c_i^t}} \Delta_\delta + \mathcal{O}\left(\Delta_\delta^2\right), \tag{6.36}$$

where c_i^t is the i^{th} element on the diagonal of $C_t = \sigma_s^2 A_t^{-1}$ and d_i^t is the i^{th} element on the diagonal of $D_t = \sigma_s^2 A_t^{-1} B A_t^{-1}$. The standard deviation σ_i^t must be non-negative, which implies an upper bound on Δ_δ, which is already assumed to be small. Similarly, δ_0^2 must be non-negative, which implies a lower bound on Δ_δ. The implied bounds on Δ_δ are

$$-\bar{\delta}_0^2 = -\frac{\sigma_s^2}{\bar{\sigma}_0^2} \leq \Delta_\delta \leq \frac{2c_i^t}{d_i^t},$$

which, together with the requirement that Δ_δ be small with respect to $\bar{\delta}_0^2$, gives a bound on the values of Δ_δ for which the linearization is valid.

Similarly, Equation (6.31) for $\boldsymbol{\mu}_t$ becomes

$$\boldsymbol{\mu}_t = E_t + F_t \Delta_\mu + G_t \Delta_\delta + \mathcal{O}\left(\Delta^2\right), \tag{6.37}$$

where Δ^2 denotes second-order terms in the deviation variables Δ_δ and Δ_μ, and E_t, F_t, and G_t are the $N \times 1$ vectors

$$E_t = \bar{\mu}_0 \mathbf{1}_N + \frac{\Sigma H_t^T}{\bar{\delta}_0^2}\left(I_t - H_t A_t^{-1} H_t^T\right)(\bar{\mathbf{m}}^t - H_t \bar{\mu}_0 \mathbf{1}_N) \tag{6.38}$$

$$F_t = \mathbf{1}_N - \frac{\Sigma H_t^T}{\bar{\delta}_0^2}\left(I_t - H_t A_t^{-1} H_t^T\right) H_t \mathbf{1}_N \tag{6.39}$$

$$G_t = -A_t^{-1} B A_t^{-1}(H_t^T \mathbf{m}^t - \mathbf{n}^t \bar{\mu}_0). \tag{6.40}$$

Define e_i^t, f_i^t, and g_i^t as the i^{th} components of E_t, F_t, and G_t, respectively. Then the linearized heuristic is

$$\frac{\tilde{Q}_i^t \log t}{\nu} \approx Q_i^t = \boldsymbol{\theta}^T \mathbf{x}_i^t = \theta_1 x_{i,1}^t + \theta_2 x_{i,2}^t + \theta_3 x_{i,3}^t, \tag{6.41}$$

where the parameters $\boldsymbol{\theta}$ are defined by

$$\theta_1 = \frac{1}{\nu}, \theta_2 = \frac{\Delta_\mu}{\nu}, \theta_3 = \frac{\Delta_\delta}{\nu} \tag{6.42}$$

and the explanatory variables $\mathbf{x}_i^t$ are defined as

$$x_{i,1}^t = \left(e_i^t + \sqrt{c_i^t}\Phi^{-1}(1-\alpha_t)\right)\log t \tag{6.43}$$

$$x_{i,2}^t = f_i^t \log t \tag{6.44}$$

$$x_{i,3}^t = \left(g_i^t - \frac{d_i^t}{2\sqrt{c_i^t}}\Phi^{-1}(1-\alpha_t)\right)\log t. \tag{6.45}$$

The linearized heuristic (6.41) defines a softmax decision-making model with a linear objective function, so we can apply the bound ML algorithm to estimate its parameters $\boldsymbol{\theta}$. By inverting the definition of the parameters (6.42) we can use the estimate of $\boldsymbol{\theta}$ to provide an estimate of the parameters (μ_0, σ_0^2, ν).

6.7.2 Example fits

We tested the estimation procedure described above by simulating runs of the stochastic UCL algorithm using Landscape B from Chapter 5 for various parameter values. See Figures 6.4 and 6.5 for two example fits to simulated data from the stochastic UCL algorithm with parameters $(\mu_0, \sigma_0^2, \lambda, \nu) = (200, 1, 1, 4)$. These parameters result in the algorithm achieving logarithmic regret in numerical simulations (see Figure 5.4). Figure 6.4 shows the fit based on linearization about $(\bar{\boldsymbol{\mu}}, \bar{\sigma}_0^2) = (150, 2)$. Following (6.42), this corresponds to parameters θ_1, θ_2, and θ_3 having true values $\theta_1 = \frac{1}{\nu} = 0.25, \theta_2 = \frac{\mu_0 - \bar{\mu}_0}{\nu} = 12.5$, and $\theta_3 = 1.25 \times 10^{-3}$. Figure 6.5 shows the fit based on linearization about $(\bar{\boldsymbol{\mu}}, \bar{\sigma}_0^2) = (250, 0.5)$. This corresponds to true parameter values $\theta_1 = 0.25, \theta_2 = -12.5$, and $\theta_3 = -2.5 \times 10^{-3}$.

In both cases the estimator converges to the true value of $\boldsymbol{\theta}$ within the horizon $T = 100$ of the decision task and the true value of the parameter is within the 95% confidence interval after 30 observed choices. There are two implications from this result. First, the estimation procedure is at least somewhat robust to the choice of linearization point for this set of algorithm parameters. Second, the estimator is useful for the empirical data reported in Chapter 5. In this case the horizon is $T = 90$ choices. For this amount of data, the simulations show that the estimation procedure can identify the true value of the parameter in a statistically significant way.

The amount of data required to get a reliable estimate can depend on the true value of the algorithm parameters, as shown in Figure 6.6. In this case, the true value of the algorithm parameters are $(\mu_0, \sigma_0^2, \lambda, \nu) = (30, 10^3, 0, 0.5)$ and the linearization point is $(\bar{\boldsymbol{\mu}}, \bar{\sigma}_0^2) = (40, 950)$. This corresponds to true parameter values $\theta_1 = 2, \theta_2 = -20$, and $\theta_3 = -1.05 \times 10^{-6}$. With this prior, the agent is sufficiently uncertain about the rewards to make most of the initial

100 choices at random in order to gain information about the rewards. This choice behavior results in linear regret (see Figure 5.4). Since the initial choices are effectively made at random, they do not provide useful information about the parameter values (except that they represent an uncertain prior). This can be seen from the width of the confidence interval around the mean parameter estimates shown in Figure 6.6. For θ_1 and θ_2 their width is effectively infinite and they are not displayed. For θ_3, the estimate exhibits persistent bias away from the true value, but the width of the associated confidence interval is significantly larger than the bias. Therefore, for such parameter values, one must observe more data to be able to shrink the confidence intervals and provide precise estimates of the parameter values.

6.7.3 Discussion

The linearization procedure described above yields a local linear approximation to the likelihood maximization problem (6.8), and Theorem 6.10 provides conditions under which the local approximation results in an identified model for which the parameter estimation problem is a convex optimization problem. However, the performance of the estimator based on the linearized model is dependent on the linearization point $(\bar{\boldsymbol{\mu}}, \bar{\delta}_0^2)$, which must be chosen by the person using the estimator. This choice may be non-trivial, since it requires picking a linearization point such that the linear approximation is valid at the true value of the parameters. In the worst case, one might not have any intuition about which linearization point to choose, making the above procedure no better than any other local optimization technique for which one must choose a starting point.

Fortunately, several aspects of the problem come to the rescue. The first is generic to any heuristic function, and relies on the fact that the likelihood function forms a unique objective for judging the "goodness" of the estimated parameter. If one is unsure of which linearization point to choose, one can simply fit the model assuming two different linearization points and compare the resulting estimates $\hat{\boldsymbol{\theta}}$. If the two linearization points result in identical estimates there is no conflict, while if the estimates differ the one with the higher likelihood value is better.

Second, one may have intuition about the location of an acceptable linearization point due to the structure of the model. In Chapter 5, we showed that behavior of the stochastic UCL model falls broadly into three classes as a function of the parameters $(\mu_0, \sigma_0^2, \lambda, \nu)$: linear, power law, or logarithmic regret. By categorizing a given data set into one of the three classes, one can narrow the search for a linearization point to the associated region of parameter space. Third, the stochastic UCL model is relatively insensitive to the choice of

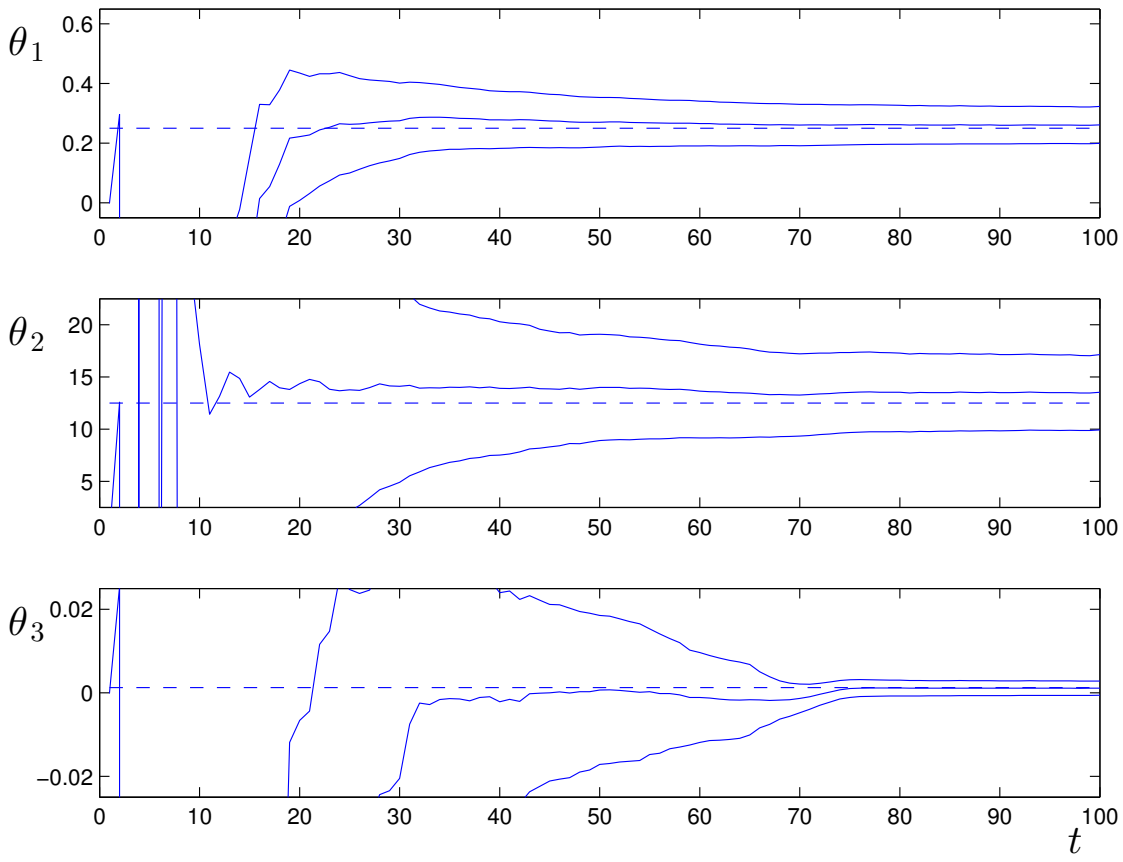


Figure 6.4: Estimates of the vector of parameters $\boldsymbol{\theta}$ fitted to simulated data from the UCL algorithm. The linearization point was taken to be $\bar{\mu}_0 = 150, \bar{\sigma}_0^2 = 2$. The true algorithm parameters were $\mu_0 = 200, \sigma_0^2 = 1, \lambda = 1$, and $\nu = 4$. The dashed lines show the true value of each element θ_i of the parameter vector for the linearized objective function. The estimator converges to the true parameter values as the number of observations t grows. For each value of t, an ensemble of 100 parameter estimates was formed by repeatedly simulating the data $\{(\mathbf{x}_t, \mathbf{y}_t)\}_{t=1}^T$ while holding the parameters $\boldsymbol{\theta}$ fixed, and using the estimator to compute the value of the parameter. The solid lines show the mean parameter estimates and the 95% confidence intervals implied by the asymptotic normal distribution (6.28). The width of the confidence intervals roughly scales with the magnitude of the parameter values, similar to the behavior seen in Figure 6.3.

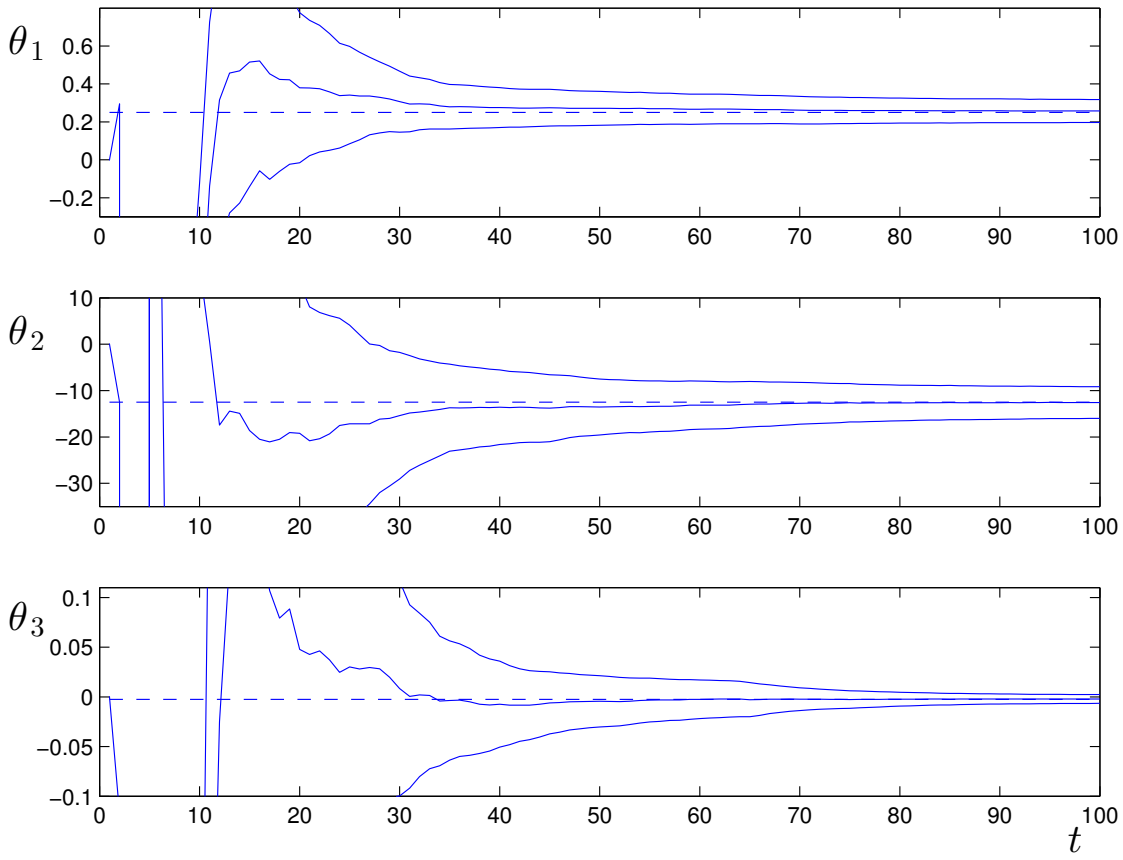


Figure 6.5: Estimates of the vector of parameters $\boldsymbol{\theta}$ fitted to simulated data from the UCL algorithm. The linearization point was taken to be $\bar{\mu}_0 = 250, \bar{\sigma}_0^2 = 0.5$. The true algorithm parameters were $\mu_0 = 200, \sigma_0^2 = 1, \lambda = 1$, and $\nu = 4$. The dashed lines show the true value of each element θ_i of the parameter vector for the linearized objective function. The estimator converges as the number of observations t grows. For each value of t, an ensemble of 100 parameter estimates was formed by repeatedly simulating the data $\{(\mathbf{x}_t, \mathbf{y}_t)\}_{t=1}^{T}$ while holding the parameters $\boldsymbol{\theta}$ fixed, and using the estimator to compute the value of the parameter. The solid lines show the mean parameter estimates and the 95% confidence intervals implied by the asymptotic normal distribution (6.28). The width of the confidence intervals roughly scales with the magnitude of the parameter values, similar to the behavior seen in Figure 6.3.

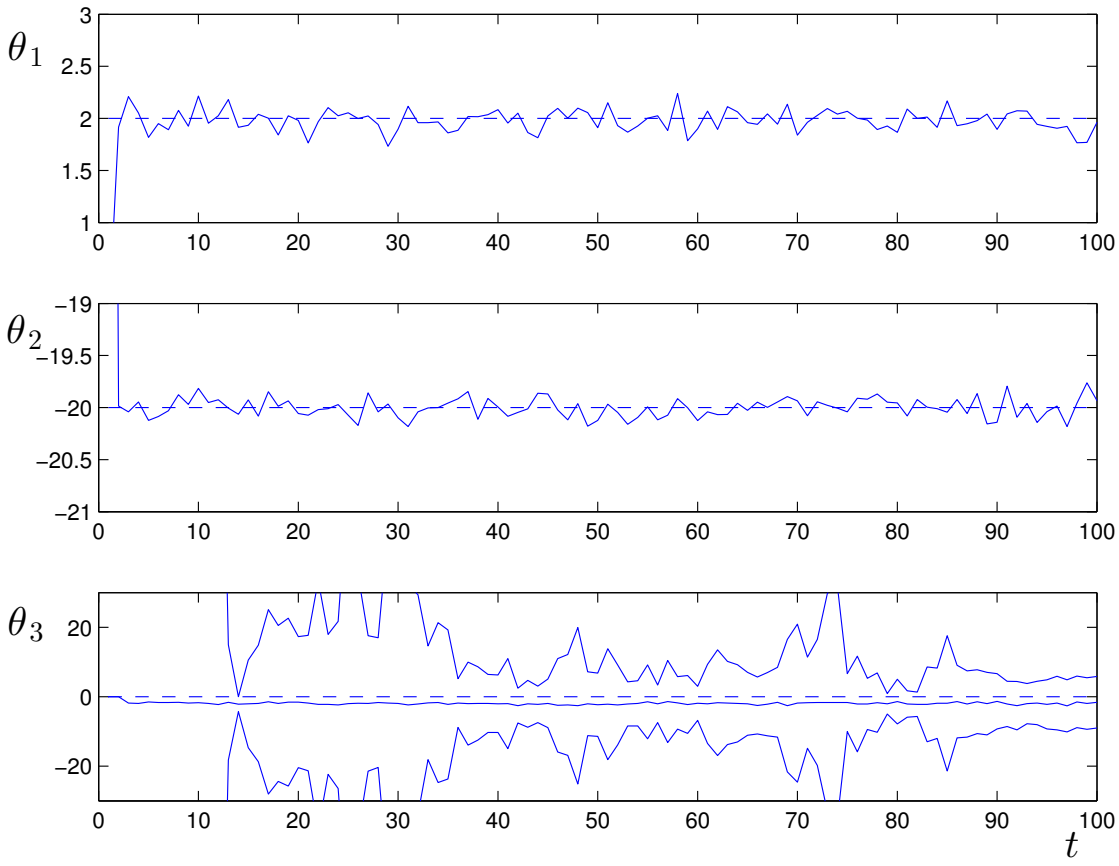


Figure 6.6: Estimates of the vector of parameters $\boldsymbol{\theta}$ fitted to simulated data from the UCL algorithm with a weakly-informative prior. This prior makes the algorithm's choice behavior more random, which makes the estimation problem more difficult. The linearization point was taken to be $\bar{\mu}_0 = 150, \bar{\sigma}_0^2 = 2$. The true algorithm parameters were $\mu_0 = 200, \sigma_0^2 = 1, \lambda = 1$, and $\nu = 4$. The dashed lines show the true value of each element θ_i of the parameter vector for the linearized objective function. The estimator converges as the number of observations t grows. For each value of t, an ensemble of 100 parameter estimates was formed by repeatedly simulating the data $\{(\mathbf{x}_t, \mathbf{y}_t)\}_{t=1}^{T}$ while holding the parameters $\boldsymbol{\theta}$ fixed, and using the estimator to compute the value of the parameter. The solid lines show the mean parameter estimates and, in the panel corresponding to θ_3, the 95% confidence interval implied by the asymptotic normal distribution (6.28). For parameters θ_1 and θ_2, the confidence intervals have essentially infinite width and are omitted for legibility.

linearization point within the region of parameter space associated with a given behavioral class, as we saw for the case of logarithmic regret in Figures 6.4 and 6.5.

6.8 Conclusions

Motivated by the parameter estimation problem for the stochastic UCL algorithm, we developed a general algorithm for likelihood-based parameter estimation in softmax decision-making models with linear objective functions. Such models occur frequently in the neuroscience and machine learning literatures. We derived conditions under which the general algorithm converges on the correct parameter value and characterized the rate of convergence, which can be used to formulate confidence intervals for the parameter estimates. In developing the algorithm, we constructed several new matrix operations and established some of their important properties.

We then showed that the stochastic UCL algorithm could be transformed into a softmax decision-making model with a linear objective function by linearizing the objective function about a known point in parameter space. By performing parameter estimation on the linearized model using simulated data, we showed that we could estimate the true value of the stochastic UCL algorithm parameters. The amount of data required to perform useful estimation depended on the region of parameter space, with parameters representing strong priors being easier to estimate. This is intuitive, as a strongly-held belief will influence behavior in a way that is more readily observable than a weakly-held belief.

The stochastic UCL algorithm, together with the estimation procedure developed in this section, provides a plant-observer pair for human choice behavior in multi-armed bandit problems. This pair allows the system-theoretic design of human-machine teams. Fortunately for the development of such teams, humans with high-quality, strong priors are the ones who can provide the most value in the form of intuition from experience. A key issue is to identify humans with high-quality priors.

The methods devised in Chapter 5 to classify human performance in multi-armed bandit tasks using observed regret $\mathcal{R}(t)$ provide a way to perform this identification. By estimating observed regret (a quantity similar to the psychologically-relevant notions of regret discussed in Section 2.1), a system could identify human operators with good performance, i.e., high-quality priors, in real time. The system could then estimate these operators' priors and employ them to inform its decision making using the stochastic UCL algorithm. Such a system would be an implementation of the framework for human-machine search that is a core goal of this thesis.

Chapter 7

Satisficing in Gaussian multi-armed bandits[1]

Multa petentibus
Desunt multa.
Bene est, cui Deus obtulit
Parca, quod satis est manu.[2]
(Horace)

In this chapter we turn to some more theoretical questions concerning decision making with a human-inspired objective function. We stay within the context of the multi-armed bandit problem but consider the so-called "satisficing" objective, in which the decision-making agent is satisfied if its reward is above some known threshold value. This objective induces different behavior from the standard maximizing objective (2.1). Notably, it encourages the decision maker to balance risk, in the form of variance of the rewards from a given decision, and return, in the form of the expected value of the rewards. Such a risk-return tradeoff can be intuitively desirable in decision-making problems, for example in ecology or finance.

The work in this chapter is complementary to the work presented in the previous chapters in that it provides an alternative objective for the decision-making process. We show that the multi-armed bandit problem with the satisficing objective is equivalent to a standard multi-armed bandit problem with a maximizing objective. In the case of Gaussian rewards, we show that this equivalent standard multi-armed bandit problem can be optimally solved by the UCL algorithm, so the framework from the previous chapters can be brought to bear.

[1]This chapter is adapted from [95], with most text taken verbatim.

[2]Those who seek much are left wanting much. Happy is he to whom God has given, with sparing hand, as much as is enough.

The risk-return tradeoff is one example of a more general class of objectives that could be applied to multi-armed bandit problems. In general, one can consider an objective defined by a *utility function* that accounts for any number of tradeoffs between different desirable outcomes. As such, the work in this chapter is a first step towards a theory of multi-armed bandit problems with utility function objectives.

7.1 Introduction

Engineering solutions to decision-making problems are often designed to maximize an objective function. However, in many contexts maximization of an objective function as in (2.1) is an unreasonable goal, either because the objective itself is poorly defined or because solving the resulting optimization problem is intractable or costly. In these contexts, it is valuable to consider alternative decision-making frameworks.

Herbert Simon considered alternative models of rational decision making with the goal of making them "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" [110]. A major feature of the models he considered is what he called "satisficing". In [110], Simon discussed in very broad terms a variety of simplifications to the classical economic concept of rationality, most importantly the idea that payoffs should be simple, defined by doing well relative to some threshold value. In [111], he introduced the word "satisficing" to refer to this thresholding concept and considered an ecological example of food foraging behavior in detail using mathematical terms. He also briefly discussed how satisficing relates to problems in inventory control and more complicated decision processes like playing chess.

Since Simon's pioneering work, satisficing has been studied in many fields such as psychology [107], economics [15], management science [78, 134], and ecology [127, 26]. In engineering, satisficing is of interest for the same reasons that motivated its introduction in the social science literature, specifically that it can simplify decision-making problems. Furthermore, many engineering problems are naturally posed using a satisficing objective, for example design problems that have to meet given specifications. A design that meets all the required specifications is acceptable, and the designers may be indifferent among all such designs. In this context, optimization may be poorly defined, for example if there are several competing performance measures that trade off in complicated ways. Satisficing can be a simpler decision paradigm than maximizing, which requires additional information about preferences among possible tradeoffs.

Satisficing has been studied in the engineering literature in several contexts. In [81], Nakayama studied design optimization using a satisficing objective and found that it is effective in many practical fields. In [43], the authors studied control theory using a satisficing objective function, and in [135], the authors used satisficing to study optimal software design.

Satisficing can be implemented in a variety of ways. In this chapter, we consider satisficing in the context of the multi-armed bandit problem. The standard multi-armed bandit problem uses a maximizing objective, for which there is a known performance bound. We propose a satisficing objective for the multi-armed bandit problem based on the number of times the decision maker receives a reward that is above a threshold value and show that the multi-armed bandit problem with this objective is equivalent to a related standard multi-armed bandit problem. We use the equivalent problem to derive a performance bound for the new satisficing problem.

For Gaussian bandit problems, i.e., where the reward distributions are Gaussian with unknown mean and known variance, we show that solving the problem with the satisficing objective is equivalent to solving a standard Gaussian multi-armed bandit problem. We then apply the UCL algorithm developed in Chapter 4 to the standard problem, and show that this algorithm achieves optimal performance in terms of the original satisficing objective.

7.2 The multi-armed bandit problem with satisficing objective

The standard multi-armed bandit problem is defined with a maximizing objective. We now propose a new satisficing objective for the multi-armed bandit problem and find bounds on optimal performance in terms of the new objective.

Consider an N-armed bandit problem. As before, the reward associated with each arm i is drawn from a stationary probability distribution p_i, whose mean m_i is unknown to the decision maker. At time $t \in \{1, \dots, T\}$, the decision maker selects arm i_t and receives a stochastic reward $r_t \in \mathbb{R}$.

The decision maker has a certain satisfaction level $M \in \mathbb{R}$, and is satisfied at time t only if the reward r_t is at least M. Let s_t be the random variable denoting the decision maker's satisfaction at time t:

$$s_t = \begin{cases} 0, & r_t < M \\ 1, & r_t \geq M. \end{cases}$$

Then s_t is a Bernoulli random variable with success probability π_{i_t}, where

$$\pi_i = \mathbb{P}\left(s_t = 1 | i_t = i\right) = \mathbb{P}\left(r_t \geq M | i_t = i\right) \tag{7.1}$$

is the probability of satisfaction upon picking arm i. We propose a satisficing objective in terms the number of times the satisfaction level is met.

Definition 7.1 (Satisficing objective)**.** *The* satisficing objective *is to maximize the function*

$$\mathbb{E}\left[\sum_{t=1}^{T} s_t\right] = \sum_{t=1}^{T} \pi_{i_t}. \tag{7.2}$$

The satisficing objective differs from the maximization objective in several important ways. First, it exhibits thresholding, that is, it is indifferent among rewards r_t above the threshold value M. Second, it exhibits risk aversion, that is, it prefers smaller, consistent rewards (that will often be above the threshold) to larger, more variable ones (that may often be below it). Risk aversion is a characteristic often studied in economics and psychology [91], and is often incorporated in models of human decision-making.

The satisficing objective consists of maximizing the number of times the agent is satisfied, which is equivalent to minimizing the number of times they are not satisfied. Let $\pi_{i^*} = \max_i \pi_i$ and define $\bar{\Delta}_i = \pi_{i^*} - \pi_i$ as the expected regret of selecting an arm i. We can rewrite (7.2) in terms of minimizing cumulative regret J_S:

$$J_S = T\pi_{i^*} - \mathbb{E}\left[\sum_{t=1}^{T} s_t\right] = \mathbb{E}\left[\sum_{t=1}^{T} \bar{\Delta}_{i_t}\right] = \sum_{i=1}^{N} \bar{\Delta}_i \mathbb{E}\left[n_i^T\right], \tag{7.3}$$

where n_i^T is the number of times arm i has been chosen up to time T. This is a standard multi-armed bandit problem with Bernoulli rewards. Therefore the Lai-Robbins bound (2.5) holds, yielding a logarithmic lower bound on $\mathbb{E}\left[n_i^T\right]$ and cumulative regret J_S, as formalized by the following corollary:

Corollary 7.2 (Satisficing regret bound)**.** *Any policy solving the multi-armed bandit problem with the satisficing objective* (7.3) *obeys*

$$\mathbb{E}\left[n_i^T\right] \geq \left(\frac{1}{D(\pi_i||\pi_{i^*})} + o(1)\right) \log T, \tag{7.4}$$

for suboptimal arms $i \neq i^*$ where $D(\pi_i||\pi_{i^*}) = \pi_{i^*} \log\left(\frac{\pi_i}{\pi_{i^*}}\right) + (1 - \pi_{i^*}) \log\left(\frac{1-\pi_i}{1-\pi_{i^*}}\right)$ is the Kullback-Leibler divergence between the two Bernoulli distributions with success probabilities π_i and π_{i^*}.

Proof. Apply the Lai-Robbins bound (2.5) to the standard multi-armed bandit problem with Bernoulli rewards. □

The implication of writing the satisficing objective as the minimizing of cumulative regret is that if one can use the rewards r_t to estimate the satisfaction probability π_{i_t}, one can use algorithms designed to solve the multi-armed bandit problem with a maximizing objective to solve the satisficing problem. In the next sections we study the Gaussian multi-armed bandit problem with a satisficing objective and show how to link rewards and probabilities in this case.

7.3 Satisficing with Gaussian rewards

In this section we study a Gaussian multi-armed bandit problem with the satisficing objective (7.3). By Gaussian multi-armed bandit problem, we mean that the reward r_t due to selecting arm i_t is $r_t \sim \mathcal{N}(m_{i_t}, \sigma^2_{s,i_t})$, where σ^2_{s,i_t} is the known variance of arm i_t.

Define the quantity

$$x_i = \frac{m_i - M}{\sigma_{s,i}} \tag{7.5}$$

for each arm i. The following lemma states that the Gaussian multi-armed bandit problem with a satisficing objective is equivalent to a standard Gaussian multi-armed bandit problem with transformed reward distributions.

Lemma 7.3 (Equivalence for Gaussian rewards)**.** *The Gaussian multi-armed bandit problem with satisficing objective is equivalent to a standard Gaussian multi-armed bandit problem with rewards $\tilde{r}_t \sim \mathcal{N}(x_{i_t}, 1)$ in the sense that the ordering of the arms in terms of x_i is identical to the ordering in terms of π_i. In particular, the arm with maximal x_i is the arm with maximal π_i*

Proof. With Gaussian rewards, the probability (7.1) of satisfaction from choosing arm i is

$$\begin{aligned}\pi_i &= \mathbb{P}\left(m_i + \sigma_{s,i} z \geq M\right) \\ &= \Phi\left(\frac{m_i - M}{\sigma_{s,i}}\right) = \Phi(x_i),\end{aligned}$$

where $z \sim \mathcal{N}(0,1)$ is a standard normal random variable and $\Phi(z)$ is its cumulative distribution function. Let $i^* = \arg\max_i \pi_i$. The key insight is that $\Phi(\cdot)$ is a monotonically increasing function, which implies that the ordering of arms in terms of π_i is identical to the ordering in terms of x_i. In particular, arm i^* is the arm with maximal x_i. Therefore, the goal of an agent playing the satisficing bandit problem is to find the arm i^* that maximizes x_i.

This is again a Gaussian bandit problem: consider the transformed reward

$$\tilde{r}_t = \frac{r_t - M}{\sigma_{s,i}},$$

which is a Gaussian random variable $\tilde{r}_t \sim \mathcal{N}(x_{i_t}, 1)$. The quantity x_i plays the role of the mean reward m_i from the original maximizing problem and the transformed rewards have uniform variance $\tilde{\sigma}_s^2 = 1$. Solving this problem with a maximizing objective is equivalent to solving the original problem with the satisificing objective. □

Remark 7.4 (Location-scale families)**.** *The above analysis is easily generalized to reward distributions belonging to location-scale families. A location-scale family is a set of probability distributions closed under affine transformations, i.e., if the random variable X is in the family, so is the variable $Y = a + bX$, where $a, b \in \mathbb{R}$. Any random variable X in such a family with mean μ and standard deviation σ can be written as $X = \mu + \sigma Z$, where Z is a zero-mean, unit-variance member of the family. Examples include the Uniform or Student's t-distribution.*

7.4 Logarithmic satisficing regret

In Section 7.3, we showed that solving the Gaussian multi-armed bandit problem with a satisficing objective is equivalent to a transformed standard Gaussian multi-armed bandit problem with maximizing objective. Therefore, we can apply the UCL algorithm to the satisficing problem. A prior belief $\mathbf{m} \sim \mathcal{N}(\boldsymbol{\mu}_0, \Sigma_0)$ is transformed into prior beliefs on $\mathbf{x}$ by

$$\mathbf{x} \sim \mathcal{N}(\tilde{\boldsymbol{\mu}}_0, \tilde{\Sigma}_0),$$

where $(\tilde{\boldsymbol{\mu}}_0)_i = ((\boldsymbol{\mu}_0)_i - M)/\sigma_{s,i}$, $(\tilde{\Sigma}_0)_{ij} = (\Sigma_0)_{ij}/(\sigma_{s,i}\sigma_{s,j})$, and M is the satisfaction level. Define $x_{i^*} = \max_i x_i$ and $\tilde{\Delta}_i = x_{i^*} - x_i$.

We refer to the UCL algorithm using the transformed reward $\tilde{r}_t$ and prior as the *satisficing UCL algorithm*. We define $\{R_t^{\mathrm{SaUCL}}\}_{t\in\{1,\dots,T\}}$ as the sequence of expected regret for the satisficing UCL algorithm. The satisficing UCL algorithm achieves logarithmic regret, as formalized in the following theorem.

Theorem 7.5 (Regret of the satisficing UCL algorithm)**.** *The following statements hold for the Gaussian multi-armed bandit problem with a satisficing objective and the satisficing UCL algorithm with uncorrelated uninformative prior and $K = \sqrt{2\pi e}$:*

1. the expected number of times a suboptimal arm i is chosen until time T satisfies

$$\mathbb{E}\left[n_i^T\right] \leq \left(\frac{8\beta^2}{\tilde{\Delta}_i^2} + \frac{2}{\sqrt{2\pi e}}\right)\log T + \frac{4\beta^2}{\tilde{\Delta}_i^2}(1 - \log 2 - \log\log T) + 1 + \frac{2}{\sqrt{2\pi e}}\,;$$

2. the cumulative expected regret until time T satisfies

$$\sum_{t=1}^{T} R_t^{\mathrm{SaUCL}} \leq \sum_{i=1}^{N} \tilde{\Delta}_i \left(\left(\frac{8\beta^2}{\tilde{\Delta}_i^2} + \frac{2}{\sqrt{2\pi e}}\right)\log T + \frac{4\beta^2}{\tilde{\Delta}_i^2}(1 - \log 2 - \log\log T) + 1 + \frac{2}{\sqrt{2\pi e}} \right). \tag{7.6}$$

Proof. Apply Theorem 4.2 to the Gaussian multi-armed bandit problem with mean rewards $\mathbf{x}$ and reward distributions $\tilde{r}_t \sim \mathcal{N}(x_{i_t}, 1)$ defined in Lemma 7.3. □

The regret in the satisficing problem is upper bounded by a logarithmic function of T. Therefore, the satisficing UCL algorithm achieves optimal performance in the satisficing problem up to a constant factor.

7.5 Numerical example

In this section, we present the results of two numerical simulations of the satisficing UCL algorithm solving a multi-armed bandit problem with Gaussian rewards and the satisficing objective. The first simulation demonstrates the performance guarantees and allows us to compare the optimal regret bound (7.4) and the bound (7.6) obeyed by the satisficing UCL algorithm. The second simulation demonstrates the risk-averse nature of the satisficing objective.

For the simulations presented in Figure 7.1, we set $N = 4$. The satisfaction level M was set equal to 2, the mean rewards $\mathbf{m}$ were equal to [1 2 3 4] and the standard deviations equal to [1 1 1 3], so $\mathbf{x} = [-1\ 0\ 1\ \frac{2}{3}]$ and $i^* = 3$ was the optimal arm. The algorithm used an uninformative prior. These values were chosen such that the arm with maximal mean reward was not the optimal arm, so satisficing induces different behavior than maximizing.

Figure 7.1 plots the regret incurred by the satisficing UCL algorithm along with the two regret bounds (7.4) and (7.6). The mean regret obeys the performance bound (7.6) from Theorem 7.5. Mean regret from the 1,000 simulations is actually better than the asymptotic

lower bound (7.4). This apparent violation of the bound is due to the fact that even at horizon $T = 10^3$, we are not yet in the asymptotic regime where the bound applies.

For the simulations presented in Figure 7.2, we set $N = 2$. The mean rewards $\mathbf{m}$ were equal to [12.2 12.1] and the standard deviations equal to [10 1], so $\mathbf{x} = [0.02\ 0.1]$. This meant $i = 1$ was the optimal arm for the maximizing objective while $i = 2$ was the optimal arm for the satisficing objective. The algorithm used an uninformative prior. The problem was simulated 100 times with each objective.

Figure 7.2 demonstrates the risk aversion inherent in the satisficing objective by comparing the results of the same problem solved with the satisficing and the maximizing objectives. The satisfaction level M was set equal to 12. We considered cumulative surplus (rewards in excess of the satisfaction level) for both objectives. Negative values of the surplus represent deficits, which are to be avoided. Results from the maximizing objective are presented in black. The solid line shows mean cumulative surplus and the shaded region the 95% confidence interval around that mean. Results from the satisficing objective are presented in blue. The solid line shows the mean cumulative surplus, and the dashed lines show the 95% confidence interval. The lower limit of the confidence intervals measures worst-case performance. The measure for the satisficing objective is consistently above the one for the maximizing objective, so satisficing results in better worst-case performance in this example.

7.6 Conclusion

Satisficing, the concept of doing well relative to a reference value, is a useful alternative to maximizing that can be applied to a variety of decision-making scenarios. Considering satisficing objectives instead of maximizing ones can simplify decision-making problems and can result in policies that are more robust in the sense that they are risk-averse.

In this chapter, we considered the multi-armed bandit problem using a satisficing objective. The multi-armed bandit problem is a canonical decision-making problem that is widely studied in machine learning and adaptive control using a maximization objective. We proposed a satisficing objective for stochastic multi-armed bandit problems and showed that solving the problem with a satisficing objective is equivalent to solving the problem with a modified maximization objective. Using the modified maximization objective, we derived bounds on optimal performance in these problems. We studied the case of Gaussian rewards and showed how to use the UCL algorithm to solve this problem, achieving optimal performance.

This work opens the door to many future extensions. The satisficing objective with Gaussian rewards bears a strong resemblance to the CreditMetrics two-state credit risk

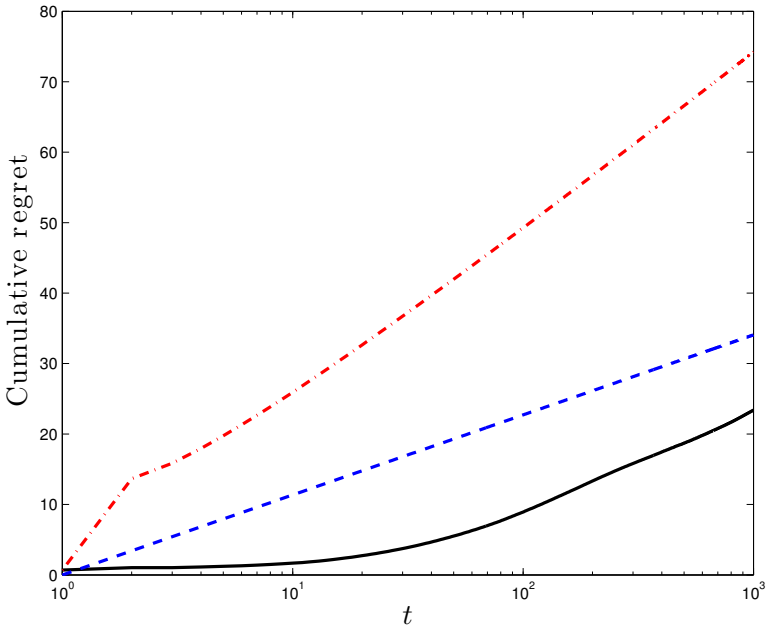


Figure 7.1: Regret incurred by the UCL algorithm while solving a satisficing Gaussian multi-armed bandit problem, along with two theoretical bounds, plotted against time t on a logarithmic scale. The solid black line shows mean cumulative expected regret from 1,000 simulations. The dashed blue line shows the asymptotic bound on regret (7.4), which appears as a straight line due to the scaling of the axes. The dash-dotted red line shows the regret bound (7.6), which provides guarantees on the algorithm's performance.

model used in quantitative finance [44]. This could allow the credit investment portfolio problem studied in finance to be posed as a multi-armed bandit problem with satisficing objective.

The risk aversion effects of multi-armed bandits with satisficing objectives will result in more robust policies for solving the multi-armed bandit problem in cases with reward variance σ_s^2 that are heterogeneous across arms. Precisely quantifying the gains in robustness will be the subject of future work. Risk aversion and robustness are important for engineering applications (where standard bandit algorithms are known to have poor risk-aversion characteristics [8]), and also in the field of optimal foraging theory [26]. The multi-armed bandit framework has been used to study foraging [117] using a maximizing objective, but a satisficing objective is more ecologically plausible.

We developed a policy for the satisficing problem with Gaussian rewards, but development of optimal policies for the satisficing problem with other reward distributions remains an open problem. For all satisficing problems, picking the appropriate satisfaction level is a non-trivial problem in its own right. Both of these problems will be the subject of future work.

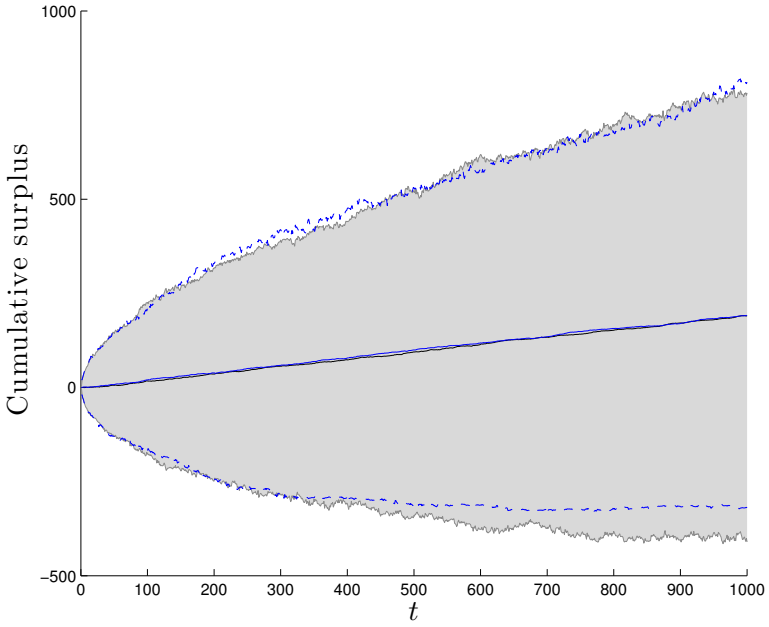


Figure 7.2: Cumulative surplus earned by the UCL algorithm while solving a Gaussian multi-armed bandit problem, once with a satisficing (blue curves) and again with a maximizing objective (black curve and shaded region). Both objectives achieve similar mean performance (solid curves) but using the satisficing objective results in better worst-case performance. The shaded region shows the 95% confidence interval around the mean cumulative surplus for the satisficing objective and the blue dashed lines show the same interval for the maximizing objective. The lower limit of the confidence intervals measures worst-case performance. The lower limit for the satisficing objective is consistently above the one for the maximizing objective, so satisficing results in better worst-case performance.

Chapter 8

Conclusions

"A human must turn information into intelligence or knowledge. We've tended to forget that no computer will ever ask a new question."

(Grace Hopper)

This thesis has been motivated by applications in adaptive sensing and robotics, such as spatial search, that push the boundary of what is achievable with current automation technology. For such applications, human supervision of automated systems is essential to guarantee that the automated system performs as desired. The goal of this thesis is to facilitate the principled integration of humans and automation in mixed human-machine decision-making teams. We make progress towards this goal by developing a framework for human-machine *search*, which is the word we use to describe rational decision making under uncertainty.

We have considered the problem of decision making under uncertainty in a variety of tasks. To such tasks, humans bring the ability to quickly discern patterns in data and intuition based on prior experience, while machines bring abundant computational power and memory, as well as the ability to precisely follow repetitive rules. The desire to integrate humans and machines has led us to build a series of models of human decision-making behavior in search tasks. The models are based on statistical representations of the task, and rely on Bayesian estimators to allow us to quantitatively capture human intuition. By formally modeling the tasks as well-defined optimization problems, we make the models quantitatively rigorous. This allows us to prove conditions under which the models perform optimally.

8.1 Summary

In this section we review the work presented in this thesis in detail and highlight the major contributions made. The overarching contribution of this thesis is to develop a principled model of human decision-making under uncertainty, whose parameters quantify the human's representation of the decision-making task, and an estimator for the model parameters. We rigorously analyze the performance of the model and prove conditions under which the model achieves optimal performance. We also consider empirical data from a human-subject spatial search task and show that the model captures the major features in the data. For future applications, the model provides a common language for humans and machines to share relevant information about the task. By estimating the parameters, a machine can access this representation and potentially improve its performance. In control systems terminology, the model and associated estimator form a plant–observer pair for human decision making that can be used for system design.

As explained in Chapter 2, the key step that facilitated the framework was the choice to model the spatial search task as a *spatial multi-armed bandit problem*, which is a multi-armed bandit problem where the arms are embedded in an underlying space. The spatial multi-armed problem is related to recent work [55, 23, 116] on so-called continuous-armed bandit problems, where each point of a continuous metric space is considered as an arm of a multi-armed bandit. All multi-armed bandit problems are prototypical models of decision making under uncertainty, where an important tradeoff is between *exploration*, which is making decisions to learn more about the decision space, and *exploitation*, which is making decisions to maximize the quality of the immediate decision.

Key to the development of our framework is the fact that the multi-armed bandit problem has been well studied in both the machine learning and neuroscience literatures, so there is a wealth of results on which to draw. From the machine learning literature we drew on results that bounded the achievable performance in multi-armed bandit problems and developed algorithms that achieve optimal performance in some cases. From the neuroscience literature, we drew on recent work that studied the features of human decision making in multi-armed bandit problems.

In Chapter 3, we made a first step towards connecting these two literatures by taking the ambiguity bonus heuristic, a model of human decision making that was developed in the neuroscience literature for a two-armed bandit, and extending it to the general case of $N \geq 2$ arms. We studied the properties of the resulting model through simulation and showed that the optimal values of the model parameters implied an interesting tradeoff between exploration, which could be generated by either a directed or a random mechanism,

and exploitation. We studied the parameter tuning problem analytically in some simple cases and gained some insight into the ways the different parameters interact.

In Chapter 4, we developed an alternative heuristic-based model, which we term the Upper Credible Limit (UCL) algorithm. We developed UCL by adapting algorithms from the machine learning literature, notably Bayes-UCB [49], to the spatial multi-armed bandit problem. UCL captures the major features of human decision-making behavior in multi-armed bandit problems using a heuristic that is similar to the ambiguity bonus heuristic developed in Chapter 3 but its performance is simpler to analyze in terms of *regret*. By extending analysis from the machine learning literature, we proved that UCL achieves logarithmic regret, which is optimal performance in the multi-armed bandit task. The components of the UCL heuristic can be interpreted in terms of the components of the ambiguity bonus heuristic, so UCL can be interpreted as giving optimal tunings to the heuristic parameters. The UCL algorithm is a major contribution because it formalizes the link between the heuristics studied in the neuroscience literature and the algorithms studied in the machine learning literature.

Key to the value of the UCL algorithm as a model of human decision making and as an algorithm for solving the spatial multi-armed bandit task is the introduction of correlated priors in the Bayesian estimator used to maintain its belief state. Such priors model the (human or algorithmic) agent's beliefs about the smoothness of the reward surface, and allow information learned about the reward values in one location to be propagated to neighboring areas. This propagation of information causes the estimator to converge more quickly than in the case of an uncorrelated prior, where the rewards are assumed to be independent and no propagation takes place. When the reward surface is smooth, propagating information is valuable and allows the UCL algorithm to achieve sub-logarithmic, i.e., better than logarithmic, regret. The idea of assuming a degree of smoothness in the reward surface exists in the literature on continuous-armed bandit problems, but to the best of our knowledge our work is the first time correlation has been introduced into the literature on multi-armed bandits with a finite number of arms.

In Chapter 5 we studied data from an experiment in which human subjects performed a spatial multi-armed bandit task. We showed that the performance of many human subjects was very good. We further showed that human performance could be classified into three categories, termed *phenotypes*, that align with various known bounds on performance from the machine learning literature. We also showed that the UCL algorithm could emulate behavior falling into these various categories by tuning a small number of algorithm parameters, in particular the algorithm's prior. On the basis of this evidence, we interpreted the high performance of the subjects in the best-performing phenotype as due to these subjects having a good prior for the reward surface. We noted that such good priors would provide

useful information for a human-machine system if a machine could estimate the corresponding model parameter values.

In Chapter 6, we focused on the parameter estimation problem for the UCL algorithm as parametrized in Chapter 5. The UCL algorithm trivially defines a likelihood function for observed choice data which can be used to perform parameter estimation but the likelihood function is poorly behaved in general, which makes the estimation problem difficult. Having noted that a linearized version of the likelihood function can be interpreted as a special type of Generalized Linear Model (GLM), a well-known class of statistical models, we studied the parameter estimation problem for the relevant form of GLM. We proved conditions under which the GLM's likelihood function is convex, which implies that the estimator can be implemented using off-the-shelf optimization routines and that it converges to the correct parameter value. We then showed that this estimator can be applied to the parameter estimation problem for the UCL algorithm by linearizing the heuristic function and that it can provide accurate estimates of the parameters in some regions of parameter space. This provides an estimator to complete the plant–observer model pair for human decision making.

In proving conditions under which the GLM's likelihood function is convex, we define a binary operation for block matrices, which is a special case of the Khatri-Rao product [52], and a block contraction operator for block matrices. As far as we are aware, both of these operators are novel contributions. We elucidate some of the important properties of these operators and use them to prove a result analogous to the Schur product theorem [106] for block matrices. These results are contributions to the mathematics literature in their own right.

In Chapter 7, we returned to the study of decision-making behavior and studied the multi-armed bandit problem with a satisficing objective. Satisficing refers to an alternative to optimization in which the objective is not to achieve the best possible outcome but rather to achieve an outcome that is above a desired threshold. It has been studied in a variety of contexts including in psychology, organizational theory, engineering, and ecology, and can provide a more natural objective than optimization. We showed that the multi-armed bandit problem with a satisficing objective could be related to a standard multi-armed bandit problem with optimizing objective. Using the related standard multi-armed bandit problem, we proved bounds on optimal performance in terms of the satisficing objective and derived conditions under which an adapted version of the UCL algorithm achieved optimal performance. This result is important as an analysis of optimal decision making, and also because it strengthens the connection between the multi-armed bandit problem and problems such as optimal foraging behavior in ecology. It is also a first step towards a more general theory of multi-armed bandits with a general utility function objective.

8.2 Ongoing and future work

The questions addressed in this thesis have raised a variety of followup questions that are the subject of ongoing work. These questions span the fields of decision theory, neuroscience, and a number of engineering applications.

8.2.1 Decision theory

The work presented in this thesis leads naturally to a number of questions in decision theory. In decision theory the generic question is, for a given decision-making situation, what is the optimal strategy to take, and how do the various aspects of that strategy trade off against each other? In the context of multi-armed bandit problems, the work of Lai and Robbins provides bounds on the performance of optimal strategies and the various UCB algorithms from the literature and the UCL algorithms from this thesis provide strategies that achieve optimal performance.

However, there are a number of issues with the UCB algorithms, notably robustness. The standard multi-armed bandit problem optimizes an objective which consists of the expected value of cumulative rewards. Therefore, good performance in this problem corresponds to having large rewards "on average", where the average is taken over many instantiations of the problem. However, having good average performance does not imply having good performance on any individual task. Standard UCB algorithms that achieve logarithmic regret, i.e., optimal average performance, are known to have poor risk-aversion characteristics in that they occasionally suffer from anomalously bad performance due to an unlucky series of rewards [8]. These instances of bad performance are rare enough not to adversely affect the average performance of the algorithm but could be an issue if a bound on worst-case performance is required.

In this thesis we have studied two mechanisms that could improve the robustness of algorithms for solving multi-armed bandit problems: noise and satisficing. In the simulations presented in Chapter 3 we noted a tradeoff between information-based exploration based on the agent's statistical model of the task and noise-based exploration that operated in a model-free way. We postulated that the inclusion of decision noise provided a mechanism for exploration that could compensate for an incorrect statistical model, thereby increasing robustness at the cost of decreased average performance. Decision noise is known to improve the robustness of optimization algorithms [88], and it has recently been shown [129] that human decision-making behavior in multi-armed bandit tasks is stochastic. Therefore, providing an analytical quantification of the tradeoff between robustness and performance would be a major contribution to the field of decision theory and provide an explanation for

the stochasticity of human decision making. As far as this author is aware, deriving such an analytical quantification is an open research question.

Another mechanism for improving robustness is the use of a satisficing objective, as studied in Chapter 7. By maximizing the probability that the reward from a given decision is above a threshold value, the satisficing objective naturally improves worst-case performance relative to the standard objective which maximizes expected value of rewards. Further work needs to be done to quantify this improvement.

8.2.2 Neuroscience

The work presented in this thesis has been strongly connected with concurrent work in neuroscience. The heuristics used by the ambiguity bonus algorithm (Chapter 3) and the UCL algorithm (Chapter 4) were inspired by work in neuroscience that studied the heuristics used by humans to make decisions in multi-armed bandit problems. By rigorously developing these heuristics into algorithms with provably optimal performance, we have developed a model of human decision making that can serve as a reference to neuroscientists. Further work needs to be done to enhance the utility of this model in neuroscience.

First, fitting to human subject data should be performed. The stochastic UCL algorithm serves as a model of human decision making that depends on a number of parameters, including the decision noise parameter υ, the priors $(\boldsymbol{\mu}_0, \Sigma_0)$, and the credible limit parameter α_t. These parameters should be fit to data to better understand the strategies used by human subjects. The estimator developed in Chapter 6 provides a tool to do so.

Second, a connection should be made with the literature on psychological regret. As described in Chapter 2, the cumulative expected regret in the multi-armed bandit objective is a purely analytical quantity that is not directly psychologically relevant. However, there are other notions of regret that are psychologically relevant and are known to affect decision-making behavior [120, 32, 31, 75]. These notions of regret should be formally quantified and could enter as an additional term in the heuristic function of a new model.

Third, the structure of the UCL algorithm should be linked to the relevant neural hardware. The UCL algorithm provides a structured model of human decision making that is computationally simple and neurologically plausible. For example, approximate Bayesian inference should be implementable in the brain [130]. If UCL is an accurate model of what is truly occurring in the brain, then there should be neural signatures in areas corresponding to, e.g., information gain ΔI_i^t. These areas could be detected in experiments involving fMRI. Previous work has been done involving multi-armed bandit tasks and fMRI, for example [83], but work should investigate the structure of the UCL model.

8.2.3 Engineering

There are a host of potential applications of this work in various fields of engineering. In some situations, the representation of foraging as a multi-armed bandit problem can be used to develop more advanced purely automated systems, while the overall framework for human-machine collaboration can be used in other situations where human input is essential.

In robotics and autonomy, the spatial multi-armed bandit framework could provide a method to improve automation. For robotics applications, where each arm might represent a patch of physical space, it is likely that there would be a cost for transitioning from one arm to another, or that only a subset of the arms would only be accessible from a given currently-selected arm. In such a case, additional work needed to be done to develop algorithms that could handle the transition costs and motion constraints. This work has been done in [99], resulting in the block UCL and graphical block UCL algorithms, respectively. Some work has been done to apply the graphical block UCL algorithm to robotics [118], but more remains to be done.

To apply the multi-armed bandit framework to the original motivating problem in adaptive sensor networks, work needs to be done to interpret gathered information as reward. This will result in a reward surface that is time-varying in response to the arms that are selected by the algorithm, so additional work is required to address this case of time-varying rewards.

In a number of applications, which often involve solving optimization problems, humans are essential components to a system. For such applications the models developed in this thesis should be used to develop human-machine systems. Many optimization problems are formulated so that the objective function is convex, in which case there are many off-the-shelf algorithms that can solve them quickly. However, in many applications, the objective function is non-convex, and most optimization algorithms can only find local optima. The likelihood function from the stochastic UCL algorithm is one such example. In practice people often solve non-convex optimization problems by applying a local optimization algorithm at a variety of initial points and choosing the best solution that results. Fortunately, humans often have intuition about where the global optimum lies, which guides their choice of initial points. The process of finding a global optimum by selecting initial points can be cast in a multi-armed bandit framework by identifying patches of the optimization space as arms of the bandit and posing a so-called *best-arm identification* problem [7, 22]. By casting the optimization problem as a multi-armed bandit problem, one could rationalize this process and formally incorporate human intuition into the initial point selection problem. As we saw in Chapter 5, such intuition can drastically increase performance in a multi-armed ban-

dit problem, so it is reasonable to assume that similar benefits would result in the global optimization problem.

An additional application is in the field of multi-objective optimization, where there are several objectives to be optimized simultaneously. Because of the multiple objectives, a generic multi-objective optimization problem is not well-posed and requires additional information to identify a unique solution. This information must ultimately come from a human supervisor, so any multi-objective optimization system is inherently a human-machine system. By casting the multi-objective optimization problem as a best-arm identification multi-armed bandit problem, one could use the human-machine multi-armed bandit framework to rationalize the human-machine decision-making process as well. We have already taken beginning steps towards this goal [96, 93], and further work is ongoing.

8.3 Closing remarks

In this thesis, we have studied decision making under uncertainty with a focus on human decision making in spatial search tasks. We formally modeled such tasks using the *spatial multi-armed bandit* problem, and used results from the multi-armed bandit literature to link human and algorithmic decision making. This work is relevant both to neuroscience and to engineering.

For neuroscience, we provide mathematically rigorous models of human decision making (in particular the UCL algorithm) in multi-armed bandit problems. These models capture the main empirically-observed features of human decision making in such problems, and can be shown to achieve optimal performance with appropriate parameter tunings.

For engineering, we show that the quality of Bayesian decision making can be improved by the availability of a high-quality prior. This is not a new result, but importantly we show through experiments with human subjects that humans often have good priors for spatial search tasks which allow them to achieve better performance than an algorithm with an uninformative prior. Therefore, engineered systems could benefit from human intuition captured in the form of an informative prior. The UCL algorithm can be used as a model to learn a human operator's prior by observing their choices, thereby making the prior accessible to a machine.

The work presented in this thesis will form the basis for future work in neuroscience (helping further elucidate the behavioral and neural mechanisms behind human decision making) and engineering (providing a principled framework for human-machine systems, and a methodology for developing automation that can perform abstract decision making).

We hope that this thesis has shown the utility of collaboration between neuroscience and engineering and that it can be the basis of further mutually fruitful work.

Appendix A

Pseudocode implementations of the UCL algorithms

Algorithm 1: *Deterministic UCL Algorithm*

Input : prior $\mathcal{N}(\boldsymbol{\mu}_0, \sigma_0^2 I_N)$, variance σ_s^2;
Output : allocation sequence $\{i_t\}_{t\in\{1,\dots,T\}}$;

set $n_i \leftarrow 0, \bar{m}_i \leftarrow 0$, for each $i \in \{1,\dots,N\}$;

set $\delta^2 = \dfrac{\sigma_s^2}{\sigma_0^2}$; $K \leftarrow \sqrt{2\pi e}$; $T_0^{\text{end}} \leftarrow 0$;

% at each time pick the arm with maximum upper credible limit

for $\tau \in \{1,\dots,T\}$ **do**
  for $i \in \{1,\dots,N\}$ **do**
    $Q_i \leftarrow \dfrac{\delta^2\mu_i^0 + n_i\bar{m}_i}{\delta^2 + n_i} + \dfrac{\sigma_s}{\sqrt{\delta^2 + n_i}}\Phi^{-1}\Big(1 - \dfrac{1}{K\tau}\Big)$;
  $i_\tau \leftarrow \text{argmax}\{Q_i \mid i \in \{1,\dots,N\}\}$;
  collect reward m^{real};
  $\bar{m}_{i_\tau} \leftarrow \dfrac{n_{i_\tau}\bar{m}_{i_\tau} + m}{n_{i_\tau} + 1}$;
  $n_{i_\tau} \leftarrow n_{i_\tau} + 1$;

Algorithm 2: *Stochastic UCL Algorithm*

Input : prior $\mathcal{N}(\boldsymbol{\mu}_0, \sigma_0^2 I_N)$, variance σ_s^2;
Output : allocation sequence $\{i_t\}_{t\in\{1,\dots,T\}}$;

set $n_i \leftarrow 0, \bar{m}_i \leftarrow 0$, for each $i \in \{1,\dots,N\}$;

set $\delta^2 = \dfrac{\sigma_s^2}{\sigma_0^2}$; $K \leftarrow \sqrt{2\pi e}$; $T_0^{\text{end}} \leftarrow 0$;

% at each time pick an arm using Boltzmann probability distribution

for $\tau \in \{1,\dots,T\}$ **do**

  for $i \in \{1,\dots,N\}$ **do**

    $Q_i \leftarrow \dfrac{\delta^2\mu_i^0 + n_i\bar{m}_i}{\delta^2 + n_i} + \dfrac{\sigma_s}{\sqrt{\delta^2 + n_i}}\Phi^{-1}\Big(1 - \dfrac{1}{K\tau}\Big)$;

  $\Delta Q_{\min} = \min_{i,t} |Q_i - Q_j|$;

  $\upsilon_\tau \leftarrow \dfrac{\Delta Q_{\min}}{2\log\tau}$;

  select i_τ with probability $p_i \propto \exp(Q_i/\upsilon_\tau)$;

  collect reward m^{real};

  $\bar{m}_{i_\tau} \leftarrow \dfrac{n_{i_\tau}\bar{m}_{i_\tau} + m}{n_{i_\tau} + 1}$;

  $n_{i_\tau} \leftarrow n_{i_\tau} + 1$;

Bibliography

[1] D. Acuña and P. Schrater. Bayesian modeling of human sequential decision-making on the multi-armed bandit problem. In B. C. Love, K. McRae, and V. M. Sloutsky, editors, *Proceedings of the 30th Annual Conference of the Cognitive Science Society*, pages 2065–2070, Washington, DC, USA, July 2008.

[2] D. E. Acuña and P. Schrater. Structure learning in human sequential decision-making. *PLoS Computational Biology*, 6(12):e1001003, 2010.

[3] R. Agrawal. Sample mean based index policies with $O(\log n)$ regret for the multi-armed bandit problem. *Advances in Applied Probability*, 27(4):1054–1078, 1995.

[4] S. Agrawal and N. Goyal. Analysis of Thompson Sampling for the multi-armed bandit problem. In S. Mannor, N. Srebro, and R. C. Williamson, editors, *JMLR: Workshop and Conference Proceedings*, volume 23: COLT 2012, pages 39.1–39.26, 2012.

[5] K. J. Åström and B. Wittenmark. *Adaptive control*. Courier Dover Publications, Minneola, NY, 2013.

[6] J.-Y. Audibert and S. Bubeck. Minimax policies for adversarial and stochastic bandits. In S. Dasgupta and A. Klivans, editors, *Proceedings of the 22nd Annual Conference on Learning Theory (COLT)*, pages 217–226, 2009.

[7] J.-Y. Audibert and S. Bubeck. Best-arm identification in multi-armed bandits. In A. T. Kalai and M. Mohri, editors, *COLT: 23rd Conference on Learning Theory*, pages 41–53, 2010.

[8] J.-Y. Audibert, R. Munos, and C. Szepesvári. Exploration–exploitation tradeoff using variance estimates in multi-armed bandits. *Theoretical Computer Science*, 410(19):1876–1902, 2009.

[9] P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2):235–256, 2002.

[10] P. Auer and R. Ortner. Logarithmic online regret bounds for undiscounted reinforcement learning. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 49–56, Cambridge, MA, 2007. MIT Press.

[11] M. G. Azar, A. Lazaric, and E. Brunskill. Online stochastic optimization under correlated bandit feedback. In *Proceedings of the 31st International Conference on Machine Learning, JMLR W&CP 32 (1)*, pages 1557–1565, 2014.

[12] R. Bellman. *Dynamic Programming*. Princeton University Press, Princeton, NJ, 1957.

[13] D. Bertsimas and J. N. Tsitsiklis. Simulated annealing. *Statistical Science*, 8(1):10–15, 1993.

[14] D. Böhning. Multinomial logistic regression algorithm. *Annals of the Institute of Statistical Mathematics*, 44(1):197–200, 1992.

[15] R. Bordley and M. LiCalzi. Decision analysis using targets instead of utility functions. *Decisions in Economics and Finance*, 23(1):53–74, 2000.

[16] F. Bourgault, T. Furukawa, and H. F. Durrant-Whyte. Optimal search for a lost target in a Bayesian world. In S. Yuta, H. Asama, E. Prassler, T. Tsubouchi, and S. Thrun, editors, *Field and Service Robotics*, volume 24 of *Springer Tracts in Advanced Robotics*, pages 209–222. Springer Berlin Heidelberg, 2006.

[17] S. P. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[18] M. Brooks. The Matrix Reference Manual. [online] http://www.ee.imperial.ac.uk/hp/staff/dmb/matrix/intro.html, 2011.

[19] C. G. Broyden. The convergence of a class of double-rank minimization algorithms. *IMA Journal of Applied Mathematics*, 6(1):76–90, 1970.

[20] S. Bubeck and N. Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Machine Learning*, 5(1):1–122, 2012.

[21] S. Bubeck and C.-Y. Liu. Prior-free and prior-dependent regret bounds for Thompson sampling. In L. Bottou, C. Burges, M. Welling, Z. Ghahramani, and K. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, 2013.

[22] S. Bubeck, R. Munos, and G. Stoltz. Pure exploration in multi-armed bandits problems. In *Proceedings of the 20th International Conference on Algorithmic Learning Theory*, pages 23–37, 2009.

[23] S. Bubeck, R. Munos, G. Stoltz, and C. Szepesvari. $\mathcal{X}$-armed bandits. *The Journal of Machine Learning Research*, 12:1655–1695, 2011.

[24] M. Buhrmester, T. Kwang, and S. D. Gosling. Amazon's Mechanical Turk: A new source of inexpensive, yet high-quality, data? *Perspectives on Psychological Science*, 6(1):3–5, 2011.

[25] M. Cao, A. Stewart, and N. E. Leonard. Convergence in human decision-making dynamics. *Systems and Control Letters*, 59:87–97, 2010.

[26] Y. Carmel and Y. Ben-Haim. Info-gap robust-satisficing model of foraging behavior: Do foragers optimize or satisfice? *The American Naturalist*, 166(5):633–641, 2005.

[27] N. Cesa-Bianchi and P. Fischer. Finite-time regret bounds for the multiarmed bandit problem. In *Proceedings of the Fifteenth International Conference on Machine Learning*, pages 100–108, Madison, Wisconsin, USA, July 1998.

[28] H.-L. Choi. *Adaptive sampling and forecasting with mobile sensor networks*. PhD thesis, Dept. of Aeronautics and Astronautics, MIT, Cambridge, MA, 2008.

[29] H.-L. Choi and J. P. How. A multi-UAV targeting algorithm for ensemble forecast improvement. In *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, number AIAA-2007-6753, Hilton Head, SC, 2007.

[30] J. D. Cohen, S. M. McClure, and A. J. Yu. Should I stay or should I go? How the human brain manages the trade-off between exploitation and exploration. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 362(1481):933–942, 2007.

[31] M. D. Cohen. Learning with regret. *Science*, 319(5866):1052–1053, 2008.

[32] G. Coricelli, R. J. Dolan, and A. Sirigu. Brain, emotion and decision making: the paradigmatic example of regret. *Trends in Cognitive Sciences*, 11(6):258–265, 2007.

[33] N. D. Daw. Trial-by-trial data analysis using computational models. *Decision making, affect, and learning: Attention and performance XXIII*, 23:3–38, 2011.

[34] N. D. Daw, J. P. O'Docherty, P. Dayan, B. Seymour, and R. J. Dolan. Cortical substrates for exploratory decisions in humans. *Nature*, 441(7095):876–879, 2006.

[35] M. Evans, N. Hastings, and B. Peacock. *Statistical Distributions*. Wiley, 3rd edition, 2000.

[36] P. Fan. New inequalities of Mill's ratio and its application to the inverse Q-function approximation. *arXiv preprint arXiv:1212.4899*, Dec 2012.

[37] R. Fletcher. A new approach to variable metric algorithms. *The Computer Journal*, 13(3):317–322, 1970.

[38] K. Friston, P. Schwartenbeck, T. Fitzgerald, M. Moutoussis, T. Behrens, and R. J. Dolan. The anatomy of choice: Active inference and agency. *Frontiers in Human Neuroscience*, 7:598, 2013.

[39] J. Gittins, K. Glazebrook, and R. Weber. *Multi-armed Bandit Allocation Indices*. Wiley, 2nd edition, 2011.

[40] J. C. Gittins. Bandit processes and dynamic allocation indices. *Journal of the Royal Statistical Society. Series B (Methodological)*, 41(2):148–177, 1979.

[41] A. S. Goldberger. *A Course in Econometrics*. Harvard University Press, 1991.

[42] D. Goldfarb. A family of variable-metric methods derived by variational means. *Mathematics of Computation*, 24(109):23–26, 1970.

[43] M. Goodrich, W. Stirling, and R. Frost. A theory of satisficing decisions and control. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 28(6):763–779, 1998.

[44] M. B. Gordy. A comparative anatomy of credit risk models. *Journal of Banking & Finance*, 24(1):119–149, 2000.

[45] A. M. Hein and S. A. McKinley. Sensing and decision-making in random search. *Proceedings of the National Academy of Sciences*, 109(30):12070–12074, 2012.

[46] R. A. Horn and C. R. Johnson. *Matrix analysis*. Cambridge University Press, 1st edition, 1990.

[47] A. E. Joel et al. *A History of Engineering and Science in the Bell System*. Bell Telephone Laboratories, 1975.

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

[49] E. Kaufmann, O. Cappé, and A. Garivier. On Bayesian upper confidence bounds for bandit problems. In *International Conference on Artificial Intelligence and Statistics*, pages 592–600, La Palma, Canary Islands, Spain, Apr. 2012.

[50] S. M. Kay. *Fundamentals of Statistical Signal Processing, Volume I : Estimation Theory*. Prentice Hall, 1993.

[51] T. Keasar, E. Rashkovich, D. Cohen, and A. Shmida. Bees in two-armed bandit situations: Foraging choices and possible decision mechanisms. *Behavioral Ecology*, 13(6):757–765, 2002.

[52] C. Khatri and C. R. Rao. Solutions to some functional equations and their applications to characterization of probability distributions. *Sankhyā: The Indian Journal of Statistics, Series A*, 30(2):167–180, 1968.

[53] S. Kirkpatrick, C. D. G. Jr., and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

[54] R. Kleinberg. Nearly tight bounds for the continuum-armed bandit problem. In *Proceedings of the 18th Advances on Neural Information Processing Systems*, volume 697–704, 2004.

[55] R. Kleinberg and A. Slivkins. Sharp dichotomies for regret minimization in metric spaces. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms, SODA, 2010*, pages 827–846, 2010.

[56] R. Kleinberg, A. Slivkins, and E. Upfal. Bandits and experts in metric spaces. *arxiv:1312.1277*, 2013.

[57] A. Krause and C. E. Guestrin. Near-optimal nonmyopic value of information in graphical models. In *Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence*, pages 324–331, Edinburgh, Scotland, July 2005.

[58] J. R. Krebs, A. Kacelnik, and P. Taylor. Test of optimal sampling by foraging great tits. *Nature*, 275(5675):27–31, 1978.

[59] B. Krishnapuram, L. Carin, M. A. T. Figueiredo, and A. J. Hartemink. Sparse multinomial logistic regression: Fast algorithms and generalization bounds. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(6):957–968, 2005.

[60] P. Kumar. A survey of some results in stochastic adaptive control. *SIAM Journal on Control and Optimization*, 23(3):329–380, 1985.

[61] T. L. Lai. Adaptive treatment allocation and the multi-armed bandit problem. *The Annals of Statistics*, 15(3):1091–1114, 1987.

[62] T. L. Lai and H. Robbins. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, 6(1):4–22, 1985.

[63] K. Lange, D. R. Hunter, and I. Yang. Optimization transfer using surrogate objective functions. *Journal of Computational and Graphical Statistics*, 9(1):1–59, 2000.

[64] B. Lau and P. W. Glimcher. Dynamic response-by-response models of matching behavior in rhesus monkeys. *Journal of the Experimental Analysis of Behavior*, 84(3):555–579, 2005.

[65] M. D. Lee, S. Zhang, M. Munro, and M. Steyvers. Psychological models of human and optimal performance in bandit problems. *Cognitive Systems Research*, 12(2):164–174, 2011.

[66] P. M. Lee. *Bayesian Statistics: An Introduction*. John Wiley & Sons, 4th edition, 2012.

[67] N. E. Leonard, D. A. Paley, R. E. Davis, D. M. Fratantoni, F. Lekien, and F. Zhang. Coordinated control of an underwater glider fleet in an adaptive ocean sampling field experiment in Monterey Bay. *Journal of Field Robotics*, 27(6):718–740, 2010.

[68] N. E. Leonard, D. A. Paley, F. Lekien, R. Sepulchre, D. M. Fratantoni, and R. E. Davis. Collective motion, sensor networks, and ocean sampling. *Proceedings of the IEEE*, 95(1):48–74, 2007.

[69] F. L. Lewis, D. Vrabie, and K. Vamvoudakis. Reinforcement learning and feedback control: Using natural decision methods to design optimal adaptive controllers. *IEEE Control Systems Magazine*, 32(6):76–105, 2012.

[70] K. Liu and Q. Zhao. Extended UCB policy for multi-armed bandit with light-tailed reward distributions. *arXiv:1112.1768*, 2011.

[71] S. Liu. Matrix results on the Khatri-Rao and Tracy-Singh products. *Linear Algebra and its Applications*, 289(1):267–277, 1999.

[72] S. Liu and G. Trenkler. Hadamard, Khatri-Rao, Kronecker and other matrix products. *International Journal of Information and Systems Sciences*, 4(1):160–177, 2008.

[73] S. Lohr. Algorithms get a human hand in steering web. *The New York Times*, March 11:A1, 2013.

[74] W. S. Lovejoy. Computationally feasible bounds for Partially Observed Markov Decision Processes. *Operations Research*, 39(1):162–175, 1991.

[75] D. Marchiori and M. Warglien. Predicting human interactive learning by regret-driven neural networks. *Science*, 319(5866):1111–1113, 2008.

[76] T. McMillen, P. Simen, and S. Behseta. Hebbian learning in linear–nonlinear networks with tuning curves leads to near-optimal, multi-alternative decision making. *Neural Networks*, 24(5):417–426, 2011.

[77] D. Mitra, F. Romeo, and A. Sangiovanni-Vincentelli. Convergence and finite-time behavior of simulated annealing. *Advances in Applied Probability*, 18(3):747–771, 1986.

[78] T. M. Moe. The new economics of organization. *American Journal of Political Science*, 28(4):739–777, 1984.

[79] T. K. Moon. The expectation-maximization algorithm. *IEEE Signal Processing Magazine*, 13(6):47–60, 1996.

[80] G. Moore. Robots in arc welding. *Electronics and Power*, 31(4):279–282, April 1985.

[81] H. Nakayama and Y. Sawaragi. Satisficing trade-off method for multiobjective programming. In *Interactive decision analysis*, pages 113–122. Springer, 1984.

[82] M. R. Nassar and J. I. Gold. A healthy fear of the unknown: Perspectives on the interpretation of parameter fits from computational models in neuroscience. *PLoS Computational Biology*, 9(4):e1003015, 2013.

[83] A. Nedic. *Models for individual decision-making with social feedback*. PhD thesis, Dept. of Electrical Engineering, Princeton Univ., Princeton, NJ, 2011.

[84] A. Nedic, D. Tomlin, P. Holmes, D. A. Prentice, and J. D. Cohen. A decision task in a social context: Human experiments, models, and analyses of behavioral data. *Proceedings of the IEEE*, 100(3):713–733, 2012.

[85] J. Nelder and R. Wedderburn. Generalized linear models. *Journal of the Royal Statistical Society. Series A (General)*, 135(3):370–384, 1972.

[86] W. K. Newey and D. McFadden. Large sample estimation and hypothesis testing. In R. F. Engle and D. L. McFadden, editors, *Handbook of Econometrics*, volume 4, chapter 36, pages 2111–2245. Elsevier, 1994.

[87] A. Y. Ng and S. J. Russell. Algorithms for inverse reinforcement learning. In *The International Conference on Machine Learning*, pages 663–670, 2000.

[88] V. Nissen and J. Propach. On the robustness of population-based versus point-based optimization in the presence of noise. *IEEE Transactions on Evolutionary Computation*, 2(3):107–119, Sep 1998.

[89] H. V. Poor. *An introduction to signal detection and estimation*. Springer, New York, 2nd edition, 1998.

[90] W. B. Powell. *Approximate Dynamic Programming: Solving the curses of dimensionality*. John Wiley & Sons, 2007.

[91] J. W. Pratt. Risk aversion in the small and in the large. *Econometrica: Journal of the Econometric Society*, pages 122–136, 1964.

[92] D. Racey, M. E. Young, D. Garlick, J. N. Pham, and A. P. Blaisdell. Pigeon and human performance in a multi-armed bandit task in response to changes in variable interval schedules. *Learning & Behavior*, 39(3):245–258, 2011.

[93] A. S. Reddy. Integration of human preference into Multidisciplinary Design Optimization. B.S.E. thesis, Princeton Univ., 2013.

[94] P. Reverdy and N. E. Leonard. Parameter estimation in softmax decision-making models with linear objective functions. In preparation.

[95] P. Reverdy and N. E. Leonard. Satisficing in Gaussian bandit problems. In *Proceedings of the IEEE Conference on Decision and Control*, 2014. To appear.

[96] P. Reverdy, A. Reddy, L. Martinelli, and N. E. Leonard. Integrating a human designer's preferences in multidisciplinary design optimization. In *Proceedings of the 15th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, number AIAA 2014-2167, 2014.

[97] P. Reverdy, V. Srivastava, and N. E. Leonard. Modeling human decision-making in multi-armed bandits. In *Multidisciplinary Conference on Reinforcement Learning and Decision Making*, Princeton, NJ, 2013.

[98] P. Reverdy, V. Srivastava, and N. E. Leonard. Algorithmic models of human decision making in Gaussian multi-armed bandit problems. In *Proceedings of the 13th European Control Conference*, pages 2210–2215, Strasbourg, France, 2014.

[99] P. Reverdy, V. Srivastava, and N. E. Leonard. Modeling human decision-making in generalized Gaussian multi-armed bandits. *Proceedings of the IEEE*, 102(4):544–571, 2014.

[100] P. Reverdy, R. C. Wilson, P. Holmes, and N. E. Leonard. Towards optimization of a human-inspired heuristic for solving explore-exploit problems. In *Proceedings of the IEEE Conference on Decision and Control*, pages 2820–2825, Maui, HI, USA, 2012.

[101] H. Robbins. Some aspects of the sequential design of experiments. *Bulletin of the American Mathematical Society*, 58:527–535, 1952.

[102] S. M. Ross. *Introductory Statistics*. Academic Press, 3rd edition, 2010.

[103] S. Russell. Learning agents for uncertain environments. In *Proceedings of the Eleventh annual Conference on Computational Learning Theory*, pages 101–103. ACM, 1998.

[104] I. O. Ryzhov, W. B. Powell, and P. I. Frazier. The knowledge gradient algorithm for a general class of online learning problems. *Operations Research*, 60(1):180–195, 2012.

[105] K. Samejima, Y. Ueda, K. Doya, and M. Kimura. Representation of action-specific reward values in the striatum. *Science*, 310(5752):1337–1340, 2005.

[106] J. Schur. Bemerkungen zur Theorie der beschränken Bilinearformen mit unendlich nielen Veränderlichen. *Journal für die reine un angewandte Mathematik (Crelle's Journal)*, 140:1–28, 1911.

[107] B. Schwartz, A. Ward, J. Monterosso, S. Lyubomirsky, K. White, and D. R. Lehman. Maximizing versus satisficing: happiness is a matter of choice. *Journal of Personality and Social Psychology*, 83(5):1178–1197, 2002.

[108] D. F. Shanno. Conditioning of quasi-Newton methods for function minimization. *Mathematics of Computation*, 24(111):647–656, 1970.

[109] J. Sherman and W. J. Morrison. Adjustment of an inverse matrix corresponding to a change in one element of a given matrix. *Annals of Mathematical Statistics*, 21(1):124–127, 1950.

[110] H. A. Simon. A behavioral model of rational choice. *The Quarterly Journal of Economics*, 69(1):99–118, 1955.

[111] H. A. Simon. Rational choice and the structure of the environment. *Psychological Review*, 63(2):129–138, 1956.

[112] R. D. Smallwood and E. J. Sondik. The optimal control of Partially Observable Markov Processes over a finite horizon. *Operations Research*, 21(5):1071–1088, 1973.

[113] T. Smithers. Autonomy in robots and other agents. *Brain and Cognition*, 34(1):88–106, 1997.

[114] E. J. Sondik. The optimal control of Partially Observable Markov Processes over the infinite horizon: Discounted costs. *Operations Research*, 26(2):282–304, 1978.

[115] J. C. Spall. *Introduction to stochastic search and optimization: estimation, simulation, and control*. John Wiley & Sons, Hoboken, NJ, 2003.

[116] N. Srinivas, A. Krause, S. M. Kakade, and M. Seeger. Information-theoretic regret bounds for Gaussian process optimization in the bandit setting. *IEEE Transactions on Information Theory*, 58(5):3250–3265, 2012.

[117] V. Srivastava, P. Reverdy, and N. E. Leonard. On optimal foraging and multi-armed bandits. In *Proceedings of the 51st Annual Allerton Conference on Communication, Control, and Computing*, pages 494–499, Monticello, IL, USA, 2013.

[118] V. Srivastava, P. Reverdy, and N. E. Leonard. Surveillance in an abruptly changing world via multiarmed bandits. In *Proceedings of the IEEE Conference on Decision and Control*, 2014. To appear.

[119] V. Srivastava, P. B. Reverdy, and N. E. Leonard. Correlated and dynamic multiarmed bandit problems: Bayesian algorithms and regret analysis. In preparation.

[120] A. P. Steiner and A. D. Redish. Behavioral and neurophysiological correlates of regret in rat decision-making on a neuroeconomic task. *Nature Neuroscience*, Advance online publication, 2014. doi:10.1038/nn.3740.

[121] A. R. Stewart, M. Cao, A. Nedic, D. Tomlin, and N. E. Leonard. Towards human-robot teams: Model-based analysis of human decision making in two-alternative choice tasks with social feedback. *Proceedings of the IEEE*, 100(3):751–775, 2012.

[122] M. Steyvers, M. D. Lee, and E. Wagenmakers. A Bayesian analysis of human decision-making on bandit problems. *Journal of Mathematical Psychology*, 53(3):168–179, 2009.

[123] L. D. Stone, C. M. Keller, T. M. Kratzke, and J. P. Strumpfer. Search for the wreckage of Air France Flight AF 447. *Statistical Science*, 29(1):69–80, 2014.

[124] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1998.

[125] W. R. Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294, 1933.

[126] D. Tomlin, A. Nedic, M. T. Todd, R. C. Wilson, D. A. Prentice, P. Holmes, and J. D. Cohen. Group foraging task reveals separable influences of individual experience and social information. In *Neuroscience 2011 Abstracts*, Washington, DC, 2011.

[127] D. Ward. The role of satisficing in foraging theory. *Oikos*, pages 312–317, 1992.

[128] C. J. C. H. Watkins and P. Dayan. Q-learning. *Machine learning*, 8(3-4):279–292, 1992.

[129] R. C. Wilson et al. Human decision making in multi-armed bandit problems. Manuscript in preparation.

[130] R. C. Wilson and L. H. Finkel. A neural implementation of the Kalman filter. In *Advances in Neural Information Processing Systems 22*, pages 2062–2070, 2009.

[131] R. C. Wilson, A. Geana, J. M. White, E. A. Ludvig, and J. D. Cohen. Why the grass is greener on the other side: Behavioral evidence for an ambiguity bonus in human exploratory decision-making. In *Neuroscience 2011 Abstracts*, Washington, DC, Nov. 2011.

[132] R. C. Wilson and Y. Niv. Is model fitting necessary for model-based fMRI? Submitted, 2014.

[133] R. C. Wilson and M. T. Todd. Bayesian inference for Damon's task. Unpublished notes, March 2011.

[134] S. G. Winter. The satisficing principle in capability learning. *Strategic Management Journal*, 21(10–11):981–996, 2000.

[135] B. Yin et al. Finding optimal solution for satisficing non-functional requirements via 0-1 programming. In *Proceedings of the IEEE 37th Annual Computer Software and Applications Conference*, pages 415–424, 2013.

[136] S. Zhang and A. J. Yu. Cheap but clever: Human active learning in a bandit setting. In *Proceedings of the 35th Annual Conference of the Cognitive Science Society*, pages 1647–1652, Berlin, Germany, 2013.