Advances in decision-making under uncertainty: inference, finite-time analysis, and health applications

Yingfei Wang

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
Computer Science
Adviser: Warren B. Powell

June 2017

Abstract

This thesis considers the problem of sequentially making decisions under uncertainty, exploring the ways where efficient information collection influences and improves decision-making strategies. Most previous optimal learning approaches are restricted to fully sequential settings with Gaussian noise models where exact analytic solutions can be easily obtained. In this thesis, we bridge the gap between statistics, machine learning and optimal learning by providing a comprehensive set of techniques that span from designing appropriate stochastic models to describing the uncertain environment, to proposing novel statistical models and inferences, to finite-time and asymptotic guarantees, with an emphasis on how efficient information collection can expand access, decrease costs and improve quality in health care.

Specifically, we provide the first finite-time bound for the knowledge gradient policy. Since there are many situations where the outcomes are dichotomous, we consider the problem of sequentially making decisions that are rewarded by "successes" and "failures". The binary outcome can be predicted through an unknown relationship that depends on partially controllable attributes of each instance. With the adaptation of an online Bayesian linear classifier, we design a knowledge gradient (KG) policy to guide the experiment. Motivated by personalized medicine where a treatment regime is a function that maps individual patient information to a recommended treatment, hence explicitly incorporating the heterogeneity in need for treatment across individuals, we further extend our knowledge gradient policy to a Bayesian contextual bandits setting. Since the sparsity and the relatively small number of patients make learning more difficult, we design an ensemble optimal learning method, in which multiple models are strategically generated and combined to minimize the incorrect selection of a particularly poorly performing statistical model. Driven by numerous needs among materials science society, we developed a KG policy for sequential experiments when experiments can be conducted in parallel and/or

there are multiple tunable parameters which are decided at different stages in the process. Finally, we present a new Modular, Optimal Learning Testing Environment (**MOLTE**) as a public-domain test environment to facilitate the process of more comprehensive comparisons, on a broader set of test problems and a broader set of policies.

Acknowledgements

First and foremost, I would like to express my deep gratitude to my advisor and mentor, Professor Warren B. Powell, for his patience, enthusiasm, and immense knowledge. His great passion for research taught me how to conduct myself in every aspect of my academic life. In addition, my extended appreciation goes towards his constant support, his confidence in my scholarly abilities, and other numerous reasons which cannot be expressed in the space provided.

I would like to thank Professor Robert Schapire for guiding my research in the area of machine learning and for his support during my Ph.D. study. I have learnt a lot from his scientific insights and broad knowledge in machine learning, which laid the ground work for my own research. I am also grateful to the members of my committee, Professor Bernard Chazelle, Professor Mengdi Wang, Professor Han Liu and Professor Szymon Rusinkiewicz for their patience, support and helpful comments.

I also would like to express my sincere gratitude to my former teachers. I am sincerely grateful to everyone in the department of Computer Science and the department of Operations Research and Financial Engineering, for creating the intellectually stimulating atmosphere. I would particularly like to thank my fellow students and friends in the CASTLE Lab for meaningful discussions, and their constant supports. My special thanks go to my co-authors, Professor Chad Mirkin, Dr. Kris Reyes, Dr. Keith Brown, Dr. Chu Wang, Dr. Tsvetan Asamov and Dr. Yan Li, for our research collaborations.

Finally, I would like to dedicate this thesis to my husband who has been a constant source of support and encouragement in my life. I am truly thankful for having you in my life. Without you, I would not become the person I am today. I own my everlasting gratitude to my parents who have always loved and supported me unconditionally. I simply cannot imagine a life without them.

To my family.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

In this thesis, we consider the problem of sequentially making decisions under uncertainty, exploring the ways where efficient information collection influences and improves decision-making strategies. In sequential decision problems, at each time step, we choose one of finitely many alternatives and observe a random reward. The rewards are independent of each other and follow some unknown probability distribution. One goal can be to identify the alternative with the best expected performance within a limited measurement budget, which is the objective of offline ranking and selection. Another goal can be to maximize the expected cumulative sum of rewards obtained in a sequence of allocations, a problem class often addressed under the umbrella of multi-armed bandit problems. Both ranking and selection problems and bandit problems are examples of sequential decision making problems with partial information that address the exploration-exploitation trade-off. Since the learner does not know the true distribution of each alternative, it needs to explore the choices that might give good rewards in the future as well as exploit the alternatives that appear to be better based on previous observations.

Ranking and selection problems and/or multi-armed bandits arise in many settings. We may have to choose a type of material that has the best performance, the

features in a laptop or car that produce the highest sales, or the molecular combination that produces the most effective drug. In health services, physicians have to make medical decisions (e.g. a course of drugs, surgery, and expensive tests) to provide the best treatment. In online advertisements, the system would like to display advertisements to gather the most ad-clicks.

Despite the previous successes of applying optimal learning techniques to transportation, drug discovery, e-commerce, and material sciences, most approaches are restricted to fully sequential settings with non-parametric Gaussian noise models where exact analytic solutions can be easily obtained. My thesis work provides a comprehensive set of techniques that span from designing efficient optimal learning algorithms in parallel computing environments, to making decisions under parametric belief models which introduce additional computational hurdles, to finite-time and asymptotic guarantees, with an emphasis on how efficient information collection can expand access, decrease costs and improve quality in health care.

1.1 Background

There has been an enormous body of literature on the problem of optimizing the expectation of an unknown function $F(x, W)$ with each noisy observation depending on our choice $x \in \mathcal{X}$ and a random variable W. The utility function $F(x, W)$ can be understood as costs, rewards or losses. This is largely different from the case of deterministic optimization where the problem can be concisely formulated as:

$$\max_{x \in \mathcal{A} \subset \mathbb{R}^d} f(x),$$

with $\mathcal{A}$ a compact set and the objective function $f(x)$ typically assumed to be convex, or at least cheap to evaluate. Yet since many learning problems do not conform to these strong assumptions, in stochastic optimization, function evaluation is usu-

ally expensive, and the derivatives and convexity properties are not required. For example, assume that a doctor faces a discrete set of medical choices, and that we can characterize an outcome as a success (patient does not need to return for more treatment) or a failure (patient does need followup care such as repeated operations). Testing a medical decision may require several weeks to determine the outcome. This creates a situation where experiments are time-consuming and expensive, requiring that we learn from our decisions as quickly as possible.

The stochastic optimization problem has been studied in different communities, most of which actively choose the next decision point based on the previous observations (Bubeck and Cesa-Bianchi, 2012; Powell and Ryzhov, 2012; Brochu et al., 2010; Powell, 2016). An initial state S^0 is used to capture all information given as prior input. A policy π, also referred to as a decision function $X^\pi(S)$, is defined as a mapping from the states $S \in \mathcal{S}$ to decisions $x \in \mathcal{X}$. At each time step n, we use some policy to choose one alternative to measure $x^n = X^\pi(S^n)$ and receive a stochastic reward $\hat{F}^{n+1} = F(x^n, W^{n+1})$. After the decision and information, the system transitions to the state of belief at the next point in time according to some known transition function $S^{n+1} = S^M(S^n, x^n, \hat{F}^{n+1})$. This is a case of partial observation, meaning that we can only observe the value of the alternative we actually measured but not others. Hence it faces the classic exploration/exploitation dilemma: (1) recommend the decisions as effectively as possible while (2) learning to improve decisions in the future. Making what we think is currently the best decision may not be the best given the uncertainty in our model, forcing us to recognize that we have to learn to make better decisions in the future.

There are two ways to write an objective function:

- Terminal reward – considered in Bayesian optimization, ranking and selection problems, and also known as simple regret in multi-armed bandits. Here we assume have a limited budget of N function evaluations which have to be se-

quentially allocated over the different alternatives $x \in \mathcal{X}$ using a policy π. We denote the best solution with some policy as $X^{\pi,N}$. We can state the problem of finding the best experimental policy as

$$\max_{\pi} \mathbb{E}\Big[F(X^{\pi,N}, W)|S^0\Big]. \tag{1.1}$$

In this case, we are not punished for errors incurred during training and instead are only concerned with the final recommendation after the offline training phases. It should be noted that the expectation is over different sets of random variables. The first is the sequence of observations $W^1, \ldots, W^N$ which then produces the random $X^{\pi,N}$. The second expectation is over W in the equation, which is used to evaluate the solution. If a Bayesian approach is used, there is a third level of expectation over the prior.

- Cumulative reward – extensively studied under the umbrella of multi-armed bandits. If we have to experience the rewards while we do our learning/exploring, we may want to maximize contributions over some time horizon. The (online) objective function would be written as

$$\max_{\pi} \mathbb{E}\Big[\sum_{n=0}^{N-1} F(X^{\pi}(S^n), W^{n+1})|S^0\Big], \tag{1.2}$$

where the expectation is over the sequence of observations $W^1, \ldots, W^N$ and the prior if any.

Despite different styles of objective functions, a general algorithm for sequential decision problems can be summarized in Algorithm 1:

Bayesian optimization is a powerful strategy for optimizing objective functions that are expense to evaluate (Mockus, 1994; Jones et al., 1998; Jones, 2001; Gutmann, 2001). It is suitable for cases when one does not have a closed-form expression

Algorithm 1: General algorithm for sequential decision problems

input : time horizon N, initial state S^0, policy π, transition function S^M
for $n = 0$ *to* N **do**
  Select the point $x^n = X^\pi(S^n)$
  Observe $\hat{F}^{n+1} = F(x^n, W^{n+1})$
  Update the state $S^n = S^M(S^n, x^n, \hat{F}^{n+1})$
end

for the objective function, but one can obtain (possibly noisy) function evaluations at sampled points. For example, function evaluations can involve actual physical experiments. An alternative x can be a specific set of values of controllable parameters in a physical experiments. After choosing the values of controllable parameters, the experiments may take several weeks to run in order to gather one experimental result. In Bayesian optimization, a prior belief is incorporated to represent our knowledge about the space of possible objective functions, such as the smoothness and continuity. Let's define x^n as the nth sample, and Y^{n+1} as the noisy evaluation of the objective function at x^n. As we accumulate observations $D^n = \{x^t, y^{t+1}\}_{t=0}^n$, the posterior distribution of $\mu_x := \mathbb{E}F(x, W)$ can be obtained by the prior distribution and the likelihood function as follows:

$$P(\boldsymbol{\mu}|D^n) = P(D^n|\boldsymbol{\mu})P(\boldsymbol{\mu}).$$

To sample efficiently so as to minimize the number of function evaluations required, Bayesian optimization uses a decision function (or a policy) $X^\pi(S) : \mathcal{S} \mapsto \mathcal{X}$ to determine the next location $x^{n+1} = X^\pi(S^n)$ to sample. Examples of decision functions include expected improvement (Huang et al., 2006), Bayesian expected loss minimization (Osborne et al., 2009), probability of improvement (Kushner, 1964), Thompson sampling (Thompson, 1933), response surface/surrogate models (Gutmann, 2001; Jones, 2001; Regis and Shoemaker, 2005) and the knowledge gradient method (Frazier et al., 2008).

Raiffa and Schlaifer (1961) established the Bayesian framework for ranking and selection (R&S) problems. Several two-stage and sequential procedures exist for selecting the best alternative (*terminal reward*). Branke et al. (2007) made a thorough comparison of several fully sequential sampling procedures. They indicate that the optimal computing budget allocation (OCBA) (Chen et al., 1996, 2000; He et al., 2007) and value of information procedures (VIP) (Chick, 2001) perform quite well and better than a deterministic or two-stage policy (Chen et al., 2006). Another single-step Bayesian look-ahead policy first introduced by Gupta and Miescke (1996) and then further studied by Frazier et al. (2008) is called the "knowledge-gradient policy" (KG). It chooses to measure the alternative that maximizes the single-period expected value of information. Whereas the above mentioned policies assumed an independent normal or one-dimensional Wiener process prior on the alternatives' true means, Frazier et al. (2009) modified the knowledge-gradient policy to handle correlated multivariate normal belief on the mean values of these rewards.

Another similar setting is multi-armed bandit problems (Auer et al., 2002; Bubeck and Cesa-Bianchi, 2012; Filippi et al., 2010; Mahajan et al., 2012) for cumulative regret minimization in an online setting. The bandit problem was originally studied under Bayesian assumptions (Gittins, 1979). At each time step t, the learner selects a single action I_t and observes some payoff X_{t,I_t}. In stochastic multi-armed bandit, the reward of each arm is assumed to be drawn from some unknown probability distribution. The goal is to maximize the *cumulative reward* obtained in a sequence of n allocations over time, or equivalently minimizing the *regret*. A widely used class of policies for multi-armed bandit problems is called *upper confidence bounding* policies (UCB). The UCB algorithm is based on the principle of *optimism in the face of uncertainty*. That is, despite our lack of knowledge in what actions are best we will construct an optimistic guess as to how good the expected payoff of each action is, and pick the action with the highest guess. The "optimism" comes in the form of an

upper confidence bound which is the largest plausible estimate of the mean for each alternative. For example, for Bernoulli distributed alternatives, the index $\hat{\theta}_x^n + \sqrt{\frac{2\log n}{N_x^n}}$ of the first UCB policy (Agrawal, 1995; Auer et al., 2002) is the sum of two terms, with N_x^n the number of times the alternative x has been measured, up to time n. The first term is the average reward of each alternative x. The second term is related to the size (according to the Chernoff-Hoeffding inequality) of the one-sided confidence interval for the average reward within which the true expected reward falls with dominant probability. Different UCB-type variants have been developed for many types of reward distributions and have provable logarithmic regret bounds (Lai and Robbins, 1985; Auer et al., 2002; Kleinberg et al., 2010; Bubeck et al., 2012). Multi-armed bandits has been extensively studied, and many algorithms are proposed and have found applications in various domains. Some successful applications are news and movie recommendation and online advertising (Li et al., 2011; Chapelle and Li, 2011; Chu et al., 2011a).

Scientists can draw on an extensive body of literature on the classic design of experiments (DeGroot, 1970; Wetherill and Glazebrook, 1986; Montgomery, 2008) whose goal is to decide what observations to make when fitting a function. Yet in settings considered in this thesis, the decisions are guided by a well-defined utility function (that is, maximize the terminal/cumulative reward).

This thesis focuses on the field of optimal learning, exploring the ways where efficient information collection influences and improves decision-making strategies. This issue is especially essential in situations where information collection is very expensive. There are many real-world optimization tasks where observations are time consuming and/or expensive. One example arises in health services, where physicians have to make medical decisions (e.g. a course of drugs, surgery, and expensive tests). Assume that a doctor faces a discrete set of medical choices, and that we can characterize an outcome as a success (patient does not need to return for more treatment)

or a failure (patient does need followup care such as repeated operations). Testing a medical decision may require several weeks to determine the outcome. This creates a situation where experiments are time consuming and expensive, requiring that we learn from our decisions as quickly as possible. In contrast to most experimental work on UCB policies which tends to assume large observation budgets (which might fit applications such as optimizing ad-clicks), we argue that the setting of expensive experiments represents a different type of learning challenge. Some of the early work in this field includes literature on determining the optimal number of samples form an unknown distribution so as to answer a statistical question about that distribution. Advanced research covers a wide range of problem classes with both discrete and continuous decision spaces with different belief models. For example, Gupta and Miescke (1996) introduce the idea of maximizing the marginal value of information. Frazier et al. (2008) extend this idea to the knowledge gradient policy (KG), which is first proposed for offline (context-free) ranking and selection problems by maximizes the value of information, with the performance of each alternative represented by a (non-parametric) lookup table model. After its first appearance, KG has been extended to various belief models (e.g. hierarchical belief model in Mes et al. (2011), linear belief model in Negoescu et al. (2011)). Although originally developed for offline learning (where we do not pay attention to successes while we are learning), it is easily adapted to online learning where we seek to maximize the cumulative number of successes (Ryzhov et al., 2012).

1.2 Overview and Contributions

In many applications, decisions are made sequentially over time. For a rational decision maker, the perception of the "optimal" action should change as he observes the feedbacks of past decisions. This is a challenging problem for several reasons. First,

information has an economic value and yet information collection can be very expensive. This creates a situation where information should be measured and balanced against other economic concerns as part of the decision-making process, requiring that we learn from our decisions as quickly as possible. Second, we need to come up with appropriate stochastic models to describe the uncertain environment in which decisions can be implemented. Third, we need optimization to balance between exploiting short term earning and exploring information with long-term benefits. Finally, proper statistical models and inferences are required to represent our changing beliefs about the environment as new information collected, which is of high importance especially when high-dimension or sparsity is a crucial characteristic of the environment.

Most previous optimal learning approaches are restricted to fully sequential settings with non-parametric Gaussian noise models where exact analytic solutions can be easily obtained. As already mentioned, my thesis work provides a comprehensive set of techniques that span from designing efficient optimal learning algorithms in parallel computing environments, to making decisions under parametric belief models which introduce additional computational hurdles, to finite-time and asymptotic guarantees, with an emphasis on how efficient information collection can expand access, decrease costs and improve quality in health care. Here, we provide an overview and summarize the main contributions of the thesis.

Chapter 2: Finite-time Analysis for the Knowledge-Gradient Policy. Although many value of information policies exist with nice asymptotic guarantees and empirical performance, there is no finite-time bound for such policies mainly due to the adaptive nature of the policies, that is, the current decision depends on the stochastic outcomes of past decisions. We fill in this gap by offering a new perspective of interpreting ranking and selection problems as adaptive stochastic multi-set maximization problems and deriving the first finite-time bound of the knowledge-gradient,

which characterizes KG as a near-optimal algorithm with an approximation ratio of $e/(e-1) \approx 1.582$. In addition, we introduced the concept of prior-optimality and provide another insight into the performance of the knowledge gradient policy based on the submodular assumption on the value of information.

Chapter 3: Optimal Learning with Stochastic Binary Feedbacks. Since there are many situations where the outcomes are dichotomous, we consider the problem of sequentially making decisions that are rewarded by "successes" and "failures" which can be predicted through an unknown relationship that depends on a partially controllable vector of attributes for each instance. Our problem is motivated by real-world applications where observations are time consuming and/or expensive. We propose a stochastic binary feedback (success/failure) model and designed a knowledge gradient (KG) policy under Bayesian generalized linear models. Unlike prior work with the knowledge gradient which assumed Gaussian noise and/or linear belief models, the non-linearity introduced by the link functions causes additional computational hurdle. To this end, different analytical approximation methods are developed to overcome computational challenges. Theoretically, we provide a finite-time analysis and showed that the KG policy is asymptotically optimal. We demonstrate the performance of the proposed algorithm on both synthetic problems and benchmark UCI datasets.

Chapter 4: Bayesian Contextual Bandits for Personalized Health Care. We study the problem of how sequentially assignment of physicians/facilities to individual patients can reduce the health care costs. This is an example of the broader area of personalized medicine, which takes into consideration the heterogeneity in needs and responses of different patients. A treatment regime is a function that maps individual patient information (including measures of disease stage severity, medical history, clinical diagnosis) to a recommended treatment, hence explicitly capturing

patient characteristics on treatment decisions. Patient responses can be predicted through an unknown relationship that depends on the patient information and the selected treatment. The goal is to find the treatments that lead to the best patient responses, on average, over time. Each experiment is expensive, forcing us to learn the most from each health episode.

We describe a methodology for quickly learning a contextual, binary response model for personalized healthcare. We introduce a two-step Bellman's equation for Bayesian contextual bandits and develop an optimal learning policy to guide the treatment assignment by maximizing the expected value of information. Due to the intrinsic sparsity of health datasets, we use network modularity detection and LASSO to perform feature selection. A detailed study on knee replacement dataset demonstrates the significant value of an optimal learning policy to reduce health care costs.

Chapter 5: Ensemble Bayesian Optimization. As in the healthcare example, a patient can have a number of attributes, spanning from the age, weight, to diagnoses and to their medical history. If we directly use these features, the highly sparsity makes learning difficult and computationally expensive. We could instead find lower dimension feature representations based on previously learned patient profiles. Yet if a patient deviates from stereotypical patients, then a reduced space may not include enough explanation power. One question is how to appropriately choose the explanatory variables. Another essential question is what type of prediction model should be chosen among many competing models, such as perceptron, support vector machines (SVM), and decision trees. Ensemble learning is of vital importance in these cases. Since the high sparsity and the relatively small number of patients makes learning more difficult, with the adaptation of an online boosting framework, we use Bayesian

learning with expert advice as the belief model and develop optimal learning policies to sequentially make decisions, especially in high-dimensional settings.

Chapter 6: Parallel Knowledge Gradient Method for Nested-batch Bayesian Optimization. Most previous work in optimal learning assumes a fully sequential setting where at each time step only one decision is made. However, the sequential design fails to account for the ability to run several parallel experiments in batches. Driven by numerous needs among materials science society, we develop a Nested-Batch-KG policy for sequential experiments when experiments can be conducted in parallel and/or there are multiple tunable parameters which are decided at different stages in the process. We demonstrate the effectiveness of our approach on the material design problem of maximizing output current of a photoactive device.

Chapter 7: MOLTE, a Modular Optimal Learning Environment. There has been a long history in the optimal learning literature of proving some sort of bound, supported at times by relatively thin empirical work by comparing a few policies on a small number of randomly generated problems. To this end, we address the relative paucity of empirical testing of learning algorithms (of any type) by introducing a new public-domain, Modular, Optimal Learning Testing Environment (MOLTE) for Bayesian ranking and selection problem, stochastic bandits or sequential experimental design problems, to facilitate the process of more comprehensive comparisons, on a broader set of test problems and a broader set of policies.

Chapter 8: Conclusion. We summarized the conclusion and important extensions of this thesis and describes ongoing and future work. Specifically, it describes new optimal learning strategies for a wide range of belief models that are arisen from real-world applications of interest within healthcare, revenue management, and market research, and new theoretical directions.

1.3 Notes on Publication

The work included in this thesis has been submitted to academic conferences and journals, and is in various stages of review. Chapter 2 was submitted as Wang and Powell (2016a). Chapter 3 was submitted to Operations Research, with a conference version published as Wang et al. (2016). The work in Chapter 4 was submitted as Wang and Powell (2016c). The material in Chapter 6 was published as Wang et al. (2015). The environment presented in Chapter 7 is described in the technical report (Wang and Powell, 2016b). As of this writing, this work was also presented at the following conferences: INFORMS Annual Meeting (2014, 2015, 2016), International Conference on Machine Learning (ICML), INFORMS Optimization Society Conference, and Modeling and Optimization: Theory and Applications (MOPTA).

This research was supported in part by AFOSR grant contract FA9550-12-1-0200 for Natural Materials, Systems and Extremophiles and the program in Optimization and Discrete Mathematics.

Chapter 2

Finite-time Analysis for the Knowledge Gradient Policy

In this chapter, we consider sequential decision problems in which at each time step, we choose one of finitely many alternatives and observe a random reward with our goal as identifying the alternative with the best expected performance within a limited measurement budget (*terminal reward*). Since the learner does not know the true distribution of each alternative, it needs to explore the choices that might give good rewards in the future as well as exploit the alternatives that appear to be better based on previous observations.

We are particularly interested in a single-step Bayesian look-ahead policy, which is called the knowledge gradient policy introduced by Gupta and Miescke (1996) and then further studied by Frazier et al. (2008). Although the knowledge gradient policy has been extended in various ways and enjoy some nice theoretical properties, it has never been characterized by the type of regret bounds for which *upper confidence bounding* (UCB) policies (Lai and Robbins, 1985; Agrawal, 1995; Auer et al., 2002; Kleinberg et al., 2010; Bubeck et al., 2012), a widely used class of multi-armed bandit algorithm, are famous.

In what follows, we first establish the connection between Bayesian ranking and selection problem and adaptive stochastic multi-set function maximization problems where each multi-set corresponds to a set of selected alternatives. The multi-set representation captures our ability to evaluate the same alternative more than once. This new perspective offers a new line of analysis for the properties of value-of-information policies. We derive the first distribution-free finite-time bound for the knowledge gradient policy for R&S problems under the assumption that the utility function is adaptive submodular, which characterizes KG as a near-optimal algorithm with an approximation ratio of $e/(e-1) \approx 1.582$. However, adaptive submodularity, which is effectively a pathwise assumption, can fail in offline learning settings when the utility function itself involves a maximum. To this end, instead of the pathwise behavior analyses of the utility function, we further study its average behavior by taking expectations over the observations given any fixed sample allocation, resulting in a well-known quantity: the value of information. As a result, we introduce the concept of the prior-value of a policy and analyze the prior-optimality of the KG policy to provide another insight into its performance based on the submodular assumption of the value of information that is weaker than adaptive submodularity. To accomplish this, we build on the general structure of the analysis of greedy algorithms given in Nemhauser et al. (1978) and Golovin and Krause (2010). We demonstrate submodularity for the two-alternative case and provide other conditions for more general problems, filling in a gap in the analysis of the knowledge gradient policy. Finally, we propose experiments to illustrate our theoretical analysis on the finite time behavior of the knowledge gradient policy. We further compare the KG policy with other policies with or without theoretical guarantees. Aside from the fact that the KG policy performs competitively with or significantly better than other policies especially in early iterations, we draw the conclusion that there is no universal best policy for all problem classes, which means that theoretical guarantees are not by themselves

reliable indicators of which policy is best for a particular problem class and empirical experiments are needed to better understand their finite time performance.

2.1 Literature Review

We consider sequential decision problems in which at each time step, we choose one of finitely many alternatives and observe a random reward. The rewards are independent of each other and follow some unknown probability distribution. One goal can be to identify the alternative with the best expected performance within a limited measurement budget, which is the objective of Bayesian ranking and selection problems. Ranking and selection problems are examples of sequential decision making problems with partial information that address the exploration-exploitation trade-off. Since the learner does not know the true distribution of each alternative, it needs to explore the choices that might give good rewards in the future as well as exploit the alternatives that appear to be better based on previous observations.

Ranking and selection (R&S) problems arise in many settings. We may have to choose a type of material that has the best performance, the features in a laptop or car that produce the highest sales, or the molecular combination that produces the most effective drug. Often, the cost of a measurement may be substantial. Laboratory or field experiments may take a day or several weeks. For this reason, we assume we have a limited budget for making measurements.

Raiffa and Schlaifer established the Bayesian framework for R&S problems (Raiffa and Schlaifer, 1961). In this section, we are interested in a single-step Bayesian look-ahead policy, first introduced by Gupta and Miescke (1996) and then further studied by Frazier et al. (2008), called the "knowledge-gradient policy" (KG). It chooses to measure the alternative that maximizes the single-period expected value of information. In what follows, we first review the Bayesian R&S with terminal

reward objective function and the knowledge gradient policy with a lookup table belief model.

2.1.1 Ranking and Selection Problems

Suppose we have a collection $\mathcal{X}$ of M alternatives, each of which can be measured sequentially to estimate its unknown mean μ_x. We assume normally distributed measurement noise with known variance σ_W^2. We have a finite measurement budget of N. Our goal is to sequentially decide which alternative to measure so that when the measurement budget is exhausted, we have maximized our ability to find the best alternative.

We first introduce the model for independent Gaussian processes (GP). Although the cost function is unknown, it is reasonable to assume that there is some knowledge about some of its properties, such as smoothness. Since there is the possibility of measurement noise, which is assumed to be Gaussian white noise, a GP prior is well-suited due to conjugacy. In fact, Gaussian process priors for Bayesian optimization date back at least to the work of O'Hagan (1978). Mockus (1994) later set additional conditions for defining priori distributions: 1) continuity of the objective; 2) homogeneity of a priori distribution; 3) independence of m-th differences. This includes a very large family of optimization tasks and Mockus (1994) showed that a GP prior is well-suited to the task. In the meantime, the advantage of using Gaussian processes includes, but not limited to, having confidence intervals for predictions, usability, flexibility in implementation, and its ability to encode various linear models with different basis functions by choosing different kernel functions.

We begin with a normally distributed Bayesian prior belief on the sampling means that is independent across alternatives, $\mu_x \sim \mathcal{N}(\theta_x^0, \sigma_x^0)$. At the nth iteration, we choose one alternative $x^n = x$ to measure and observe $W^{n+1} = \mu_x + \epsilon_x^{n+1}$, where ϵ_x^{n+1}

is a Gaussian random measurement noise, $\epsilon_x^{n+1} \sim \mathcal{N}(0, \sigma_W)$. Here we assume that σ_W is known to the learner.

For convenience, we introduce the σ-algebras $\mathcal{F}^n$ for any $n = 0, 1, ..., N-1$ which is formed by the previous n measurement choices and outcomes, $x^0, W^1, ..., x^{n-1}, W^n$. We define $\theta_x^n = \mathbb{E}[\mu_x|\mathcal{F}^n]$ and $(\sigma_x^n)^2 = \text{Var}[\mu_x|\mathcal{F}^n]$. Then conditionally on $\mathcal{F}^n$, $\mu_x \sim \mathcal{N}(\theta_x^n, \sigma_x^n)$. Let $\beta_x^n = \frac{1}{(\sigma_x^n)^2}$ be the conditional precision of μ_x and our state of knowledge be $S^n = (\theta_x^n, \beta_x^n)_{x \in \mathcal{X}}$. After the nth measurement we update our beliefs using Bayes' rule (Gelman et al., 2014):

$$\theta_x^{n+1} = \begin{cases} \frac{\beta_x^n \theta_x^n + \beta^W W^{n+1}}{\beta^n + \beta^W} & \text{if } x^n = x \\ \theta_x^n & \text{otherwise,} \end{cases} \qquad \beta_x^{n+1} = \begin{cases} \beta^n + \beta^W & \text{if } x^n = x \\ \beta_x^n & \text{otherwise,} \end{cases}$$

where $\beta^W = 1/\sigma_W^2$.

We may impose correlated beliefs between alternatives in order to strengthen the effect of each measurement. Starting from a prior distribution $\mathcal{N}(\theta^0, \Sigma^0)$ and after measurement W^{n+1} of alternative x, by Bayes' theorem, a posterior distribution is also a normal distribution, with the mean and covariance matrix as follows (Gelman et al., 2014):

$$\theta^{n+1} = \Sigma^{n+1}\left((\Sigma^n)^{-1}\theta^n + \beta^W W^{n+1} e_x\right), \tag{2.1}$$

$$\Sigma^{n+1} = \left((\Sigma^n)^{-1} + \beta^W e_x e_x^T\right)^{-1}, \tag{2.2}$$

where e_x is the vector with 1 in the entry corresponding to alternative x and 0 elsewhere. $S^n = (\theta^n, \Sigma^n)$ is then our state of knowledge in this case. We may rewrite this formula using the Sherman-Morrison formula (see, e.g. Golub and Van Loan

(2012)) to obtain a recursion that does not require matrix inversion:

$$\theta^{n+1} = \theta^n + \frac{W^{n+1} - \theta^n_x}{\lambda^W + \Sigma^n_{xx}} \Sigma^n e_x, \tag{2.3}$$

$$\Sigma^{n+1} = \Sigma^n - \frac{\Sigma^n e_x (e_x)^T \Sigma^n}{\lambda^W + \Sigma^n_{xx}}, \tag{2.4}$$

where $\lambda^W = \sigma^2_W$ and Σ_{xx} is the variance of x in the covariance matrix Σ. An example of the update with four observations are illustrated in Figure 2.1.

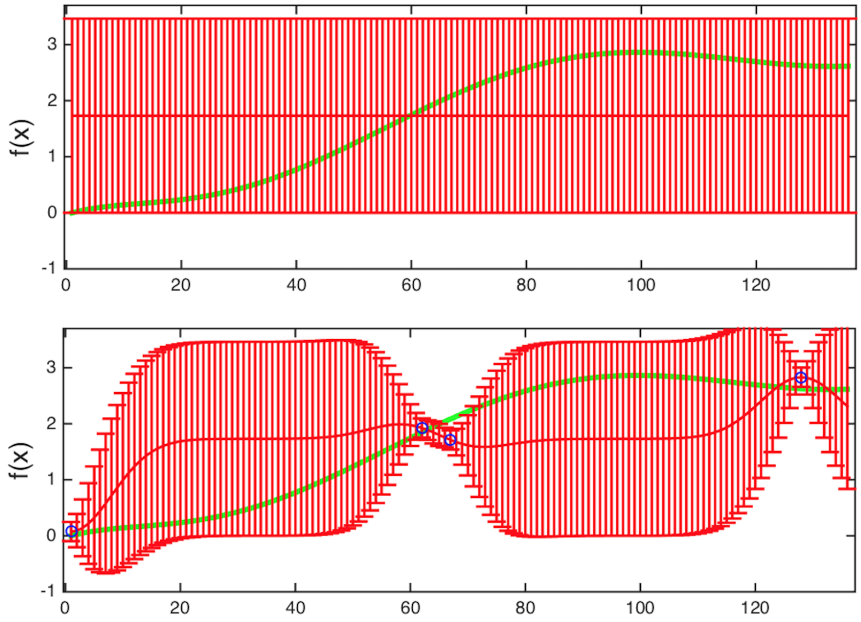


Figure 2.1: Sample 1-d Gaussian process with four observations. The green line is the true function values μ_x. The first figure represents the prior distribution and the the second figure is illustrate the posterior after the four observations. The solid red line is the GP surrogate mean prediction of the objective function given the observed data, and the error bar represents one standard deviation. The measured points and their observed values are circled in blue.

A policy π, also referred to as a decision function $X^\pi(S)$, is defined as a mapping from the states $S \in \mathcal{S}$ to decisions $x \in \mathcal{X}$. For example, for the case of independent beliefs, the state space $\mathcal{S}$ can be formally defined as $\mathcal{S} := \mathbb{R}^M \times (0, \infty]^M$. In other words, a policy sequentially guides our experiments by deciding which alternative to measure based on past observations. If we are limited to N measurements, the

objective is to find out the optimal policy that maximizes the expected reward of the final recommended alternative:

$$\max_{\pi \in \Pi} \mathbb{E}\left[\mu_{x^\pi}\right], \tag{2.5}$$

where $x^\pi \in \arg\max_{x \in \mathcal{X}} \theta_x^N$ and $x^n = X^\pi(S^n)$ for $0 \leq n < N$.

2.1.2 The Knowledge Gradient Policy

For R&S problems, the knowledge gradient is a policy that at the nth iteration chooses its $(n+1)$st measurement from $\mathcal{X}$ to maximize the single-period expected increase in value (Frazier et al., 2008, 2009). To be more specific, if we represent the state of knowledge at time n as $S^n = (\theta_x^n, \Sigma^n)$, then the value of being in state S^n is

$$V^n(S^n) = \max_{x \in \mathcal{X}} \theta_x^n.$$

If we choose to measure $x^n = x$ right now, allowing us to observe W_x^{n+1}, then we transition to a new state of knowledge $S^{n+1} = (\theta^{n+1}, \Sigma^{n+1})$. At iteration n, θ_x^{n+1} is a random variable since we do not yet know what W^{n+1} is going to be. We would like to choose x at iteration n which maximizes the expected value of $\max_{x \in \mathcal{X}} \theta_x^{n+1}$. We can think of this as choosing an alternative to maximize the incremental value, given by

$$\nu_x^{\mathrm{KG},n} = \mathbb{E}[\max_{x'} \theta_{x'}^{n+1} - \max_{x'} \theta_{x'}^n | x^n = x, S^n]. \tag{2.6}$$

The knowledge gradient policy $X^{\mathrm{KG}}(S^n)$ is defined by

$$X^{\mathrm{KG}}(S^n) \in \arg\max_{x \in \mathcal{X}} \nu_x^{\mathrm{KG},n}, \tag{2.7}$$

where ties are broken randomly.

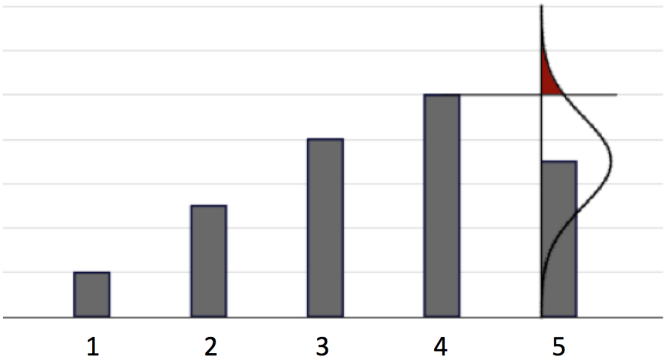


Figure 2.2: Illustration of the knowledge gradient if we were to measure choice 5.

The knowledge gradient, $\nu^{KG,n}$, is the amount by which the solution improves if we choose to measure alternative x. The knowledge gradient for independent normal belief is illustrated in Figure 2.2, where the posterior mean of the fourth alternative is currently the best. If we would like to measure alternative 5 at current time step, the estimated mean of alternative 5 will go up or down according to a normal distribution. The shaded area under the curve that exceeds the estimate of alternative 4 is the probability that measuring alternative 5 will produce a value that is better than the current best alternative. The knowledge gradient value is the expected amount by which it will increase.

An illustration of the KG acquisition function for correlated normal beliefs is shown in Figure 2.3. It can be seen from the figure that the KG value of a previously measured point will be decreased. In the meantime, if an alternative has a higher estimated mean, or a higher variance, the KG value tends to be higher, since it provides more value of information.

The knowledge gradient policy can handle the presence of a variety of belief models such as linear (Negoescu et al., 2011) or nonparametric (Mes et al., 2011; Barut and Powell, 2013).

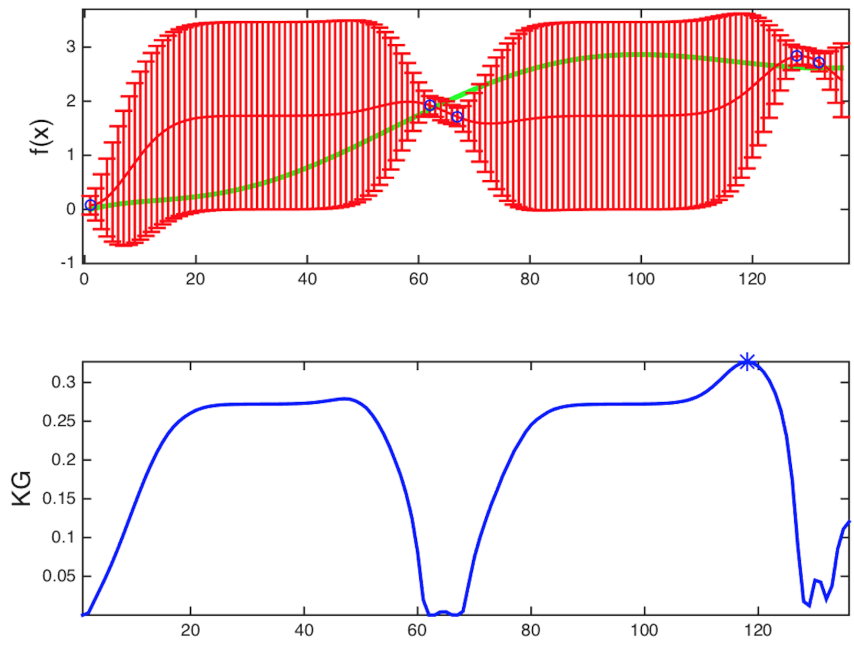


Figure 2.3: Examples of the knowledge gradient policy. The GP posterior after five measurements (highlighted in blue circles) is shown at the top. The other image shows the knowledge gradient value for the GP. The maximum is shown with a star.

The knowledge gradient policy has some nice properties (Frazier et al., 2009, 2008). For Bayesian ranking and selection problems, the knowledge gradient policy is optimal (by definition) if the measurement budget $N = 1$. The knowledge gradient is guaranteed to find the best alternative as the measurement budget N tends to infinity. If there are only two choices, the knowledge gradient policy is optimal for any measurement budget. The knowledge gradient policy is the only stationary policy that is both myopically and asymptotically optimal. However, the KG has not enjoyed the finite-time bounds that have been popular in the UCB policies.

2.2 Finite-time Analysis of the Knowledge Gradient Policy

We follow the general structure of the analysis of greedy approximation (Nemhauser et al., 1978; Goldengorin et al., 1999) to develop the first finite-time bound for the knowledge gradient policy for R&S problems. Nemhauser et al. (1978); Goldengorin et al. (1999) provide guarantees for simple greedy algorithms, which adds the element that maximally increases the objective value. Yet, in this chapter, the challenge lies in generalizing the results to adaptive planning, where the action taken in each step depends on the information collected in previous steps, and thus the feasible solutions are (adaptive) policies, that mapping from states to actions, rather than subsets. In Section 2.2.1, by interpreting the Bayesian R&S problems as the adaptive stochastic multi-set maximization problems, we demonstrate how the KG policy enjoys the performance guarantees similar to the greedy algorithm for classic nonadaptive submodular maximization problems, if the utility function is adaptive submodular. We theoretically analyze the adaptive submodular assumption and point out that it can fail in the ranking and selection problems. In such cases, instead of the pathwise behavior analyses of the utility function, we study its average behavior by taking expectation over the observations in Section 2.2.2. In Section 2.2.3, based on a well-understood quantity: value of information, we propose a natural definition of the prior value of an adaptive policy. We show how the results of Nemhauser et al. (1978); Goldengorin et al. (1999) generalize to the adaptive setting under the concept of *prior-optimality*. This provides another insight into the performance of the KG policy based on value of information.

It is important to note that both the submodular maximization reduction and the theoretical analyses on the prior-optimality are not limited to the specific setup of Gaussian noise in observations and Gaussian prior structure. The theoretical guar-

antees are more generally applicable to any prior and measurement noise model as long as the adaptive submodular assumption or the submodular value of information assumption holds.

2.2.1 The Reduction of R&S to Adaptive Stochastic Set Maximization

We first introduce the adaptive stochastic set maximization problem. Let E be a finite set of items. Each of the Φ_e is a random variable that maps the sample space Ω to a set O of possible values. We use Φ to denote the multivariate random variable $\Phi = (\Phi_e)_{e \in E}$. A realization is defined as $\phi := \Phi(\omega)$ with each ϕ_e representing the observation of item e in the ground set E. Under Bayesian interpretation, we assume that there is a known prior probability distribution $p(\phi) := \mathbb{P}(\Phi = \phi)$ over all possible realizations. The adaptive stochastic optimization problem consists of sequentially picking an item $e \in E$, revealing its outcome Φ_e and picking the next item. In adaptive stochastic set maximization problem, each item can be picked only once. After each pick, the observations so far can be represented as a partial realization ψ. For notational convenience, we sometimes also represent ψ as a relation, i.e. $\psi \subseteq E \times O$ equals $\{(e, o) : \psi_e = o\}$. A partial realization ψ is consistent with realization ϕ, denoted as $\phi \sim \psi$, if they are equal everywhere in the domain of ψ, where the domain (the set of items observed in ψ) of ψ is defined as $\mathrm{dom}(\psi) = \{e : \exists o, s.t.(e, o) \in \psi\}$. If ψ and ψ' are both consistent with ϕ, and $\mathrm{dom}(\psi) \subseteq \mathrm{dom}(\psi')$, then ψ is said to be a sub-realization of ψ', denoted as $\psi \subseteq \psi'$.

We wish to maximize some utility function $f : 2^E \times O^E \mapsto \mathbb{R}$ that depends on which items we pick and which states they are in. A policy π in this case is a function from a partial realization ψ to E, specifying which item to pick next based on previous observations. We use the notation $Z^\pi(\phi)$ to denote the set of items chosen by policy π under realization ϕ. The expected utility of a policy π is $f_{\mathrm{avg}}(\pi) := \mathbb{E}\big[f\big(Z^\pi(\Phi), \Phi\big)\big]$

where the expectation is taken over the prior distribution $p(\phi)$. The goal of adaptive stochastic set maximization problem is to find an optimal policy π^* that maximizes its expected utility under a cardinality constraint,

$$\pi^* \in \arg\max_{\pi} f_{\text{avg}}(\pi), \text{ subject to } |Z^{\pi}(\phi)| \leq N, \text{ for all } \phi,$$

where N is the measurement budget.

It is not obvious to treat the ranking and selection problem in an adaptive stochastic multi-set maximization way of thinking. To see this, define the ground set $E = \mathcal{X}$. The outcomes are real numbers with $O = \mathbb{R}$. Each alternative $e = x$ can be selected multiple times. After each selection, its random outcome $\Phi_e = W_x \in O$ is revealed.

Since the true values μ_x are assumed to be random variables in a Bayesian interpretation, we can let φ be a sample realization of the truth with a (correlated) prior distribution $p(\varphi) = \mathcal{N}(\theta^0, \Sigma^0)$. We use ϕ to denote a realization of the random observations in ranking and selection problems. The prior probability distribution over the realizations ϕ is determined by $p(\varphi)$ and the noise distribution $\mathcal{N}(0, \sigma_W)$. For example, if in the ranking and selection problems each alternative can only be selected only once, $\Phi = (\Phi_x)_{x \in \mathcal{X}}$. Since in ranking and selection problems, we can choose each alternative more than once, one way of defining the realization is by first making replicas of each item to construct $\mathcal{X}'$ and then selecting each $x' \in \mathcal{X}'$ at most once. To be more specific, it is equivalent to a set representation if we make N replicas of each alternative $x \in \mathcal{X}$ to construct set $\mathcal{X}'$. At the same time, the prior distribution $p(\varphi)$ needs to be extended to the set of $\mathcal{X}'$ with the correlation of different replicas of each alternative x set to be one. The sample realization in this case should also be extended to the set $\mathcal{X}'$ such that $\phi' := \Phi'(\omega)$ with each $\phi'_{x'} \in O$ representing the observation of each item x' in the extended set $\mathcal{X}'$.

Consider any sampling allocation $z = (z_x)_{x\in\mathcal{X}}$, by which we measure alternative x for $z_x \in \mathbb{N}$ times. We use Z to represent its corresponding multi-set. Each of the sampling allocation Z corresponds to at least one subset of $\mathcal{X}'$. We use $Z^\pi(\phi')$ to refer to the alternatives selected by π under realization ϕ'. It is worth noting that if in the translation between ranking and selection problems and set maximization, we stipulate that if a policy π chooses an alternative x for the nth time, exactly the observation of the nth replica of x in $\mathcal{X}'$ is revealed, then $Z^\pi(\phi')$ maps to only one subset of $\mathcal{X}'$. Let θ^n be our vector of estimates of the means after n measurements according to allocation Z under realization ϕ', where $|Z| = n$. θ^n can be obtained according to the updating equation (2.1) and (2.2), and does not depend on the order of the allocations according to independency and the Bayes' theorem. It can thus be denoted as $\theta^n(Z, \phi') : \mathbb{N}^{\mathcal{X}} \times O^{\mathcal{X}\times\mathbb{N}} \mapsto \mathbb{R}^M$. The next lemma states the equivalence of $\mathbb{E}[\mu_{x^\pi}]$ and $\mathbb{E}[\max_x \theta_x^N]$, where $x^\pi \in \arg\max_x \theta_x^N$. Hence, the utility function $\tilde{f} : \mathbb{N}^{\mathcal{X}} \times O^{\mathcal{X}\times\mathbb{N}} \mapsto \mathbb{R}$ can be defined as $\max_x \theta_x^n(Z, \phi')$ and $\tilde{f}_{\text{avg}}(\pi) := \mathbb{E}\Big[\max_x \theta_x^N\big(Z^\pi(\Phi'), \Phi'\big)\Big]$. The R&S objective (2.5) can then be re-written as

$$\pi^* \in \arg\max_\pi \tilde{f}_{\text{avg}}(\pi), \text{ subject to } |Z^\pi(\phi')| \le N, \text{ for all } \phi'.$$

Lemma 2.2.1 (Chapter 4.4.2 of Powell and Ryzhov (2012)). *Let π be a policy, θ^n be the vector of estimates of the means after n measurements following policy π, and let $x^\pi \in \arg\max_x \theta_x^N$ be the alternative selected by the policy. Then*

$$\mathbb{E}[\mu_{x^\pi}] = \mathbb{E}[\max_x \theta_x^N].$$

If in ranking and selection problems, each alternative can be measured at most once, it can be seen that the definition of the knowledge gradient $\nu_x^{KG,n}$ (Eq. (7.1.5)) coincides with the *Conditional Expected Marginal Benefit* $\Delta(e|\psi)$ for stochastic set

maximization defined by Golovin and Krause (2010):

$$\Delta(e|\psi) := \mathbb{E}\Big[f\big(\mathrm{dom}(\psi) \cup \{e\}, \Phi\big) - f\big(\mathrm{dom}(\psi), \Phi\big)|\Phi \sim \psi\Big].$$

The knowledge gradient policy is thus in fact the adaptive greedy policy with uniform item costs, with a slight difference in the ability of selecting each item more than once. To this end, we generalize the definition of adaptive monotonicity and adaptive submodularity for set functions given by Golovin and Krause (2010) to the case of multi-selection as follows.

Definition 2.2.2 (Adaptive Monotonicity). *A function $\tilde{f} : \mathbb{N}^{\mathcal{X}} \times O^{\mathcal{X}\times\mathbb{N}} \mapsto \mathbb{R}$ is adaptive monotone with respect to distribution $p(\phi')$ if the conditional expected marginal benefit of any item is nonnegative: for all ψ and all $x \in \mathcal{X}$.*

$$\Delta(x|\psi) \geq 0.$$

Definition 2.2.3 (Adaptive Submodularity). *A function $\tilde{f} : \mathbb{N}^{\mathcal{X}} \times O^{\mathcal{X}\times\mathbb{N}} \mapsto \mathbb{R}$ is adaptive submodular with respect to distribution $p(\phi')$ if for all ψ and ψ' such that $dom(\psi) \subseteq dom(\psi')$ and both ψ, ψ' are consistent with some realization ϕ' (i.e. $\psi \subseteq \psi'$), we have that the conditional expected marginal benefit of any fixed item $x \in \mathcal{X}$ does not increase as more items are selected and observed,*

$$\Delta(x|\psi) \geq \Delta(x|\psi').$$

Let π^* be the optimal policy to R&S problems. By a similar argument as in Golovin and Krause (2010) for adaptive stochastic set maximization problem, we provide the first finite-time bound for the knowledge gradient policy for R&S problems as follows. This is the first bound that characterizes KG as a near-optimal algorithm with an approximation ratio of $e/(e-1) \approx 1.582$.

THEOREM 2.2.4 (Posterior optimality bound). *If* $\tilde{f} := \max_x \theta_x^n(Z, \phi')$ *is adaptive monotone and adaptive submodular with respect to the prior distribution* $p(\phi')$*, then*

$$\tilde{f}_{avg}(KG) > (1 - e^{-1})\tilde{f}_{avg}(\pi^*).$$

We next show that the instances generated by ranking and selection problems are adaptive monotone.

LEMMA 2.2.5. *In ranking and selection problems, the utility function* $\max_x \theta_x$ *is adaptive monotone with any general prior distribution.*

Proof. For any ψ, let $n = |\psi|$. For any general prior distribution, recall that $\theta_x^n = \mathbb{E}[\mu_x|\mathcal{F}^n]$ with $\mathcal{F}^n$ as the σ-algebra generated by the partial realization ψ. Then for any item $x \in \mathcal{X}$, $\Delta(x|\psi)$ can be rewritten as

$$\mathbb{E}\big[\max_{x'} \mathbb{E}[\mu_{x'}|\mathcal{F}^{n+1}] - \max_{x'} \mathbb{E}[\mu_{x'}|\mathcal{F}^n]|x^n = x, \mathcal{F}^n\big] = \nu_x^{\mathrm{KG},n}.$$

We notice that the function $\max_{x'} \mathbb{E}[\mu_{x'}|\mathcal{F}^{n+1}]$ is convex, so we have

$$\mathbb{E}\big[\max_{x'} \mathbb{E}[\mu_{x'}|\mathcal{F}^{n+1}]|x^n = x, \mathcal{F}^n\big] \geq \max_{x'} \mathbb{E}\big[\mathbb{E}[\mu_{x'}|\mathcal{F}^{n+1}]|x^n = x, \mathcal{F}^n\big]$$

by Jensen's inequality. Due to the properties of conditional expectations, we have

$$\mathbb{E}\big[\mathbb{E}[\mu_{x'}|\mathcal{F}^{n+1}]|x^n = x, \mathcal{F}^n\big] = \mathbb{E}[\mu_{x'}|\mathcal{F}^n].$$

Hence we have $\Delta(x|\psi) = \nu_x^{\mathrm{KG},n} \geq 0$. □

Even though intuition suggests that the utility function should be adaptive submodular in the amount of information collected, as we collect more information it is natural to expect that the marginal value of this information should decrease, yet

it is not always the case as shown in the next lemma. The proof can be found in Appendix A.1.

LEMMA 2.2.6. *For any independent normal prior distribution $p(\varphi)$ and nondegenerated noise distribution (i.e. $\sigma^W \neq 0$), there exists ψ, ψ' and $x \in \mathcal{X}$ such that $\psi \subseteq \psi'$ and $\Delta(x|\psi) < \Delta(x|\psi')$.*

It can be seen that the adaptive submodular assumption can fail in the ranking and selection problems with the special utility function $\tilde{f} = \max_x \theta_x^n(Z, \phi')$ that involves maximization itself. Hence, instead of the above pathwise behavior analyses of the utility function, we would like to study its average behavior by taking the expectation over the observations given any fixed sample allocation Z in the next section.

2.2.2 The Value of Information

For notational simplicity, we use $\phi(\Phi)$, instead of $\phi'(\Phi')$, to denote sample realizations (random variable) for multi-set functions for the rest of our manuscript. We define the pathwise value of information $\hat{v}(Z, \phi)$ as the incremental improvement over the best expected value that can be obtained without measurement, which is $\max_{x \in \mathcal{X}} \theta_x^0$,

$$\hat{v}(Z, \phi) := \max_{x \in \mathcal{X}} \theta_x^n(Z, \phi) - \max_{x \in \mathcal{X}} \theta_x^0.$$

The value of information $v(Z)$ is then defined to be

$$v(Z) := \mathbb{E}_\Phi[\hat{v}(Z, \Phi)],$$

where the expectation is taken over the prior distribution $p(\phi)$.

The value of information has a long history spanning the literatures of several disciplines. Stigler considers the value of information in economics when buyers search for the best price (Stigler, 1961). Howard laid the groundwork for the value of infor-

mation in a decision-theoretic context and spawned a great deal of work in this area (Howard, 1966). Yokota and Thompson gives a first comprehensive review of value of information analyses related to health risk management (Yokota and Thompson, 2004). Raiffa and Schlaifer poses the Bayesian R&S problem and defines the associated value of information (Raiffa and Schlaifer, 1961), which marked the beginning of a number of literature on the value of information within Bayesian R&S and the budgeted learning problem (Guttman et al., 1964; Kapoor and Greiner, 2005; Chen et al., 1996; Chick, 2001; Frazier et al., 2008).

Since the value of information is a multi-set function, we first generalize the definitions and properties of submodular set functions described by Nemhauser et al. (1978) to submodular multi-set functions.

DEFINITION 2.2.7. *Given a finite set E, a real-valued function g on the set of multi-sets over E is called submodular if for all multi-sets S and T whose elements belong to E,*

$$\rho_x(S) \geq \rho_x(T), \forall S \subseteq T, \forall x \in E,$$

where $\rho_x(S) \triangleq g(S \cup \{x\}) - g(S)$ is the incremental value of adding element x to the multi-set S.

PROPOSITION 2.2.1. *Each of the following statements is equivalent and defines a submodular multi-set function (S and T are multi-sets on E, x, $y \in E$):*

1. $\rho_x(S) \geq \rho_x(T), \forall S \subseteq T$ *and* $\forall x$.
2. $\rho_x(S) \geq \rho_x(S \cup \{y\}), \forall S, x, y$.
3. $g(T) \leq g(S) + \sum_{x \in T-S} \rho_x(S), \forall S \subseteq T$.
4. $g(T) \leq g(S) + \sum_{x \in T-S} \rho_x(S) - \sum_{x \in S-T} \rho_x(S \cup T - \{x\}), \forall S, T$.

This proposition follows from a similar proof of Proposition 2.1 in Nemhauser et al. (1978).

It is obvious that if $\theta_x^n(Z,\phi)$ is adaptive monotone or adaptive submodular with respect to $p(\phi)$, then so does $\hat{v}(Z,\phi)$. It is also easy to show that if $\theta_x^n(Z,\phi)$ is adaptive monotone or adaptive submodular with respect to $p(\phi)$, then by the law of total expectation, i.e. $\mathbb{E}[\mathbb{E}[U|V]] = \mathbb{E}[U]$ for an integrable random variable U and some random variable V, the value of information $v(Z)$ is monotone or submodular. We close this section by showing the monotonicity of the multi-set function v and leave the analysis of submodularity in Section 2.3.

LEMMA 2.2.8. **(Monotonicity of the value of information)**

For any sampling allocation Z_1 and Z_2, if $Z_1 \subseteq Z_2$, then $v(Z_1) \leq v(Z_2)$.

Proof. We prove the monotonicity of v by showing $v(Z) \leq v(Z \cup \{x^{n+1}\})$ for any allocation Z (with $\sum_{x\in\mathcal{X}} z_x = n$) and any additional measurement x^{n+1}. By the tower property,

$$\begin{aligned} v(Z \cup \{x^{n+1}\}) - v(Z) &= \mathbb{E}_\Phi[\mathbb{E}[\max_x \theta_x^{n+1}(Z \cup \{x^{n+1}\}) - \max_x \theta_x^n(Z) | \Phi \sim \psi_Z]] \\ &= \mathbb{E}_\Phi[\nu_x^{\mathrm{KG},n}], \end{aligned}$$

where ψ_Z is the partial realization with $\mathrm{dom}(\psi_Z) = Z$. The lemma follows from the adaptive monotonicity, $\nu_x^{\mathrm{KG},n} \geq 0$. □

2.2.3 Guarantees on the Prior-optimality of the Knowledge Gradient Policy

There are two ways to evaluate the value of a policy. The first, which we call the *posterior view*, conditions on the allocation $Z = Z^\pi(\Phi)$ that would have occurred under policy π for each sample path $\phi \in \Phi$. This is the more conventional approach for evaluating policies. The second, which we call the *prior view*, starts by characterizing the value of an arbitrary allocation Z (before we have seen any sample realizations).

More formally, the classical way to estimate the value of a policy is to calculate the incremental improvement over what we could do before we collect any information, is given by

$$f'_{\text{avg}}(\pi) \quad = \quad \mathbb{E}[\tilde{f}(Z^\pi(\Phi), \Phi)] - \max_x \theta^0_x.$$

We let $\mathbb{P}(\pi \rightsquigarrow Z)$ be the probability that policy π produces allocation Z. Since with a fixed budget of N measurements, the number of possible allocations is finite, using the tower property, we can condition on the allocation $Z^\pi = Z$ which gives us

$$f'_{\text{avg}}(\pi) \quad = \quad \sum_{Z \in \mathcal{Z}^N} \mathbb{P}(\pi \rightsquigarrow Z) \Big(\mathbb{E}[\max_x \theta^n_x(Z^\pi(\Phi), \Phi) | Z^\pi = Z] - \max_x \theta^0_x \Big),$$

where $\mathcal{Z}^N$ is the set of all possible allocations with a limited budget N. We note that in this method for evaluating a policy (which is the standard method), we only consider allocations Z that are actually produced by policy π for the outcomes in ϕ. This approach makes it much more difficult to understand the relationship between the allocation Z and the value of a policy.

For this reason, we adopt a different method of evaluating a policy which we term the *prior view*. Since this idea is new, we define it formally as follows

DEFINITION 2.2.9 (The prior-value of a policy). *Let $\mathcal{Z}^n$ be the set of all possible allocations with a limited budget n. The value of a policy π with N measurements is defined as*

$$\begin{aligned} F^\pi \quad &:= \quad \mathbb{E}\big[v(Z^\pi)\big] = \sum_{Z \in \mathcal{Z}^N} \mathbb{P}(\pi \rightsquigarrow Z) v(Z) \\ &= \quad \sum_{Z \in \mathcal{Z}^N} \mathbb{P}(\pi \rightsquigarrow Z) \Big(\mathbb{E}_\Phi[\max_x \theta^n_x(Z, \Phi)] - \max_x \theta^0_x \Big). \end{aligned}$$

In this view, we use the prior probability of an outcome $p(\phi)$ instead of the posterior $p(\phi|Z^\pi(\phi) = Z)$ which is conditioned on an allocation Z. The value of this approach is that it writes the value of a policy directly as a function of $v(Z)$, making it easier to study the effect of the properties of $v(Z)$ on the value of a policy. Intuitively, since a policy could generate different allocations Z for different sample realizations, it is natural to define the value of a policy π as the weighted sum of the expected value of information based on all possible allocations Z and the weight should be the probability of occurrence of Z based on policy π. In comparison, we term the previous bound obtained for $\tilde{f}$ (or equivalently, f') in Section 2.2.1 as the *posterior-optimality*.

We make the following assumption which is weaker than the adaptive submodularity assumption and will analyze it further in Section 2.3.

ASSUMPTION 2.2.1. *The value of information v is a submodular multi-set function on the set of alternatives $\mathcal{X}$ with respect to the prior distribution $p(\phi)$:*

$$v(Z_1 \cup \{x\}) - v(Z_1) \geq v(Z_2 \cup \{x\}) - v(Z_2), \forall Z_1 \subseteq Z_2, \forall x \in \mathcal{X}.$$

Let π^* be the optimal sequential policy under a budget of N measurements in the sense that the prior-value of π^* is the largest. We call it *prior-optimality*. In what follows, we first bound KG's sub-*prior-optimality* in Proposition 2.2.1:

$$F^{\pi^*} \leq F^{\mathrm{KG}^{[n]} \odot \pi^*} \leq F^{\mathrm{KG}^{[n-1]}} + N(F^{\mathrm{KG}^{[n]}} - F^{\mathrm{KG}^{[n-1]}}),\ n = 1, 2, ..., N.$$

Then we derive the worst-case bound for the KG policy in Theorem 2.2.14:

$$\frac{F^{KG}}{F^{\pi^*}} \geq 1 - (\frac{N-1}{N})^N \geq \frac{e-1}{e} \approx 0.632.$$

Besides the *posterior-optimality* bound (Theorem 2.2.4) obtained from the adaptive stochastic multi-set maximization, the *prior-optimality* provides another insight into the performance of the KG policy based on a well-understood quantity: value of information.

DEFINITION 2.2.10 (Policy concatenation (Golovin and Krause, 2010)). *A concatenated policy $\pi = \pi_1 \odot \pi_2$ is constructed by running π_1 to completion, and then running policy π_2 from a fresh start ignoring all the information collected while running π_1.*

To be more specific, suppose π_i has a budget of n_i, $i = 1, 2$, the first phase is to run π_1 for n_1 iterations starting from S^0 and we get a sample realization including decisions and their corresponding measurements. The second phase is to run π_2 for n_2 measurements starting from S^0 and we get another sample realization. Thus the sample realization of the concatenated process is all the decisions and their corresponding measurements collected in two phases. Note here, when running the second policy, we ignore all the information collected during running the first one, but when calculating the value of $\pi_1 \odot \pi_2$, $F^{\pi_1 \odot \pi_2}$, we use all the information collected in two phases.

DEFINITION 2.2.11 (Policy truncation (Golovin and Krause, 2010)). *For a policy π, define the j-truncation $\pi^{[j]}$ of π as the policy that runs exactly $(j+1)$ steps under π's decision rule and $\pi^{\{j\}}$ as the single step policy that randomly chooses an alternative according to the probability distribution of policy π's decision for the $(j+1)$-th step.*

To be more specific, if the policy π is a deterministic policy, then for any sample realization ϕ, following π's decision rule for j steps, a $(j+1)$-th measurement decision can be made. Then $\pi^{\{j\}}$ is the corresponding random variable representing the $(j+1)$-th measurement decision with the probability distribution obtained by grouping all the sample realizations that chooses the same $(j+1)$-th alternative. If the policy

π is a randomized policy, the probability distribution will also take into account the randomness in π.

We now show that the value of π_1 is no larger than the value of $\pi_1 \odot \pi_2$.

LEMMA 2.2.12. *$F^{\pi_1} \leq F^{\pi_2 \odot \pi_1}$ for all policies π_1 and π_2 under any prior and probability distribution that describes a measurement.*

Proof. In a concatenated policy, the two phases are independent since no information is shared among the two phases. Thus by definition, we have $F^{\pi_1 \odot \pi_2} = F^{\pi_2 \odot \pi_1}$.

Therefore $F^{\pi_1} \leq F^{\pi_1 \odot \pi_2}$ holds if and only if $F^{\pi_1} \leq F^{\pi_2 \odot \pi_1}$. We have

$$\begin{aligned} F^{\pi_1 \odot \pi_2} - F^{\pi_1} &= \mathbb{E}\big[v(Z^{\pi_1 \odot \pi_2}) - v(Z^{\pi_1})\big] \\ &= \sum_{Z_1 \in \mathcal{Z}^{n_1}} \sum_{Z_2 \in \mathcal{Z}^{n_2}} \Big[v(Z_1 \cup Z_2) - v(Z_1)\Big] \mathbb{P}(\pi_1 \rightsquigarrow Z_1) \mathbb{P}(\pi_2 \rightsquigarrow Z_2) \\ &\geq 0, \end{aligned}$$

where inequality holds because of the monotonicity of multi-set function v. □

Based on the monotonicity of v and a similar argument as in Proposition 2.2.12, F is non-decreasing with respect to the number of measurements. Thus the more measurements, the better the policy. Hence π^* has exactly N measurements. We have the following sub-optimality bound on KG's prior-value. For a proof see Appendix A.2.

PROPOSITION 2.2.1. *Let $\rho^{KG,n} = F^{KG^{[n]}} - F^{KG^{[n-1]}}$, then*

$$\begin{aligned} F^{\pi^*} \leq F^{\ KG^{[n-1]} \odot \pi^*} &\leq F^{\ KG^{[n-1]}} + N\rho^{KG,n} \\ &= \sum_{i=0}^{n-1} \rho^{KG,i} + N\rho^{KG,n},\ n = 0, 1, ..., N-1. \end{aligned} \tag{2.8}$$

We now derive a bound for the adaptive greedy policy by applying linear programming to the problem of minimizing $\frac{F^{\mathrm{KG}}}{F^{\pi^*}}$ subject to the inequalities (2.8), which is a

worst-case analysis. The following lemma states the linear program and its solution. We use it afterwards to establish the bounds.

LEMMA 2.2.13. *Given $N \in \mathbb{Z}_+$, consider the following linear program*

$$\min \sum_{i=0}^{N-1} a_i,$$
$$\sum_{i=0}^{t-1} a_i + N a_t \geq 1, \ t = 0, 1, ..., N-1.$$

Then under these N constraints, min $\sum_{i=0}^{N-1} a_i = 1 - \alpha^N$, where $\alpha = \frac{N-1}{N}$.

The proof of this lemma can be found in Nemhauser et al. (1978).

We have the following results, which generalizes the classic result of the greedy algorithm that achieves $(1 - 1/e)$-approximation to prior-optimality for ranking and selection problems.

THEOREM 2.2.14. *Assume we have a budget of N measurements. Let π^* denote the optimal sequential policy for the ranking and selection problem, then we have*

$$\frac{F^{KG}}{F^{\pi^*}} \geq 1 - (\frac{N-1}{N})^N.$$

Proof. By Proposition 2.2.1, we have $F^{\pi^*} \leq \sum_{i=0}^{n-1} \rho^{\mathrm{KG},i} + N\rho^{\mathrm{KG},n}, \ n = 0, 1, ..., N-1$. Divide by F^{π^*} on both sides of this inequality, we have

$$1 \leq \sum_{i=0}^{n-1} \frac{\rho^{\mathrm{KG},i}}{F^{\pi^*}} + N\frac{\rho^{\mathrm{KG},n}}{F^{\pi^*}}, n = 0, 1, ..., N-1.$$

Let $a_i = \frac{\rho^{\mathrm{KG},i}}{F^{\pi^*}}$, and then these inequalities are identical to the constraints in Lemma 2.2.13. We notice that

$$\min \sum_{i=0}^{N-1} a_i = \min \sum_{i=0}^{N-1} \frac{\rho^{\mathrm{KG},i}}{F^{\pi^*}} \leq \sum_{i=0}^{N-1} \frac{\rho^{\mathrm{KG},i}}{F^{\pi^*}} = \frac{F^{\mathrm{KG}}}{F^{\pi^*}}.$$

By Lemma 2.2.13, we have $\min \sum_{i=0}^{N-1} a_i = 1 - \alpha^N$, so $\frac{F^{\mathrm{KG}}}{F^{\pi^*}} \geq 1 - \alpha^N = 1 - (\frac{N-1}{N})^N$. □

2.3 Analysis of Submodularity of the Value of Information

The finite-time bounds obtained in the previous sections assume that the value of information is submodular. In real world applications, submodularity is the diminishing returns property, meaning that in all productive processes, adding more of one factor of production, while holding all others constant, will at some point yield lower incremental per-unit returns. It can be applied in situations where there is an objective function to be optimized does not feature synergies in the benefits of items conditioned on observations. In general, submodularity does not hold for arbitrary value functions. We will show in this section that value of information for measuring a single alternative can be made concave by using sufficiently precise measurements. In what follows, we analyze the submodularity of the two-alternative case for independent beliefs.

While submodularity is a property for multi-set functions, we can extend it to any continuous function by making it possible for the increment to take any positive value. This allows us to use results from real analysis to study submodularity.

DEFINITION 2.3.1. *A function $f : \mathbb{R}^n \mapsto \mathbb{R}$ is submodular if for all $x, y \in \mathbb{R}^n$, $x_i \leq y_i$ and $\delta \in \mathbb{R}^n_+$,*

$$f(x+\delta) - f(x) \geq f(y+\delta) - f(y).$$

We show that submodularity of $\mathcal{C}^2$ functions is directly related to its second derivatives and cross-derivatives (the proof is given in Appendix A.3):

THEOREM 2.3.2. *$\mathcal{C}^2$ function f: $\mathbb{R}^n \to \mathbb{R}$ is submodular if and only if every element of its Hessian is non-positive.*

The concavity of the value of information has been studied extensively by Frazier and Powell (2010). In this section, we only study the cross-derivatives of the value of information.

Let $M = 2$ and the measurement allocation $z = (z_1, z_2)$. The value of information $v(z) = s(z)f(-\frac{|\theta_1^0 - \theta_2^0|}{s(z)})$, where $s(z) = \sqrt{\tilde{\sigma}_1^2(z_1) + \tilde{\sigma}_2^2(z_2)}$, $\tilde{\sigma}_i^2(z_i) = \frac{\sigma_i^{2,0} z_i}{\sigma_W^2/\sigma_i^{2,0} + z_i}$, $f(a) = a\Phi(a) + \phi(a)$, Φ and ϕ are the standard normal cumulative distribution and density respectively (Frazier and Powell, 2010).

Although the value of information is not concave in general in the two-alternative case, v is concave on the region where all z_i's are large enough (see Theorem 2 in Frazier and Powell (2010)).

We directly calculate the first derivative and cross-derivative of v as

$$\begin{aligned}
\frac{\partial v}{\partial z_1} &= \frac{\tilde{\sigma}_1(z_1)\tilde{\sigma}_1'(z_1)}{s(z)}\left[f(-\frac{|\theta_1^0 - \theta_2^0|}{s(z)}) + |\theta_1^0 - \theta_2^0|\frac{\Phi(-\frac{|\theta_1^0-\theta_2^0|}{s(z)})}{s(z)}\right], \\
\frac{\partial^2 v}{\partial z_1 \partial z_2} &= \frac{\tilde{\sigma}_1(z_1)\tilde{\sigma}_1'(z_1)\tilde{\sigma}_2(z_2)\tilde{\sigma}_2'(z_2)}{s^3(z)}\phi(-\frac{|\theta_1^0 - \theta_2^0|}{s(z)})\left(\frac{|\theta_1^0 - \theta_2^0|^2}{\tilde{\sigma}_1^2(z_1) + \tilde{\sigma}_2^2(z_2)} - 1\right).
\end{aligned}$$

In many optimal learning scenarios, we have very limited field knowledge about the performance of different alternatives and are only allowed a small number of sample points to estimate a prior distribution. In these cases, a common practice and a reasonable way is to adopt a uniform prior with the same mean value θ_x for all

alternatives. For the two alternative case, we can show that the value of information is submodular by noting the concavity of $v(z)$ (see Remark 2 in Frazier and Powell (2010)) and $\frac{\partial^2 v}{\partial z_1 \partial z_2} \leq 0$.

For other cases, $\frac{\partial^2 v}{\partial z_1 \partial z_2} \leq 0$ is equivalent to $|\theta_1^0 - \theta_2^0|^2 \leq \tilde{\sigma}_1^2(z_1) + \tilde{\sigma}_2^2(z_2)$. Rewriting this inequality, we get

$$\frac{1}{\frac{1}{\sigma_1^{2,0}} + \frac{z_1}{\sigma_W^2}} + \frac{1}{\frac{1}{\sigma_2^{2,0}} + \frac{z_2}{\sigma_W^2}} \leq \sigma_1^{2,0} + \sigma_2^{2,0} - |\theta_1^0 - \theta_2^0|^2. \tag{2.9}$$

We need $\sigma_1^{2,0} + \sigma_2^{2,0} - |\theta_1^0 - \theta_2^0|^2 \geq 0$, which can be achieved by setting our prior variance large enough or using a uniform prior over all alternatives. This is very reasonable when we have very little information about our problem domain.

Inequality equation (2.9) defines a region in the $z_1 - z_2$ plane. Specifically, this region has the hyperbolic line $\frac{1}{\frac{1}{\sigma_1^{2,0}} + \frac{z_1}{\sigma_W^2}} + \frac{1}{\frac{1}{\sigma_2^{2,0}} + \frac{z_2}{\sigma_W^2}} = \sigma_1^{2,0} + \sigma_2^{2,0} - |\theta_1^0 - \theta_2^0|^2$ as its boundary and contains infinity. In particular, when z_1 and z_2 are large enough, or when the variance of the measurement noise σ_W^2 is small enough, the value of information is submodular.

Since there is no closed-form expression for the value of information under arbitrary allocations, we cannot verify submodularity in a simple way for problems with more than two alternatives and for correlated beliefs. Instead, it can be checked using numerical approximation and is easy to guarantee by running repeated experiments and averaging to reduce measurement noise. A necessary condition is the concavity of the value of information for measuring a fixed alternative x for n times, which can be checked exactly.

Intuitively, we may expect that the marginal value of information should decline as we make more observations. But it is not always the case. It is shown that the value of information for measuring a single alternative may form an S-curve which is concave when there are many measurements, but may be convex at the beginning

(e.g. Theorem 1 in Frazier and Powell (2010)). The S-curve behavior arises when the measurement noise is large and thus a single measurement simply contains too little information, leading to algorithmic difficulties and apparent paradoxes. This issue is not related to any specific policy, but rather is an inherent property of learning problems. Although the value of information is not necessarily concave, it can be made concave by measuring each alternative enough times or (equivalently) using sufficiently precise measurements.

2.4 Computational Experiments

Since the seminal paper by (Lai and Robbins, 1985), there has been a long history in the optimal learning literature of designing algorithms with provable asymptotic or finite-time bounds (Audibert and Bubeck, 2010; Cappé et al., 2013; Auer et al., 2002; Audibert et al., 2009). But none of these bounds are tight in finite time and different bounds can be based on different metrics. Hence, empirical experiments are needed to better understand the finite time performance of each policy. To this end, we propose experiments to illustrate the finite time behavior of both KG and other optimal learning policies. We consider the following learning settings that arise a lot in black box Bayesian optimization.

Equal-prior: $M = 100$. The true values μ_x are uniformly distributed over $[0, 60]$ and measurement noise $\sigma_W = 100$. $\theta_x^0 = 30$ and $\sigma_x^0 = 10$ for every x.

Asymmetric unimodular function (AUF): x is a controllable parameter ranging from 21 to 120. The objective function is $F(x, \xi) = \theta_1 \min(x, \xi) - \theta_2 x$, where θ_1, θ_2 and the distribution of the random variable ξ are all unknown. The aim is to solve $\max_x \mathbb{E}F(x, \xi)$ while learning θ_1, θ_2 and the parameters that determine the distribution of ξ. The true distribution of ξ is taken as a normal distribution with mean 60 and standard deviation 18 (corresponding to a 30% noise ratio).

Goldstein-Price's function with additive noise:

$$\begin{aligned} f(x,y,\phi) &= [1+(x+y+1)^2(19-14x+3x^2-14y+6xy+3y^2)]\cdot \\ &\quad [30+(2x-3y)^2(18-32x+12x^2+48y-36xy+27y^2)]+\phi, \end{aligned}$$

where $-3 \le x \le 3$, $-3 \le y \le 3$ and are uniformly discretized into 13×13 alternatives.

In order to obtain the prior distribution, we follow Jones et al. (1998) and Huang et al. (2006) to use Latin hypercube designs for initial fit. For independent beliefs, we adopt a uniform prior with the same mean value θ_x^0 and standard deviation σ_x^0 for all alternatives. For correlated beliefs, we use a constant mean value θ_x^0 for all alternatives and a prior covariance matrix of the form

$$\Sigma_{xx'}^0 = \sigma e^{-\sum_{i=1}^d \lambda_i (x_i - x_i')^2},$$

where each arm x is a d-dimensional vector and σ, λ_i are constant. We adopt the rule of thumb by Jones et al. (1998) for the default number $(10 \times p)$ of points, where p is the number of parameters to be estimated. In addition, as suggested by Huang et al. (2006), to estimate the random errors, after the first $10 \times p$ points are evaluated, we add one replicate at each of the locations where the best p responses are found. Maximum likelihood estimation is then used to estimate the parameters based on the points in the initial design.

The policies considered in this section is described as follows. All the tunable parameters of different policies are tuned using a coarse-to-fine brute-force procedure to find the optimal value that yields the highest final reward, averaged over 1000 replicas, in the range of $10^{-5} \sim 10^5$.

EXPL: A pure exploration strategy that tests each alternative equally often.

EXPT: A pure exploitation strategy, $X^{\mathrm{EXPT},n}(S^n) = \arg\max_x \hat{\mu}_x^n$.

Interval Estimation (IE): (Kaelbling, 1993)

$$X^{\mathrm{IE},n}(S^n) = \arg\max_x \theta_x^n + z_{\alpha/2}\sigma_x^n.$$

Expected Improvement (EI): (Huang et al., 2006; Picheny et al., 2013)

Let $x^* = \arg\max_x(\theta_x^n + \sigma_x^n)$, then

$$X^{\mathrm{EI},n}(S^n) = \arg\max_x(\theta_x^n - \theta_{x^*}^n)\Phi(\frac{\theta_x^n - \theta_{x^*}^n}{\sigma_x^n}) + \sigma_x^n\phi(\frac{\theta_x^n - \theta_{x^*}^n}{\sigma_x^n}),$$

where ϕ and Φ are the standard normal density and cumulative distribution functions.

UCB-E: (Audibert and Bubeck, 2010)

$$X^{\text{UCB-E},n}(S^n) = \arg\max_x \hat{\mu}_x^n + \sqrt{\frac{\alpha}{N_x^n}},$$

where $\hat{\mu}_x^n$, N_x^n are the sample mean of μ_x and number of times x has been measured up to time n. The quantity $\hat{\mu}_x^0$ is initialized by measuring each alternative once.

SR: (Audibert and Bubeck, 2010) Let $A_1 = \mathcal{X}$, $\overline{\log}(M) = \frac{1}{2} + \sum_{i=2}^{M}\frac{1}{i}$,

$$n_m = \left\lceil \frac{1}{\overline{\log}(M)} \frac{n-M}{M+1-m} \right\rceil.$$

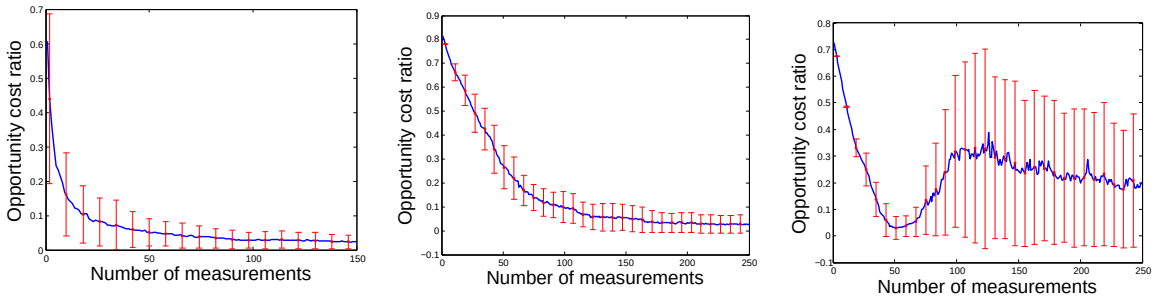
For each phase $m = 1, ..., M-1$:

1. For each $x \in A_m$, select alternative x for $n_m - n_{m-1}$ rounds.

2. Let $A_{m+1} = A_m \setminus \arg\min_{x\in A_m} \hat{\mu}_x$.

2.4.1 Finite Time Performance of Different Policies

Although the theoretical analysis in the previous section is to bound the performance of the knowledge gradient policy to the optimal policy (in theory), the optimal sequential policy is impossible to find in practice. To this end, we compare the value



(a) Equal-prior (b) AUF with $\theta_2 = 0.2\theta_1$ (c) AUF with $\theta_2 = 0.8\theta_1$

Figure 2.4: Opportunity cost ratio.

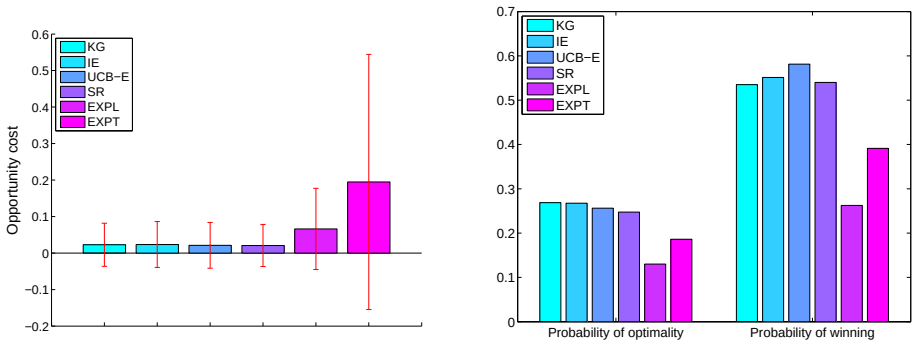
of KG to the expected value of the best alternative $\max_x \mu_x$. Define the opportunity cost (OC^π) of any policy π at any time step n as:

$$\text{OC}^\pi = \max_x \mu_x - \mu_{\tilde{x}^n},$$

where $\tilde{x}^n = \arg\max_x \theta_x^n$. We illustrate the finite time behavior of the KG policy under Equal-prior and AUF with independent normal beliefs. We run KG and calculate the opportunity cost ratio $= \frac{\max_x \mu_x - \mu_{\tilde{x}^n}}{\max_x \mu_x}$ in each iteration. We report the mean of the opportunity cost ratio averaged over 1000 experiments in Figure 2.4, with the error bar indicating one standard deviation.

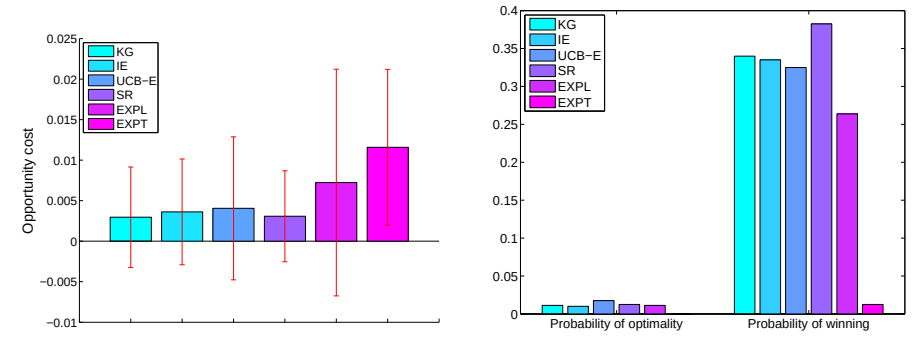
We next compare the performance of KG, IE with tuning, UCB-E with tuning, SR, EXPL and EXPT. Figure 2.5 shows the performance in problem classes AUF and Goldstein with independent beliefs under a measurement budget five times the number of alternatives. We run each policy for 1000 times. In each run, we pre-generate all the observations and share across different policies. We illustrate in the first column of Figure 2.5 the mean opportunity cost and the standard deviation of each policy over 1000 runs after the measurement budget is exhausted.

In order to give a comprehensive comparison based on different metrics, we also calculate the probability that the final recommendation of each policy is the optimal



(a) AUF: Opportunity cost

(b) AUF: Probability of optimality/winning



(c) Goldstein: Opportunity cost

(d) Goldstein:Probability of optimality/winning

Figure 2.5: Comparisons for AUF and Goldstein. (a) and (c) depict the mean opportunity cost with error bars indicating the standard deviation of each policy. The first bar group in (b) and (d) demonstrates the probability that the final recommendation of each policy is the optimal one. The second bar group in (b) and (d) illustrates the probability that the opportunity cost of each policy is the lowest.

one and the probability that the opportunity cost of each policy is the lowest, as illustrated in the figures on the right hand side of Figure 2.5.

The three criteria characterize the behavior of policies from different perspectives. For example, under AUF, if one cares about the average performance of the policy and its stability, SR is the best choice concluding from Figure 2.5 (a). Yet, if one can only run one trial (as in most cases of experimental science) and want to identify the best alternative, KG might be a better choice since it has the highest probability of finding the optimal alternative. Or if one can live with fairly good alternatives other than the optimal one, UCB-E could be the choice (although it has to be carefully

tuned). One observation is that there is no universal best policy for all problem classes or under all criteria, which means that theoretical guarantees are not by themselves reliable indicators of which policy is best for a particular problem class.

We further exploit correlated beliefs between alternatives in order to strengthen the effect of each measurement so that one measurement of some alternative can provide information for other alternatives.

First, we present the OC of different policies after each iteration under AUF ($\theta_2 = 0.5\theta_1$) in Figure 2.6. We tune z_α for IE and α for UCB for $N = 400$ measurements and the optimal values are $z_\alpha = 0.969$ and $\alpha = 6.657$. Since UCB-E needs to measure each alternative once, we omit the OC for its first 100 (which is the number of alternatives) steps. KG uses independent beliefs while KGCB, IE and EI start from MLE fitted correlated beliefs. When incorporating correlated beliefs, a measurement of one alternative tells us something about other alternatives. As a result, KGCB learns faster than KG and reaches a better performance. We draw the conclusion that correlated beliefs make learning faster and make learning possible for the case where the measurement budget is smaller (and potentially much smaller) than the number of alternatives.

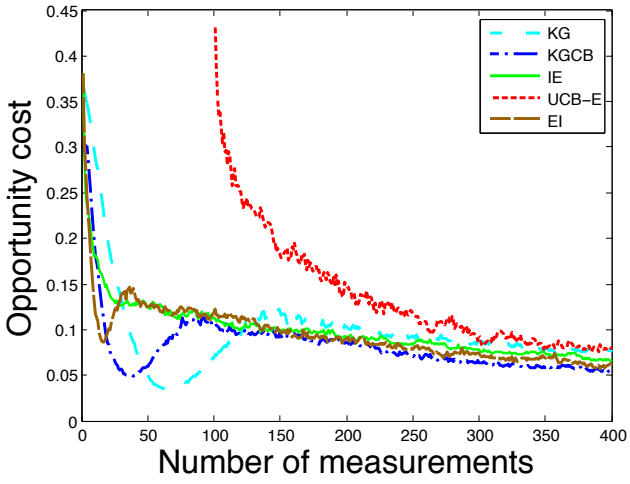


Figure 2.6: OC obtained after each measurement under AUF ($\theta_2 = 0.5\theta_1$).

2.5 Conclusion

In this chapter, we offer a new perspective of interpreting ranking and selection problems as adaptive stochastic multi-set maximization problems. We present the first finite-time bounds for the knowledge gradient on both the posterior optimality and the prior optimality. The prior view provides a cleaner relationship between the performance of the policy and the sample taken, making it possible to relate the value of information to the submodularity of the sample. Both the submodular maximization reduction and the theoretical analyses on the prior-optimality are not limited to the specific setup of Gaussian noise in observations and Gaussian prior structure, and are more generally applicable to any prior and measurement noise. We can infer from the bounds that KG is a near-optimal algorithm with an approximation ratio of 1.582. We analyze the submodularity of the two-alternative case and provide other conditions for more general problems, bringing out the issue and importance of submodularity in learning problems. We propose experiments to further illustrate the finite time behavior of the knowledge gradient policy as well as other policies with or without theoretical guarantees.

Chapter 3

Optimal Learning with Stochastic Binary Feedbacks

In this chapter, we consider the problem of sequentially making decisions that are rewarded by "successes" and "failures". The binary feedback can be predicted through an unknown relationship that depends on a partially controllable vector of attributes for each instance. The learner takes an active role in selecting samples from the instance pool. The goal is to maximize the probability of success in either offline (training) or online (testing) phases. Our problem is motivated by real-world applications where observations are time consuming and/or expensive.

A number of applications can easily fit into our success/failure model:

- Producing single-walled nanotubes. Scientists have physical procedures to produce nanotubes. It can produce either single-walled or double walled nanotubes through an unknown relationship with the controllable parameters, e.g. laser poser, ethylene, Hydrogen and pressure. Yet only the single-walled nanotubes are acceptable. The problem is to quickly learn the best parameter values with the highest probability of success (producing single-walled nanotubes).

- Personalized health care. We consider the problem of how to choose clinical pathways (including surgery, medication and tests) for different upcoming patients to maximize the success of the treatment.

- Minimizing the default rate for loan applications. When facing borrowers with different background information and credit history, a lending company needs to decide whether to grant a loan, and with what terms (interest rate, payment schedule).

- Enhancing the acceptance of the admitted students. A university needs to decide which students to admit and how much aid to offer so that the students will then accept the offer of admission and matriculate.

This chapter focus on offline settings such as laboratory experiments or medical trials where we are not punished for errors incurred during training and instead are only concerned with the final recommendation after the offline training phases. In Chapter 4, we then extend our discussion to online learning settings where the goal is to minimize cumulative regret and consider problems with partially controllable attributes, which is known as contextual bandits. For example, in the health care problems, we do not have control over the patients (which is represented by a feature vector including demographic characteristics, diagnoses, medical history) and can only choose the medical decision. A university cannot control which students are applying to the university. When deciding whether to grant a loan, the lending company cannot choose the personal information of the borrowers.

We investigate a knowledge gradient policy that maximizes the value of information, since this approach is particularly well suited to problems where observations are expensive. After its first appearance for ranking and selection problems (Frazier et al., 2008), KG has been extended to various other belief models (e.g. Mes et al. (2011); Negoescu et al. (2011); Wang et al. (2015)). Yet there is no KG vari-

ant designed for binary classification with parametric belief models. In this chapter, we extend the KG policy to the setting of classification problems under a Bayesian classification belief model which introduces the computational challenge of working with nonlinear belief models. We show that the maximum likelihood estimator based on the KG policy is consistent and asymptotically normal. We also show that the knowledge gradient policy is asymptotically optimal in an offline setting. We report the results of a series of experiments that demonstrate its efficiency.

3.1 Literature Review

Scientists can draw on an extensive body of literature on the classic design of experiments (DeGroot, 1970; Wetherill and Glazebrook, 1986; Montgomery, 2008) where the goal is to decide which observations to make when fitting a function. Yet in our setting, the decisions are guided by a well-defined utility function (that is, maximize the probability of success). The problem is related to the literature on active learning (Schein and Ungar, 2007; Tong and Koller, 2002; Freund et al., 1997; Settles, 2010), where our setting is most similar to membership query synthesis where the learner may request labels for any unlabeled instance in the input space to learn a classifier that accurately predicts the labels of new examples. By contrast, our goal is to maximize a utility function such as the success of a treatment. Moreover, the expense of labeling each alternative sharpens the conflicts of learning the prediction and finding the best alternative.

Another similar sequential decision making setting is multi-armed bandit problems (Auer et al., 2002; Bubeck and Cesa-Bianchi, 2012; Filippi et al., 2010; Mahajan et al., 2012; Srinivas et al., 2009; Chapelle and Li, 2011). Different belief models have been studied under the name of contextual bandits, including linear models (Chu et al., 2011a) and Gaussian process regression (Krause and Ong, 2011). The focus of bandit

work is minimizing cumulative regret in an online setting, while we consider the performance of the final recommendation after an offline training phase. There are recent works to address the problem we describe here by minimizing the simple regret. But first, the UCB type policies (Audibert and Bubeck, 2010) are not best suited for expensive experiments. Second, the work on simple regret minimization (Hoffman et al., 2014; Hennig and Schuler, 2012) mainly focuses on real-valued functions and do not consider the problem with stochastic binary feedbacks.

There is a literature on Bayesian optimization (He et al., 2007; Chick, 2001; Powell and Ryzhov, 2012). Efficient global optimization (EGO), and related methods such as sequential kriging optimization (SKO) (Jones et al., 1998; Huang et al., 2006) assume a Gaussian process belief model which does not scale to the higher dimensional settings that we consider. Others assume lookup table, or low-dimensional parametric methods, e.g. response surface/surrogate models (Gutmann, 2001; Jones, 2001; Regis and Shoemaker, 2005). The existing literature mainly focuses on real-valued functions and none of these methods are directly suitable for our problem of maximizing the probability of success with binary outcomes. A particularly relevant body of work in the Bayesian optimization literature is the expected improvement (EI) for binary outputs (Tesch et al., 2013). Yet when EI decides which alternative to measure, it is based on the expected improvement over current predictive posterior distribution while ignoring the potential change of the posterior distribution resulting from the next stochastic measurement (see Section 5.6 of Powell and Ryzhov 2012 and Huang et al. 2006 for detailed explanations).

3.2 Model

We assume that we have a finite set of alternatives $\boldsymbol{x} \in \mathcal{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_M\}$. The observation of measuring each $\boldsymbol{x}$ is a binary outcome $y \in \{-1, +1\}/\{\text{failure, success}\}$

with some unknown probability $p(y = +1|\boldsymbol{x})$. The learner sequentially chooses a series of points $(\boldsymbol{x}^0, \ldots, \boldsymbol{x}^{N-1})$ to run the experiments. Under a limited measurement budget N, the goal of the learner is to recommend an implementation decision $\boldsymbol{x}^N$ that maximizes $p(y = +1|\boldsymbol{x}^N)$.

We adopt probabilistic models for classification. Under general assumptions, the probability of success can be written as a link function acting on a linear function of the feature vector

$$p(y = +1|\boldsymbol{x}) = \sigma(\boldsymbol{w}^T\boldsymbol{x}).$$

In this paper, we illustrate the ideas using the logistic link function $\sigma(a) = \frac{1}{1+\exp(-a)}$ and probit link function $\sigma(a) = \Phi(a) = \int_{-\infty}^{a} \mathcal{N}(s|0, 1^2)\mathrm{d}s$ given its analytic simplicity and popularity, but any monotonically increasing function $\sigma : \mathbb{R} \mapsto [0, 1]$ can be used. The main difference between the two sigmoid functions is that the logistic function has slightly heavier tails than the normal CDF. Classification using the logistic function is called logistic regression and that using the normal CDF is called probit regression.

We start with a multivariate prior distribution for the unknown parameter vector $\boldsymbol{w}$. At iteration n, we choose an alternative $\boldsymbol{x}^n$ to measure and observe a binary outcome y^{n+1} assuming labels are generated independently given $\boldsymbol{w}$. Each alternative can be evaluated more than once with potentially different outcomes. Let $\mathcal{D}^n = \{(\boldsymbol{x}^i, y^{i+1})\}_{i=0}^n$ denote the previous measured data set for any $n = 0, \ldots, N$. Define the filtration $(\mathcal{F}^n)_{n=0}^N$ by letting $\mathcal{F}^n$ be the sigma-algebra generated by $\boldsymbol{x}^0, y^1, \ldots, \boldsymbol{x}^{n-1}, y^n$. We use $\mathcal{F}^n$ and $\mathcal{D}^n$ interchangeably. Note that the notation here is slightly different from the (passive) PAC learning model where the data are i.i.d. and are denoted as $\{(\boldsymbol{x}_i, y_i)\}$. Yet in our (adaptive) sequential decision setting, measurement and implementation decisions $\boldsymbol{x}^n$ are restricted to be $\mathcal{F}^n$-measurable so that decisions may only depend on measurements made in the past. This notation with superscript indexing time stamp is standard, for example, in control theory,

stochastic optimization and optimal learning. We use Bayes' theorem to form a sequence of posterior predictive distributions $p(\boldsymbol{w}|\mathcal{D}^n)$.

The next lemma states the equivalence of using true probabilities and sample estimates when evaluating a policy (Powell and Ryzhov, 2012).

LEMMA 3.2.1. *Let* Π *be the set of policies,* $\pi \in \Pi$*, and* $\boldsymbol{x}^\pi = \arg\max_{\boldsymbol{x}} p(y = +1|\boldsymbol{x}, \mathcal{D}^N)$ *be the implementation decision after the budget* N *is exhausted. Then*

$$\mathbb{E}_{\boldsymbol{w}}[p(y = +1|\boldsymbol{x}^\pi, \boldsymbol{w})] = \mathbb{E}_{\boldsymbol{w}}[\max_{\boldsymbol{x}} p(y = +1|\boldsymbol{x}, \mathcal{D}^N)],$$

where the expectation is taken over the prior distribution of $\boldsymbol{w}$*.*

By denoting $\mathcal{X}^I$ as an implementation policy for selecting an alternative after the measurement budget is exhausted, then $\mathcal{X}^I$ is a mapping from the history $\mathcal{D}^N$ to an alternative $\mathcal{X}^I(\mathcal{D}^N)$. Then as a corollary of Lemma 3.2.1, we have (Powell and Ryzhov, 2012)

$$\max_{\mathcal{X}^I} \mathbb{E}\big[p\big(y = +1|\mathcal{X}(\mathcal{D}^N)\big)\big] = \max_{\boldsymbol{x}} p(y = +1|\boldsymbol{x}, \mathcal{D}^N).$$

In other words, the optimal decision at time N is to go with our final set of beliefs. By the equivalence of using true probabilities and sample estimates when evaluating a policy as stated in Lemma 3.2.1, while we want to learn the unknown true value $\max_{\boldsymbol{x}} p(y = +1|\boldsymbol{x})$, we may write our objective function as

$$\max_{\pi \in \Pi} \mathbb{E}^\pi[\max_{\boldsymbol{x}} p(y = +1|\boldsymbol{x}, \mathcal{D}^N)]. \tag{3.1}$$

3.3 Background: Linear classification

Linear classification, especially logistic regression, is widely used in machine learning for binary classification (Hosmer Jr and Lemeshow, 2004). Assume that the

probability of success $p(y = +1|\boldsymbol{x})$ is a parameterized function $\sigma(\boldsymbol{w}^T\boldsymbol{x})$ and further assume that observations are independently of each other. Given a training set $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ with $\boldsymbol{x}_i$ a d-dimensional vector and $y_i \in \{-1, +1\}$, the likelihood function $p(\mathcal{D}|\boldsymbol{w})$ is $p(\mathcal{D}|\boldsymbol{w}) = \prod_{i=1}^n \sigma(y_i \cdot \boldsymbol{w}^T\boldsymbol{x}_i)$. The weight vector $\boldsymbol{w}$ is found by maximizing the likelihood $p(\mathcal{D}|\boldsymbol{w})$ or equivalently, minimizing the negative log likelihood:

$$\min_{\boldsymbol{w}} \sum_{i=1}^{n} -\log(\sigma(y_i \cdot \boldsymbol{w}^T\boldsymbol{x}_i)).$$

In order to avoid over-fitting, especially when there are a large number of parameters to be learned, l_2 regularization is often used. The estimate of the weight vector $\boldsymbol{w}$ is then given by:

$$\min_{\boldsymbol{w}} \frac{\lambda}{2}\|\boldsymbol{w}\|^2 - \sum_{i=1}^{n} \log(\sigma(y_i \cdot \boldsymbol{w}^T\boldsymbol{x}_i)). \tag{3.2}$$

It can be shown that this log-likelihood function is globally concave in $\boldsymbol{w}$ for both logistic regression or probit regression. As a result, numerous optimization techniques are available for solving it, such as steepest ascent, Newton's method and conjugate gradient ascent.

This logic is suitable for batch learning where we only need to conduct the minimization once to find the estimation of weight vector $\boldsymbol{w}$ based on a given batch of training examples $\mathcal{D}$. Yet due to the sequential nature of our problem setting, observations come one by one as in online learning. After each new observation, if we retrain the linear classifier using all the previous data, we need to re-do the minimization, which is computationally inefficient. In this paper, we instead extend Bayesian linear classification to perform recursive updates with each observation.

A Bayesian approach to linear classification models requires a prior distribution for the weight parameters $\boldsymbol{w}$, and the ability to compute the conditional posterior $p(\boldsymbol{w}|\mathcal{D})$ given the observation. Specifically, suppose we begin with an arbitrary prior $p(\boldsymbol{w})$ and apply Bayes' theorem to calculate the posterior: $p(\boldsymbol{w}|\mathcal{D}) = \frac{1}{Z}p(\mathcal{D}|\boldsymbol{w})p(\boldsymbol{w})$,

where the normalization constant Z is the unknown evidence. An l_2-regularized logistic regression can be interpreted as a Bayesian model with a Gaussian prior on the weights with standard deviation $1/\sqrt{\lambda}$.

Unfortunately, exact Bayesian inference for linear classifiers is intractable since the evaluation of the posterior distribution comprises a product of sigmoid functions; in addition, the integral in the normalization constant is intractable as well, for both the logistic function or probit function. We can either use analytic approximations to the posterior, or solutions based on Monte Carlo sampling, foregoing a closed-form expression for the posterior. In this paper, we consider different analytic approximations to the posterior to make the computation tractable.

3.3.1 Online Bayesian Probit Regression Based on Assumed Gaussian Density Filtering

Assumed density filtering (ADF) is a general online learning schema for computing approximate posteriors in statistical models (Boyen and Koller, 1998; Lauritzen, 1992; Maybeck, 1982; Sahami et al., 1998). In ADF, observations are processed one by one, updating the posterior which is then approximated and is used as the prior distribution for the next observation.

For a given Gaussian prior distribution on some latent parameter $\boldsymbol{\theta}$, $p(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ and a likelihood $t(\boldsymbol{\theta}) := p(\mathcal{D}|\boldsymbol{\theta})$, the posterior $p(\boldsymbol{\theta}|\mathcal{D})$ is generally non-Gaussian,

$$p(\boldsymbol{\theta}|\mathcal{D}) = \frac{t(\boldsymbol{\theta})p(\boldsymbol{\theta})}{\int t(\tilde{\boldsymbol{\theta}})p(\tilde{\boldsymbol{\theta}})\mathrm{d}\tilde{\boldsymbol{\theta}}}.$$

We find the best approximation by minimizing the Kullback-Leibler (KL) divergence between the true posterior $p(\boldsymbol{\theta}|\mathcal{D})$ and the Gaussian approximation. It is well known that when $q(x)$ is Gaussian, the distribution $q(x)$ that minimizes $\mathrm{KL}(p(x)||q(x))$ is the one whose first and second moments match that of $p(x)$. It can be shown that

the Gaussian approximation $\hat{q}(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta}|\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\Sigma}})$ found by moment matching is given as:

$$\hat{\boldsymbol{\mu}} = \boldsymbol{\mu} + \boldsymbol{\Sigma}\boldsymbol{g}, \quad \hat{\boldsymbol{\Sigma}} = \boldsymbol{\Sigma} - \boldsymbol{\Sigma}(\boldsymbol{g}\boldsymbol{g}^T - 2\boldsymbol{G})\boldsymbol{\Sigma}, \tag{3.3}$$

where the vector $\boldsymbol{g}$ and the matrix $\boldsymbol{G}$ are given by

$$\boldsymbol{g} = \frac{\partial \log\Big(Z(\tilde{\boldsymbol{\mu}}, \tilde{\boldsymbol{\Sigma}})\Big)}{\partial \tilde{\boldsymbol{\mu}}}|_{\tilde{\boldsymbol{\mu}}=\boldsymbol{\mu}, \tilde{\boldsymbol{\Sigma}}=\boldsymbol{\Sigma}}, \quad \boldsymbol{G} = \frac{\partial \log\Big(Z(\tilde{\boldsymbol{\mu}}, \tilde{\boldsymbol{\Sigma}})\Big)}{\partial \tilde{\boldsymbol{\Sigma}}}|_{\tilde{\boldsymbol{\mu}}=\boldsymbol{\mu}, \tilde{\boldsymbol{\Sigma}}=\boldsymbol{\Sigma}},$$

and the normalization function Z is defined by

$$Z(\boldsymbol{\mu}, \boldsymbol{\Sigma}) := \int t(\tilde{\boldsymbol{\theta}})\mathcal{N}(\tilde{\boldsymbol{\theta}}|\boldsymbol{\mu}, \boldsymbol{\Sigma})\mathrm{d}\tilde{\boldsymbol{\theta}}.$$

For the sake of analytic convenience, we only consider probit regression under assumed Gaussian density filtering. Specifically, the distribution of $\boldsymbol{w}$ after n observations is modeled as $p(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}|\boldsymbol{\mu}^n, \boldsymbol{\Sigma}^n)$. The likelihood function for the next available data $(\boldsymbol{x}, y)$ is $t(\boldsymbol{w}) := \Phi(y\boldsymbol{w}^t\boldsymbol{x})$. Thus we have,

$$p(\boldsymbol{w}|\boldsymbol{x}, y) \propto \Phi(y\boldsymbol{w}^t\boldsymbol{x})\mathcal{N}(\boldsymbol{w}|\boldsymbol{\mu}^n, \boldsymbol{\Sigma}^n).$$

Since the convolution of the normal CDF and a Gaussian distribution is another normal CDF, moment matching (5.1) results in an analytical solution to the Gaussian approximation:

$$\boldsymbol{\mu}^{n+1} = \boldsymbol{\mu}^n + \frac{y\boldsymbol{\Sigma}^n\boldsymbol{x}}{\sqrt{1 + \boldsymbol{x}^T\boldsymbol{\Sigma}^n\boldsymbol{x}}} v\left(\frac{y\boldsymbol{x}^T\boldsymbol{\mu}^n}{\sqrt{1 + \boldsymbol{x}^T\boldsymbol{\Sigma}^n\boldsymbol{x}}}\right), \tag{3.4}$$

$$\boldsymbol{\Sigma}^{n+1} = \boldsymbol{\Sigma}^n - \frac{(\boldsymbol{\Sigma}^n\boldsymbol{x})(\boldsymbol{\Sigma}^n\boldsymbol{x})^T}{1 + \boldsymbol{x}^T\boldsymbol{\Sigma}^n\boldsymbol{x}} w\left(\frac{y\boldsymbol{x}^T\boldsymbol{\mu}^n}{\sqrt{1 + \boldsymbol{x}^T\boldsymbol{\Sigma}^n\boldsymbol{x}}}\right), \tag{3.5}$$

where

$$v(z) := \frac{\mathcal{N}(z|0, 1)}{\Phi(z)} \text{ and } w(z) := v(z)\Big(v(z) + z\Big).$$

In this work, we focus on diagonal covariance matrices $\mathbf{\Sigma}^n$ with $(\sigma_i^n)^2$ as the diagonal element due to computational simplicity and its equivalence with l_2 regularization, resulting in the following update for the posterior parameters:

$$\mu_i^{n+1} = \mu_i^n + \frac{yx_i(\sigma_i^n)^2}{\tilde{\sigma}} v(\frac{y\boldsymbol{x}^T\boldsymbol{\mu}^n}{\tilde{\sigma}}), \tag{3.6}$$

$$(\sigma_i^{n+1})^2 = (\sigma_i^n)^2 - \frac{x_i^2(\sigma_i^n)^4}{\tilde{\sigma}^2} w(\frac{y\boldsymbol{x}^T\boldsymbol{\mu}^n}{\tilde{\sigma}}), \tag{3.7}$$

where $\tilde{\sigma}^2 := 1 + \sum_{j=1}^d (\sigma_j^n)^2 x_j^2$. See, for example, Graepel et al. (2010) and Chu et al. (2011b) for successful applications of this online probit regression model in prediction of click-through rates and stream-based active learning.

Due to the popularity of logistic regression and the computational limitations of ADF (on general link functions other than probit function), we develop an online Bayesian linear classification procedure for general link functions to recursively predict the response of each alternative in the next section.

3.4 Online Bayesian Linear Classification Based on Laplace Approximation

In this section, we consider the Laplace approximation to the posterior and develop an online Bayesian linear classification schema for general link functions.

3.4.1 Laplace Approximation

Laplace's method aims to find a gaussian approximation to a probability density defined over a set of continuous variables. It can be obtained by finding the mode of the posterior distribution and then fitting a Gaussian distribution centered at that mode (see Bishop et al., 2006, chap. 4.5). Specifically, define the logarithm of the

unnormalized posterior distribution as

$$\Psi(\boldsymbol{w}) = \log p(\mathcal{D}|\boldsymbol{w}) + \log p(\boldsymbol{w}).$$

Since the logarithm of a Gaussian distribution is a quadratic function, we consider a second-order Taylor expansion to Ψ around its MAP (maximum a posteriori) solution $\hat{\boldsymbol{w}} = \arg\max_{\boldsymbol{w}} \Psi(\boldsymbol{w})$:

$$\Psi(\boldsymbol{w}) \approx \Psi(\hat{\boldsymbol{w}}) - \frac{1}{2}(\boldsymbol{w} - \hat{\boldsymbol{w}})^T \boldsymbol{H} (\boldsymbol{w} - \hat{\boldsymbol{w}}), \tag{3.8}$$

where $\boldsymbol{H}$ is the Hessian of the negative log posterior evaluated at $\hat{\boldsymbol{w}}$:

$$\boldsymbol{H} = -\nabla^2 \Psi(\boldsymbol{w})|_{\boldsymbol{w}=\hat{\boldsymbol{w}}}.$$

By exponentiating both sides of Eq. (3.8), we can see that the Laplace approximation results in a normal approximation to the posterior

$$p(\boldsymbol{w}|\mathcal{D}) \approx \mathcal{N}(\boldsymbol{w}|\hat{\boldsymbol{w}}, \boldsymbol{H}^{-1}). \tag{3.9}$$

For multivariate Gaussian priors $p(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}|\boldsymbol{m}, \boldsymbol{\Sigma})$,

$$\Psi(\boldsymbol{w}|\boldsymbol{m}, \boldsymbol{\Sigma}) = -\frac{1}{2}(\boldsymbol{w} - \boldsymbol{m})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{w} - \boldsymbol{m}) + \sum_{i=1}^{n} \log(\sigma(y_i \cdot \boldsymbol{w}^T \boldsymbol{x}_i)), \tag{3.10}$$

and the Hessian $\boldsymbol{H}$ evaluated at $\hat{\boldsymbol{w}}$ is given for both logistic and normal CDF link functions as:

$$\boldsymbol{H} = \boldsymbol{\Sigma}^{-1} - \sum_{i=1}^{n} \hat{t}_i \boldsymbol{x}_i \boldsymbol{x}_i^T, \tag{3.11}$$

where $\hat{t}_i := \frac{\partial^2 \log p(y_i|\boldsymbol{x}_i, \boldsymbol{w})}{\partial f_i^2}|_{f_i = \hat{\boldsymbol{w}}^T \boldsymbol{x}_i}$ and $f_i = \boldsymbol{w}^T \boldsymbol{x}_i$.

3.4.2 Online Bayesian Linear Classification Based on Laplace Approximation

Starting from a Gaussian prior $\mathcal{N}(\boldsymbol{w}|\boldsymbol{m}^0, \boldsymbol{\Sigma}^0)$, after the first n observations, the Laplace approximated posterior distribution is $p(\boldsymbol{w}|\mathcal{D}^n) \approx \mathcal{N}(\boldsymbol{w}|\boldsymbol{m}^n, \boldsymbol{\Sigma}^n)$ according to (3.9). We formally define the state space $\mathcal{S}$ to be the cross-product of $\mathbb{R}^d$ and the space of positive semidefinite matrices. At each time n, our state of knowledge is thus $S^n = (\boldsymbol{m}^n, \boldsymbol{\Sigma}^n)$. Observations come one by one due to the sequential nature of our problem setting. After each new observation, if we retrain the Bayesian classifier using all the previous data, we need to calculate the MAP solution of (3.10) with $\mathcal{D} = \mathcal{D}^n$ to update from S^n to S^{n+1}. It is computationally inefficient even with a diagonal covariance matrix. It is better to extend the Bayesian linear classifier to handle recursive updates with each observation.

Here, we propose a fast and stable online algorithm for model updates with independent normal priors (with $\boldsymbol{\Sigma} = \lambda^{-1}\boldsymbol{I}$, where $\boldsymbol{I}$ is the identity matrix), which is equivalent to l_2 regularization and which also offers greater computational efficiency. At each time step n, the Laplace approximated posterior $\mathcal{N}(\boldsymbol{w}|\boldsymbol{m}^n, \boldsymbol{\Sigma}^n)$ serves as a prior to update the model when the next observation is made. In this recursive way of model updating, previously measured data need not be stored or used for retraining the model. By setting the batch size $n = 1$ in Eq. (3.10) and (3.11), we have the sequential Bayesian linear model for classification as in Algorithm 2, where $\hat{t} := \frac{\partial^2 \log(\sigma(yf))}{\partial f^2}|_{f=\hat{\boldsymbol{w}}^T\boldsymbol{x}}$.

It is generally assumed that $\log \sigma(\cdot)$ is concave to ensure a unique solution of Eq. (3.10). It is satisfied by commonly used sigmoid functions for classification problems, including logistic function, probit function, complementary log-log function $\sigma(a) = 1 - \exp(-\exp(a))$ and log-log function $\exp(-\exp(-a))$.

We can tap a wide range of convex optimization algorithms including gradient search, conjugate gradient, and BFGS method (see Wright and Nocedal, 1999). But

Algorithm 2: Online Bayesian linear classification

input : Regularization parameter $\lambda > 0$

$m_j = 0$, $q_j = \lambda$. (Each weight w_j has an independent prior $\mathcal{N}(w_j|m_j, q_j^{-1})$)

for $t = 1$ *to* T **do**

  Get a new point $(\boldsymbol{x}, y)$.

  Find $\hat{\boldsymbol{w}}$ as the maximizer of (3.10): $-\frac{1}{2}\sum_{j=1}^{d} q_i(w_i - m_i)^2 + \log(\sigma(y_i \boldsymbol{w}^T \boldsymbol{x}_i))$.

  $m_j = \hat{w}_j$

  Update q_i according to (3.11): $q_j \leftarrow q_j - \hat{t}x_j^2$.

end

if we set $n = 1$ and $\Sigma = \lambda^{-1}\boldsymbol{I}$ in Eq. (3.10), a stable and efficient algorithm for solving

$$\arg\max_{\boldsymbol{w}} -\frac{1}{2}\sum_{j=1}^{d} q_i(w_i - m_i)^2 + \log(\sigma(y\boldsymbol{w}^T\boldsymbol{x})) \tag{3.12}$$

can be obtained as follows. First, taking derivatives with respect to w_i and setting $\frac{\partial F}{\partial w_i}$ to zero, we have

$$q_i(w_i - m_i) = \frac{y x_i \sigma'(y\boldsymbol{w}^T x)}{\sigma(y\boldsymbol{w}^T x)}, \qquad i = 1, 2, \ldots, d.$$

Defining p as

$$p := \frac{\sigma'(y\boldsymbol{w}^T x)}{\sigma(y\boldsymbol{w}^T x)},$$

we then have $w_i = m_i + yp\frac{x_i}{q_i}$. Plugging this back into the definition of p to eliminate w_i's, we get the equation for p:

$$p = \frac{\sigma'(p\sum_{i=1}^{d} x_i^2/q_i + y\boldsymbol{m}^T x)}{\sigma(p\sum_{i=1}^{d} x_i^2/q_i + y\boldsymbol{m}^T x)}.$$

Since $\log(\sigma(\cdot))$ is concave, by its derivative we know the function σ'/σ is monotonically decreasing, and thus the right hand side of the equation decreases as p goes from 0 to ∞. We notice that the right hand side is positive when $p = 0$ and the left hand side is larger than the right hand side when $p = \sigma'(y\boldsymbol{m}^T x)/\sigma(y\boldsymbol{m}^T x)$. Hence the equation has a unique solution in interval $[0, \sigma'(y\boldsymbol{m}^T x)/\sigma(y\boldsymbol{m}^T x)]$. A simple one

dimensional bisection method is sufficient to efficiently find the root p^* and thus the solution to the d-dimensional optimization problem (3.12).

We illustrate and validate this schema using logistic functions. For logistic function $\sigma(\boldsymbol{w}^T\boldsymbol{x}) = -\log(1+\exp(-y\boldsymbol{w}^T\boldsymbol{x}))$, by setting $\partial F/\partial w_i = 0$ for all i and then by denoting $(1+\exp(y\mathbf{w}^T\mathbf{x}))^{-1}$ as p, we have

$$q_i(w_i - m_i) = ypx_i, \qquad i = 1, 2, \ldots, d,$$

resulting in the following equation for p:

$$\frac{1}{p} = 1 + \exp\left(y\sum_{i=1}^{d}(m_i + yp\frac{x_i}{q_i})x_i\right) = 1 + \exp(y\mathbf{m}^T\mathbf{x})\exp\left(y^2p\sum_{i=1}^{d}\frac{x_i^2}{q_i}\right).$$

It is easy to see that the left hand side decreases from infinity to 1 and the right hand side increases from 1 when p goes from 0 to 1, therefore the solution exists and is unique in $[0, 1]$.

3.5 Knowledge Gradient Policy for Bayesian Linear Classification Belief Model

First recall from Section 2.1.2 the definition of the knowledge gradient (KG) for ranking and selection problems, where each of the alternative can be measured sequentially to estimates its unknown underlying expected performance μ_x. The goal is to adaptively allocate alternatives to measure so as to find an implementation decision that has the largest mean after the budget is exhausted. In a Bayesian setting, the performance of each alternative is represented by a (non-parametric) lookup table model of Gaussian distribution. Specifically, by imposing a Gaussian prior $\mathcal{N}(\boldsymbol{\mu}|\boldsymbol{\theta}^0, \boldsymbol{\Sigma}^0)$, the posterior after the first n observations is denoted by $\mathcal{N}(\boldsymbol{\mu}|\boldsymbol{\theta}^n, \boldsymbol{\Sigma}^n)$. At the nth iteration, the knowledge gradient policy chooses its $(n+1)$th measurement to maximize

the single-period expected increase in value (Frazier et al., 2008):

$$\nu_x^{\mathrm{KG},n} = \mathbb{E}[\max_{x'} \theta_{x'}^{n+1} - \max_{x'} \theta_{x'}^{n} | x^n = x, S^n].$$

The knowledge gradient can be extended to online problems where we need to maximize cumulative rewards (Ryzhov et al., 2012),

$$\nu_x^{\mathrm{OLKG},n} = \theta_x^n + \tau \nu_x^{\mathrm{KG},n},$$

where τ reflects a planning horizon.

Yet there is no KG variant designed for binary classification with parametric models, primarily because of the computational intractability of dealing with nonlinear belief models. In what follows, we first formulate our learning problem as a Markov decision process and then extend the KG policy for stochastic binary outcomes where, for example, each choice (say, a medical decision) influences the success or failure of a medical outcome.

3.5.1 Markov Decision Process Formulation

Our learning problem is a dynamic program that can be formulated as a Markov decision process. Define the state space $\mathcal{S}$ as the space of all possible predictive distributions for $\boldsymbol{w}$. By Bayes' Theorem, the transition function T: $\mathcal{S} \times \mathcal{X} \times \{-1, 1\}$ is:

$$T\bigg(q(\boldsymbol{w}), \boldsymbol{x}, y\bigg) \propto q(\boldsymbol{w})\sigma(y\boldsymbol{w}^T\boldsymbol{x}). \tag{3.13}$$

If we start from a Gaussian prior $\mathcal{N}(\boldsymbol{w}|\boldsymbol{\mu}^0, \boldsymbol{\Sigma}^0)$, after the first n observed data, the approximated posterior distribution is $p(\boldsymbol{w}|\mathcal{D}^n) \approx \mathcal{N}(\boldsymbol{w}|\boldsymbol{\mu}^n, \boldsymbol{\Sigma}^n)$. The state space $\mathcal{S}$ is the cross-product of $\mathbb{R}^d$ and the space of positive semidefinite matrices. The transition function for updating the belief state depends on the belief model $\sigma(\cdot)$ and

the approximation strategy. For example, for different update equations in Algorithm 2 and (3.6)(3.7) under different approximation methods, the transition function can be defined as follows with degenerate state space $\mathcal{S} := \mathbb{R}^d \times [0, \infty)^d$:

DEFINITION 3.5.1. *The transition function based on online Bayesian classification with Laplace approximation T: $\mathcal{S} \times \mathcal{X} \times \{-1, 1\}$ is defined as*

$$T^L\Big((\boldsymbol{\mu}, \boldsymbol{\sigma}^2), \boldsymbol{x}, y\Big) = \Big(\arg\min_{\boldsymbol{w}} \Psi\big(\boldsymbol{w}|\boldsymbol{\mu}, \boldsymbol{\sigma}^{-2}\big), \boldsymbol{\sigma}^{-2} + \hat{t}(y) \cdot \boldsymbol{diag}(\boldsymbol{x}\boldsymbol{x}^T)\Big),$$

where $\hat{t}(y) := \frac{\partial^2 \log p(y|\boldsymbol{x}, \boldsymbol{w})}{\partial f^2}|_{f=\hat{\boldsymbol{w}}^T\boldsymbol{x}}$ for either logistic or probit functions, $\boldsymbol{diag}(\boldsymbol{x}\boldsymbol{x}^T)$ is a column vector containing the diagonal elements of $\boldsymbol{x}\boldsymbol{x}^T$ and $\boldsymbol{\sigma}^{-2}$ is understood as a column vector containing σ_i^{-2}, so that $S^{n+1} = T^L(S^n, \boldsymbol{x}, Y^{n+1})$. Y^{n+1} denotes the unobserved binary random variable at time n.

DEFINITION 3.5.2. *The transition function based on assumed density filtering T: $\mathcal{S} \times \mathcal{X} \times \{-1, 1\}$ is defined as*

$$T^{ADF}\Big((\boldsymbol{\mu}, \boldsymbol{\sigma}^2), \boldsymbol{x}, y\Big) = \Big(\boldsymbol{\mu} + \frac{y\boldsymbol{x}^T\boldsymbol{\sigma}^2}{\tilde{\sigma}} v(\frac{y\boldsymbol{x}^T\boldsymbol{\mu}}{\tilde{\sigma}}), \boldsymbol{\sigma}^2 - \frac{(\boldsymbol{x}^2)^T\boldsymbol{\sigma}^4}{\tilde{\sigma}^2} w(\frac{y\boldsymbol{x}^T\boldsymbol{\mu}}{\tilde{\sigma}})\Big),$$

where $\tilde{\sigma} = \sqrt{1 + (\boldsymbol{x}^2)^T\boldsymbol{\sigma}^2}$, $v(z) := \frac{\mathcal{N}(z|0,1)}{\Phi(z)}$, $w(z) := v(z)\Big(v(z) + z\Big)$ and $\boldsymbol{x}^2$ is understood as the column vector containing x_i^2, so that $S^{n+1} = T^{ADF}(S^n, \boldsymbol{x}, Y^{n+1})$.

In a dynamic program, the value function is defined as the value of the optimal policy given a particular state S^n at time n, and may also be determined recursively through Bellman's equation. In the case of stochastic binary feedback, the terminal value function $V^N : \mathcal{S} \mapsto \mathbb{R}$ is given by (3.1) as

$$V^N(s) = \max_{\boldsymbol{x}} p(y = +1|\boldsymbol{x}, s), \forall s \in \mathcal{S}.$$

The dynamic programming principle tells us that the value function at any other time $n = 1, \ldots, N$, V^n, is given recursively by

$$V^n(s) = \max_{\boldsymbol{x}} \mathbb{E}[V^{n+1}(T(s, \boldsymbol{x}, Y^{n+1}))|\boldsymbol{x}, s], \forall s \in \mathcal{S}.$$

Since the curse of dimensionality on the state space $\mathcal{S}$ makes direct computation of the value function intractable, computationally efficient approximate policies need to be considered. A computationally attractive policy for ranking and selection problems is known as the knowledge gradient (KG) (Frazier et al., 2008), which will be extended to handle Bayesian classification models in the next section.

3.5.2 Knowledge Gradient for Binary Responses

The knowledge gradient of measuring an alternative $\boldsymbol{x}$ can be defined as follows:

Definition 3.5.3. *The knowledge gradient of measuring an alternative $\boldsymbol{x}$ while in state s is*

$$\nu_{\boldsymbol{x}}^{KG}(s) := \mathbb{E}\Big[V^N\Big(T(s, \boldsymbol{x}, Y)\Big) - V^N(s)|\boldsymbol{x}, s\Big]. \tag{3.14}$$

$V^N(s)$ is deterministic given s and is independent of alternatives $\boldsymbol{x}$. Since the label for alternative $\boldsymbol{x}$ is not known at the time of selection, the expectation is computed conditional on the current belief state $s = (\boldsymbol{\mu}, \boldsymbol{\Sigma})$. Specifically, given a state $s = (\boldsymbol{\mu}, \boldsymbol{\Sigma})$, the outcome y of an alternative $\boldsymbol{x}$ is a random variable that follows from a Bernoulli distribution with a predictive distribution

$$p(y = +1|\boldsymbol{x}, s) = \int p(y = +1|\boldsymbol{x}, \boldsymbol{w}) p(\boldsymbol{w}|s) \mathrm{d}\boldsymbol{w} = \int \sigma(\boldsymbol{w}^T \boldsymbol{x}) p(\boldsymbol{w}|s) \mathrm{d}\boldsymbol{w}. \tag{3.15}$$

We can calculate the expected value in the next state as

$$\begin{aligned}\mathbb{E}[V^N(T(s,\boldsymbol{x},y))] &= p(y=+1|\boldsymbol{x},s)V^N\Big(T(s,\boldsymbol{x},+1)\Big)+p(y=-1|\boldsymbol{x},s)V^N\Big(T(s,\boldsymbol{x},-1)\Big)\\ &= p(y=+1|\boldsymbol{x},s)\cdot\max_{\boldsymbol{x}'}p\Big(y=+1|\boldsymbol{x}',T(s,\boldsymbol{x},+1)\Big)\\ &\quad +p(y=-1|\boldsymbol{x},s)\cdot\max_{\boldsymbol{x}'}p\Big(y=+1|\boldsymbol{x}',T(s,\boldsymbol{x},-1)\Big).\end{aligned}$$

The knowledge gradient policy suggests at each time n selecting the alternative that maximizes $\nu_{\boldsymbol{x}}^{\mathrm{KG},n}(s^n)$ where ties are broken randomly. Because of the errors incurred by approximation and numerical calculation, the tie should be understood as within ϵ-accuracy. The knowledge gradient policy can work with any choice of link function $\sigma(\cdot)$ and approximation procedures by adjusting the transition function $T(s,x,\cdot)$ accordingly. That is, T^{L} or T^{ADF}.

The predictive distribution $\int\sigma(\boldsymbol{w}^T\boldsymbol{x})p(\boldsymbol{w}|s)\mathrm{d}\boldsymbol{w}$ is obtained by marginalizing with respect to the distribution specified by current belief state $p(\boldsymbol{w}|s)=\mathcal{N}(\boldsymbol{w}|\boldsymbol{\mu},\boldsymbol{\Sigma})$. Denoting $a=\boldsymbol{w}^T\boldsymbol{x}$ and $\delta(\cdot)$ as the Dirac delta function, we have $\sigma(\boldsymbol{w}^T\boldsymbol{x})=\int\delta(a-\boldsymbol{w}^T\boldsymbol{x})\sigma(a)\mathrm{d}a$. Hence

$$\int\sigma(\boldsymbol{w}^T\boldsymbol{x})p(\boldsymbol{w}|s)\mathrm{d}\boldsymbol{w}=\int\sigma(a)p(a)\mathrm{d}a,$$

where $p(a)=\int\delta(a-\boldsymbol{w}^T\boldsymbol{x})p(\boldsymbol{w}|s)\mathrm{d}\boldsymbol{w}$. Since the delta function imposes a linear constraint on $\boldsymbol{w}$ and $p(\boldsymbol{w}|s)=\mathcal{N}(\boldsymbol{w}|\boldsymbol{\mu},\boldsymbol{\sigma}^2)$ is Gaussian, the marginal distribution $p(a)$ is also Gaussian. We can evaluate $p(a)$ by calculating the mean and variance of this distribution (Bishop et al., 2006). We have

$$\begin{aligned}\mu_a &= \mathbb{E}[a]=\int p(a)a\ \mathrm{d}a=\int p(\boldsymbol{w}|s)\boldsymbol{w}^T\boldsymbol{x}\ \mathrm{d}\boldsymbol{w}=\boldsymbol{\mu}^T\boldsymbol{x},\\ \sigma_a^2 &= \mathrm{Var}[a]=\int p(\boldsymbol{w}|s)\big((\boldsymbol{w}^T\boldsymbol{x})^2-(\boldsymbol{\mu}^T\boldsymbol{x})^2\big)\ \mathrm{d}\boldsymbol{w}=\sum_{j=1}^{d}\sigma_j^2x_j^2.\end{aligned}$$

Thus $\int \sigma(\boldsymbol{w}^T\boldsymbol{x})p(\boldsymbol{w}|s)\mathrm{d}\boldsymbol{w} = \int \sigma(a)p(a)\mathrm{d}a = \int \sigma(a)\mathcal{N}(a|\mu_a, \sigma_a^2)\mathrm{d}a$.

For probit function $\sigma(a) = \Phi(a)$, the convolution of a Gaussian and a normal CDF can be evaluated analytically. Thus for probit regression, the predictive distribution can be solved exactly as $p(y = +1|\boldsymbol{x}, s) = \Phi(\frac{\mu_a}{\sqrt{1+\sigma_a^2}})$. Yet, the convolution of a Gaussian with a logistic sigmoid function cannot be evaluated analytically. We apply the approximation $\sigma(a) \approx \Phi(\alpha a)$ with $\alpha = \pi/8$ (see Barber and Bishop 1998; Spiegelhalter and Lauritzen 1990), leading to the following approximation for the convolution of a logistic sigmoid with a Gaussian

$$p(y = +1|\boldsymbol{x}, s) = \int \sigma(\boldsymbol{w}^T\boldsymbol{x})p(\boldsymbol{w}|s)\mathrm{d}\boldsymbol{w} \approx \sigma(\kappa(\sigma_a^2)\mu_a),$$

where $\kappa(\sigma^2) = (1 + \pi\sigma^2/8)^{-1/2}$.

Because of the one-step look ahead, the KG calculation can also benefit from the online recursive update of the belief either from ADF or from online Bayesian linear classification based on Laplace approximation. We summarize the decision rules of the knowledge gradient policy at each iteration under different sigmoid functions and different approximation methods in Algorithm 3, 4 and 5, respectively.

Algorithm 3: Knowledge Gradient Policy for Logistic Model based on Laplace approximation

input : m_j, q_j (Each weight w_j has an independent prior $\mathcal{N}(w_j|m_j, q_j^{-1})$)
for $\boldsymbol{x}$ *in* $\mathcal{X}$ **do**
  Let $\Psi(\boldsymbol{w}, y) = -\frac{1}{2}\sum_{j=1}^d q_j(w_j - m_j)^2 - \log(1 + \exp(-y\boldsymbol{w}^T\boldsymbol{x}))$
  Use bisection method to find
  $\hat{\boldsymbol{w}}_+ = \arg\max_{\boldsymbol{w}} \Psi(\boldsymbol{w}, +1)$, $\hat{\boldsymbol{w}}_- = \arg\max_{\boldsymbol{w}} \Psi(\boldsymbol{w}, -1)$ $\mu = \boldsymbol{m}^T\boldsymbol{x}$,
  $\sigma^2 = \sum_{j=1}^d q_j^{-1}x_j^2$
  Let $\sigma(a) = \big(1 + \exp(-a)\big)^{-1}$
  $\mu_+(\boldsymbol{x}') := \hat{\boldsymbol{w}}_+^T\boldsymbol{x}'$, $\mu_-(\boldsymbol{x}') := \hat{\boldsymbol{w}}_-^T\boldsymbol{x}'$
  $\sigma_\pm^2(\boldsymbol{x}') := \sum_{j=1}^d \Big(q_j + \sigma(\hat{\boldsymbol{w}}_\pm^T\boldsymbol{x})(1 - \sigma(\hat{\boldsymbol{w}}_\pm^T\boldsymbol{x}))x_j^2\Big)^{-1}(x_j')^2$
  $\tilde{\nu}_{\boldsymbol{x}} = \sigma(\kappa(\sigma^2)\mu) \cdot \max_{\boldsymbol{x}'} \sigma\big(\kappa(\sigma_+^2(\boldsymbol{x}'))\mu_+(\boldsymbol{x}')\big) + \sigma(-\kappa(\sigma^2)\mu) \cdot$
  $\max_{\boldsymbol{x}'} \sigma\big(\kappa(\sigma_-^2(\boldsymbol{x}'))\mu_-(\boldsymbol{x}')\big)$
end
$\boldsymbol{x}^{\mathrm{KG}} \in \arg\max_{\boldsymbol{x}} \tilde{\nu}_x$

Algorithm 4: Knowledge Gradient Policy for Probit Model based on Laplace approximation

input : m_j, q_j (Each weight w_j has an independent prior $\mathcal{N}(w_j|m_j, q_j^{-1})$)
for $\boldsymbol{x}$ *in* $\mathcal{X}$ **do**
  Let $\Psi(\boldsymbol{w}, y) = -\frac{1}{2}\sum_{j=1}^{d} q_i(w_i - m_i)^2 + \log(\Phi(y\boldsymbol{w}^T\boldsymbol{x}))$
  Use bisection method to find
  $\hat{\boldsymbol{w}}_+ = \arg\max_{\boldsymbol{w}} \Psi(\boldsymbol{w}, +1), \hat{\boldsymbol{w}}_- = \arg\max_{\boldsymbol{w}} \Psi(\boldsymbol{w}, -1)$
  $\mu = \boldsymbol{m}^T\boldsymbol{x}$, $\sigma^2 = \sum_{j=1}^{d} q_j^{-1}x_j^2$
  $\mu_+(\boldsymbol{x}') := \hat{\boldsymbol{w}}_+^T\boldsymbol{x}'$, $\mu_-(\boldsymbol{x}) := \hat{\boldsymbol{w}}_-^T\boldsymbol{x}'$
  $\sigma_\pm^2(\boldsymbol{x}') := \sum_{j=1}^{d}\left(q_j + (\frac{\mathcal{N}(\hat{\boldsymbol{w}}_\pm^T\boldsymbol{x}|0,1)^2}{\Phi(\hat{\boldsymbol{w}}_\pm^T\boldsymbol{x})^2} + \frac{\hat{\boldsymbol{w}}_\pm^T\boldsymbol{x}\mathcal{N}(\hat{\boldsymbol{w}}_\pm^T\boldsymbol{x}|0,1)}{\Phi(\hat{\boldsymbol{w}}_\pm^T\boldsymbol{x})})x_j^2\right)^{-1}(x_j')^2$
  $\tilde{\nu}_{\boldsymbol{x}} = \Phi(\frac{\mu}{\sqrt{1+\sigma^2}}) \cdot \max_{\boldsymbol{x}'} \Phi(\frac{\mu_+(\boldsymbol{x}')}{\sqrt{1+\sigma_+^2(\boldsymbol{x}')}}) + \Phi(-\frac{\mu}{\sqrt{1+\sigma^2}}) \cdot \max_{\boldsymbol{x}'} \Phi(\frac{\mu_-(\boldsymbol{x}')}{\sqrt{1+\sigma_-^2(\boldsymbol{x}')}})$
end
$\boldsymbol{x}^{\mathrm{KG}} \in \arg\max_{\boldsymbol{x}} \tilde{\nu}_x$

Algorithm 5: Knowledge Gradient Policy for Probit Model based on assumed density filtering

input : m_j, q_j (Each weight w_j has an independent prior $\mathcal{N}(w_j|m_j, q_j^{-1})$)
$\sigma_j^2 = 1/q_j$
for $\boldsymbol{x}$ *in* $\mathcal{X}$ **do**
  Define $v(z) := \frac{\mathcal{N}(z|0,1)}{\Phi(z)}$ and $w(z) := v(z)\Big(v(z) + z\Big)$
  $\mu = \boldsymbol{m}^T\boldsymbol{x}$, $\sigma^2 = \sum_{j=1}^{d} \sigma_j^2 x_j^2$
  $m_{+j} = m_j + \frac{x_j\sigma_j^2}{\sqrt{1+\sigma^2}} v(\frac{\boldsymbol{m}^T\boldsymbol{x}}{\sqrt{1+\sigma^2}})$, $m_{-j} = m_j - \frac{x_j\sigma_j^2}{\sqrt{1+\sigma^2}} v(-\frac{\boldsymbol{m}^T\boldsymbol{x}}{\sqrt{1+\sigma^2}})$
  $\sigma_{+j}^2 = \sigma_j^2 - \frac{x_j^2\sigma_j^4}{\sqrt{1+\sigma^2}^2} w(\frac{\boldsymbol{m}^T\boldsymbol{x}}{\sqrt{1+\sigma^2}})$, $\sigma_{-j}^2 = \sigma_j^2 - \frac{x_j^2\sigma_j^4}{\sqrt{1+\sigma^2}^2} w(-\frac{\boldsymbol{m}^T\boldsymbol{x}}{\sqrt{1+\sigma^2}})$
  $\mu_+(\boldsymbol{x}') := \boldsymbol{m}_+^T\boldsymbol{x}'$, $\mu_-(\boldsymbol{x}) := \boldsymbol{m}_-^T\boldsymbol{x}'$
  $\sigma_+^2(\boldsymbol{x}') := \sum_{j=1}^{d} \sigma_{+j}^2 (x_j')^2$, $\sigma_-^2(\boldsymbol{x}') := \sum_{j=1}^{d} \sigma_{-j}^2 (x_j')^2$
  $\tilde{\nu}_{\boldsymbol{x}} = \Phi(\frac{\mu}{\sqrt{1+\sigma^2}}) \cdot \max_{\boldsymbol{x}'} \Phi(\frac{\mu_+(\boldsymbol{x}')}{\sqrt{1+\sigma_+^2(\boldsymbol{x}')}}) + \Phi(-\frac{\mu}{\sqrt{1+\sigma^2}}) \cdot \max_{\boldsymbol{x}'} \Phi(\frac{\mu_-(\boldsymbol{x}')}{\sqrt{1+\sigma_-^2(\boldsymbol{x}')}})$
end
$\boldsymbol{x}^{\mathrm{KG}} \in \arg\max_{\boldsymbol{x}} \tilde{\nu}_x$

3.5.3 Behavior and Asymptotic Optimality

In this section, we study theoretically the behavior of the KG policy, especially in the limit as the number of measurements N grows large. For the purposes of the theoretical analysis, we do not approximate the predictive posterior distribution. We

use logistic function as the sigmoid link function throughout this section. Yet the theoretical results can be generalized to other link functions. We begin by showing the positive value of information (benefits of measurement).

PROPOSITION 3.5.1 (Benefits of measurement). *The knowledge gradient of measuring any alternative $\boldsymbol{x}$ while in any state $s \in \mathcal{S}$ is nonnegative, $\nu_{\boldsymbol{x}}^{KG}(s) \geq 0$. The state space $\mathcal{S}$ is the space of all possible predictive distributions for $\boldsymbol{w}$.*

The next lemma shows that the asymptotic probability of success of each alternative is well defined.

LEMMA 3.5.4. *For any alternative $\boldsymbol{x}$, $p_{\boldsymbol{x}}^n$ converges almost surely to a random variable $p_{\boldsymbol{x}}^\infty = \mathbb{E}\big[\sigma(\boldsymbol{w}^T\boldsymbol{x})|\mathcal{F}^\infty\big]$, where $p_{\boldsymbol{x}}^n$ is short hand notation for $p(y = +1|\boldsymbol{x}, \mathcal{F}^n) = \mathbb{E}\big[\sigma(\boldsymbol{w}^T\boldsymbol{x})|\mathcal{F}^n\big]$.*

Proof. ProofSince $|\sigma(\boldsymbol{w}^T\boldsymbol{x})| \leq 1$, the definition of $p_{\boldsymbol{x}}^n$ implies that $p_{\boldsymbol{x}}^n$ is a bounded martingale and hence converges. □

The rest of this section shows that this limiting probability of success of each alternative is one in which the posterior is consistent and thus the KG is asymptotically optimal. We also show that as the number of measurements grows large, the maximum likelihood estimator $\boldsymbol{w}_{\text{MLE}}$ based on the alternatives measured by the KG policy is consistent and asymptotically normal. The next proposition states that if we have measured an alternative infinitely many times, there is no benefit to measuring it one more time. This is a key step for establishing the consistency of the KG policy and the MLE. The proof is similar to that by Frazier et al. (2009) with additional mathematical steps for Bernoulli distributed random variables. See Appendix A.4.2 for details.

PROPOSITION 3.5.2. *If the policy π measures alternative $\boldsymbol{x}$ infinitely often almost surely, then the value of information of that alternative $\nu_{\boldsymbol{x}}(\mathcal{F}^\infty) = 0$ almost surely under policy π.*

Without loss of generality, we assume $\|\boldsymbol{x}\|_2 \leq 1$ and that the $d \times d$ matrix P formed by $(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_d)$ is invertible. The next theorem states the strong consistency and asymptotic normality of the maximum likelihood estimator $\boldsymbol{w}^n_{\text{MLE}}$ (e.g. with $\lambda = 0$) based on KG's sequential selection of alternatives by verification of the following regularity conditions:

(C_1) The exogenous variables are uniformly bounded.

(C_2) Let λ_{1n} and λ_{dn} be respectively the smallest and the largest eigenvalue of the Fisher information of the first n observations $\boldsymbol{F}_n(\boldsymbol{w}^*)$. There exists C such that $\lambda_{dn}/\lambda_{1n} < C$, $\forall n$.

(C_3) The smallest eigenvalue of the Fisher information is divergent, $\lambda_{1n} \to +\infty$.

THEOREM 3.5.5. *The sequence of $\boldsymbol{w}^n_{MLE}$ based on KG's sequential selection of alternatives converges almost surely to the true value $\boldsymbol{w}^*$ and is asymptotically normal:*

$$\boldsymbol{F}_n^{\frac{1}{2}}(\boldsymbol{w}^n_{MLE} - \boldsymbol{w}^*) \xrightarrow{d} \mathcal{N}(0, \boldsymbol{I}).$$

Proof. Proof We first prove that for any alternative $\boldsymbol{x}$, it will be measured infinitely many times almost surely. We will prove it by contradiction. If this is not the case, then there exists a time T such that for any $n > T$,

$$\mu^{\text{KG},n}_{\boldsymbol{x}} < \max_{\boldsymbol{x} \in \mathcal{X}} \mu^{\text{KG},n}_{\boldsymbol{x}} - \epsilon. \tag{3.16}$$

This is because otherwise the difference between the KG value of $\boldsymbol{x}$ and the maximum KG value will be smaller than ϵ for infinitely many times, and thus the probability of not measuring $\boldsymbol{x}$ after T will be 0.

In addition, since the KG value is always non-negative, we have $\max_{\boldsymbol{x} \in \mathcal{X}} \mu^{\text{KG},n}_{\boldsymbol{x}} > \epsilon$ for each $n > T$. Notice that $\mathcal{X}$ is a finite set, then it follows that there exists an

alternative $\boldsymbol{x}'$ such that the following happens infinitely many times:

$$\mu_{\boldsymbol{x}'}^{\mathrm{KG},n} = \max_{\boldsymbol{x}\in\mathcal{X}} \mu_{\boldsymbol{x}}^{\mathrm{KG},n}. \tag{3.17}$$

Therefore, $\boldsymbol{x}'$ is measured infinitely many times almost surely. However, we have proved in Proposition 3.5.2 that $\mu_{\boldsymbol{x}'}^{\mathrm{KG},n}$ goes to 0 zero as long as we measure $\boldsymbol{x}'$ infinitely many times, which contradicts (3.17). The contradiction shows that our original assumption that $\boldsymbol{x}$ only being measured finite times is incorrect. As a consequence, we have prove that, almost surely, $\boldsymbol{x}$ will be measured infinitely many times.

Since our proof is for arbitrary $\boldsymbol{x}$, we actually proved that every alternative will be measured infinitely many times, which immediately leads to $\max_{\boldsymbol{x}\in\mathcal{X}} \mu_{\boldsymbol{x}}^{\mathrm{KG},n} \to 0$. Therefore the algorithm will eventually pick the alternative uniformly at random. Hence it satisfies the conditions (C_2) and (C_3), leading to the strong consistency and asymptotic normality (Gourieroux and Monfort, 1981; Fahrmeir and Kaufmann, 1985; Haberman and Haberman, 1974; Cox and Hinkley, 1979).

In particular, we can prove that the smallest eigenvalue of the Fisher's information goes to infinity in a simple way. Without loss of generality, assume that the alternatives $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_d$ are in general linear position, which means that the $d \times d$ matrix $P := (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_d)$ is invertible. We use $\mathcal{X}'$ to denote the set of these d alternatives and denote P^{-1} by Q. Then $Q\boldsymbol{x}_t\boldsymbol{x}_t^T Q^T$ is a matrix whose every element equals 0 except that its k-th diagonal element equals one.

We use $\boldsymbol{F}_n$ to denote the Fisher's information at time n. Then from the fact that the matrix of type $\boldsymbol{x}\boldsymbol{x}^T$ is positive semi-definite, it is straightforward to see that

$$\boldsymbol{F}_n(\boldsymbol{w}) \geq \min_{\boldsymbol{x}\in\mathcal{X}} (1 - \sigma(\boldsymbol{x}^T\boldsymbol{w}))\sigma(\boldsymbol{x}^T\boldsymbol{w}) \sum_{1\leq t\leq n, \boldsymbol{x}_t\in\mathcal{X}'} \boldsymbol{x}_t\boldsymbol{x}_t^T,$$

where the constant $\min_{\boldsymbol{x}\in\mathcal{X}}(1-\sigma(\boldsymbol{x}^T\boldsymbol{w}))\sigma(\boldsymbol{x}^T\boldsymbol{w}) > 0$ since $\mathcal{X}$ is a finite set. Now define matrix R_n as

$$R_n := \sum_{1\le t\le n, \boldsymbol{x}_t\in\mathcal{X}'} Q\boldsymbol{x}_t\boldsymbol{x}_t^T Q^T,$$

which is a diagonal matrix whose i-th element equals the times that $\boldsymbol{x}_i$ is estimated. Since $\boldsymbol{x}_k$ is measured infinitely many times for $1 \le k \le d$, then each diagonal element of R_n goes to infinity. Now notice that $\sum_{1\le t\le n,\boldsymbol{x}_t\in\mathcal{X}'} \boldsymbol{x}_t\boldsymbol{x}_t^T = Q^{-1}R_n(Q^{-1})^T$ is congruent to R_n and Q^{-1} is a constant matrix, then it follows that the smallest eigenvalue of $\sum_{1\le t\le n,\boldsymbol{x}_t\in\mathcal{X}'} \boldsymbol{x}_t\boldsymbol{x}_t^T$, and hence the smallest eigenvalue of $\boldsymbol{F}_n$, goes to infinity. □

After establishing the consistency and asymptotic normality for $\boldsymbol{w}^n_{\text{MLE}}$, for any $\lambda > 0$ as the inverse of the variance of the prior Gaussian distribution, the asymptotic bias of the estimator $\boldsymbol{w}^n_\lambda$ is as follows (Le Cessie and Van Houwelingen, 1992):

$$\boldsymbol{E}[\boldsymbol{w}^n_\lambda - \boldsymbol{w}^*] = -2\lambda\{\boldsymbol{F}_n(\boldsymbol{w}^*) + 2\lambda\boldsymbol{I}\}^{-1}\boldsymbol{w}^*,$$

and the asymptotic variance of $\boldsymbol{w}^n_\lambda$ is $\{\boldsymbol{F}_n(\boldsymbol{w}^*) + 2\lambda\boldsymbol{I}\}^{-1}\boldsymbol{F}_n(\boldsymbol{w}^*)\{\boldsymbol{F}_n(\boldsymbol{w}^*) + 2\lambda\boldsymbol{I}\}^{-1}$.

Finally, we show in the next theorem that given the opportunity to measure infinitely often, for any given neighborhood of $\boldsymbol{w}^*$, the probability that the posterior distribution lies in this neighborhood goes to 1 and KG will discover which one is the true best alternative. The detailed proof can be found in Appendix A.4.3.

THEOREM 3.5.6 (Asymptotic optimality). *For any true parameter value $\boldsymbol{w}^* \in \mathbb{R}^d$ and any normal prior distribution with positive definite covariance matrix, under the knowledge gradient policy, the posterior is consistent and the KG policy is asymptotically optimal:* $\arg\max_{\boldsymbol{x}} \mathbb{E}^{KG}[\sigma(\boldsymbol{w}^T\boldsymbol{x})|\mathcal{F}^\infty] = \arg\max_{\boldsymbol{x}} \sigma((\boldsymbol{w}^*)^T\boldsymbol{x})$.

UCI dataset	Number of alternatives	Number of attributes
sonar	208	60
glass identification	214	10
blood transfusion	748	5
survival	306	3
breast cancer	198	34
planning relax	182	13
climate	540	18

Table 3.1: Summary of datasets.

3.6 Experimental Results

In this section, we evaluate the proposed method in offline learning settings where we are not punished for errors incurred during training and only concern with the final recommendation after the offline training phases.

We experiment with both synthetic datasets and the UCI machine learning repository (Lichman, 2013) which includes classification problems drawn from settings including sonar, glass identification, blood transfusion, survival, breast cancer (wpbc), planning relax and climate model failure. We first analyze the behavior of the KG policy and then compare it to state-of-the-art learning algorithms. On synthetic datasets, we randomly generate a set of M d-dimensional alternatives $\boldsymbol{x}$ from $[-3, 3]$. At each run, the stochastic binary labels are simulated using a $d+1$-dimensional weight vector $\boldsymbol{w}^*$ which is sampled from the prior distribution $w_i^* \sim \mathcal{N}(0, \lambda)$. The $+1$ label for each alternative $\boldsymbol{x}$ is generated with probability $\sigma(w_0^* + \sum_{j=1}^{d} w_j^* x_j)$. For each UCI dataset, we use all the data points as the set of alternatives with their original attributes. We then simulate their labels using a weight vector $\boldsymbol{w}^*$. This weight vector could have been chosen arbitrarily, but it was in fact a perturbed version of the weight vector trained through logistic regression on the original dataset. A summary of the datasets is listed in Table 3.1.

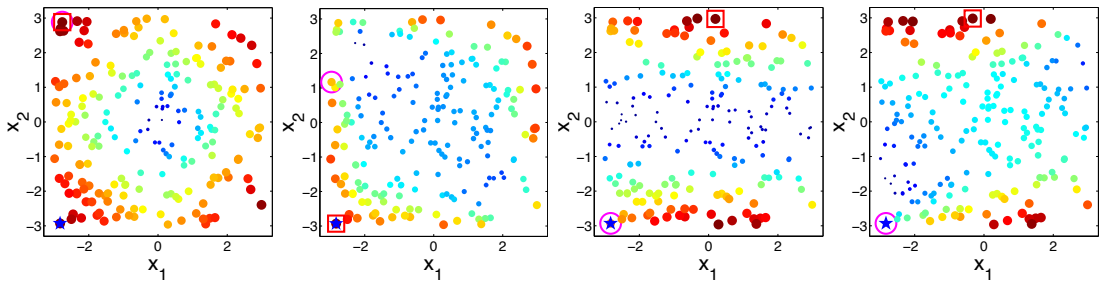


Figure 3.1: The scatter plots illustrate the KG values at 1-4 iterations from left to right with both the color and the size reflecting the magnitude. The star, the red square and pink circle indicate the true best alternative, the alternative to be selected and the implementation decision, respectively.

3.6.1 Behavior of the KG Policy

To better understand the behavior of the KG policy, Fig. 3.1 shows the snapshot of the KG policy at each iteration on a 2-dimensional synthetic dataset and a 3-dimensional synthetic dataset in one run. Fig. 3.1 shows the snapshot on a 2-dimensional dataset with 200 alternatives. The scatter plots show the KG values with both the color and the size of the point reflecting the KG value of the corresponding alternative. The star denotes the true alternative with the largest response. The red square is the alternative with the largest KG value. The pink circle is the implementation decision that maximizes the response under current estimation of $\boldsymbol{w}^*$ if the budget is exhausted after that iteration.

It can be seen from the figure that the KG policy finds the true best alternative after only three measurements, reaching out to different alternatives to improve its estimates. We can infer from Fig. 3.1 that the KG policy tends to choose alternatives near the boundary of the region. This criterion is natural since in order to find the true maximum, we need to get enough information about $\boldsymbol{w}^*$ and estimate well the probability of points near the true maximum which appears near the boundary. On the other hand, in a logistic model with labeling noise, a data $\boldsymbol{x}$ with small $\boldsymbol{x}^T\boldsymbol{x}$ inherently brings little information as pointed out by Zhang and Oles (2000). For an extreme example, when $\boldsymbol{x} = \mathbf{0}$ the label is always completely random for any $\boldsymbol{w}$ since

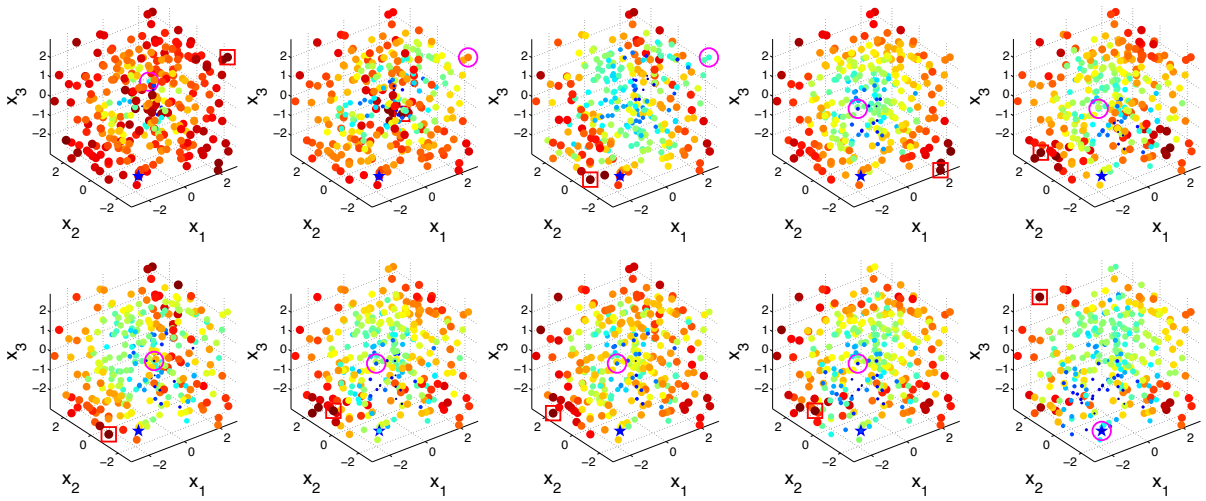
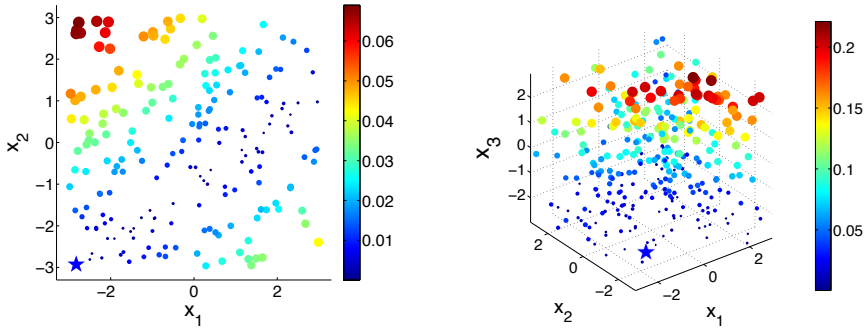


Figure 3.2: Snapshots on a 3-dimensional dataset. The scatter plots illustrate the KG values at 1-10 iterations from left to right, top to bottom. The star, the red square and pink circle indicate the best alternative, the alternative to be selected and the implementation decision.

$p(y = +1|\boldsymbol{w}, \mathbf{0}) \equiv 0.5$. This is an issue when perfect classification is not achievable. So it is essential to label a data with larger $\boldsymbol{x}^T\boldsymbol{x}$ that has the most potential to enhance its confidence non-randomly.

Fig. 3.2 illustrates the snapshots of the KG policy on a 3-dimensional synthetic dataset with 300 alternatives. It can be seen that the KG policy finds the true best alternative after only 10 measurements. This set of plots also verifies our statement that the KG policy tends to choose data points near the boundary of the region.



(a) 2-dimensional dataset (b) 3-dimensional dataset

Figure 3.3: Absolute error between the predictive probability of +1 under current estimate and the true probability.

Also depicted in Fig. 4.2 is the absolute class distribution error of each alternative, which is the absolute difference between the predictive probability of class +1 under current estimate and the true probability after 6 iterations for the 2-dimensional dataset and 10 iterations for the 3-dimensional dataset. We see that the probability at the true maximum is well approximated, while moderate error in the estimate is located away from this region of interest.

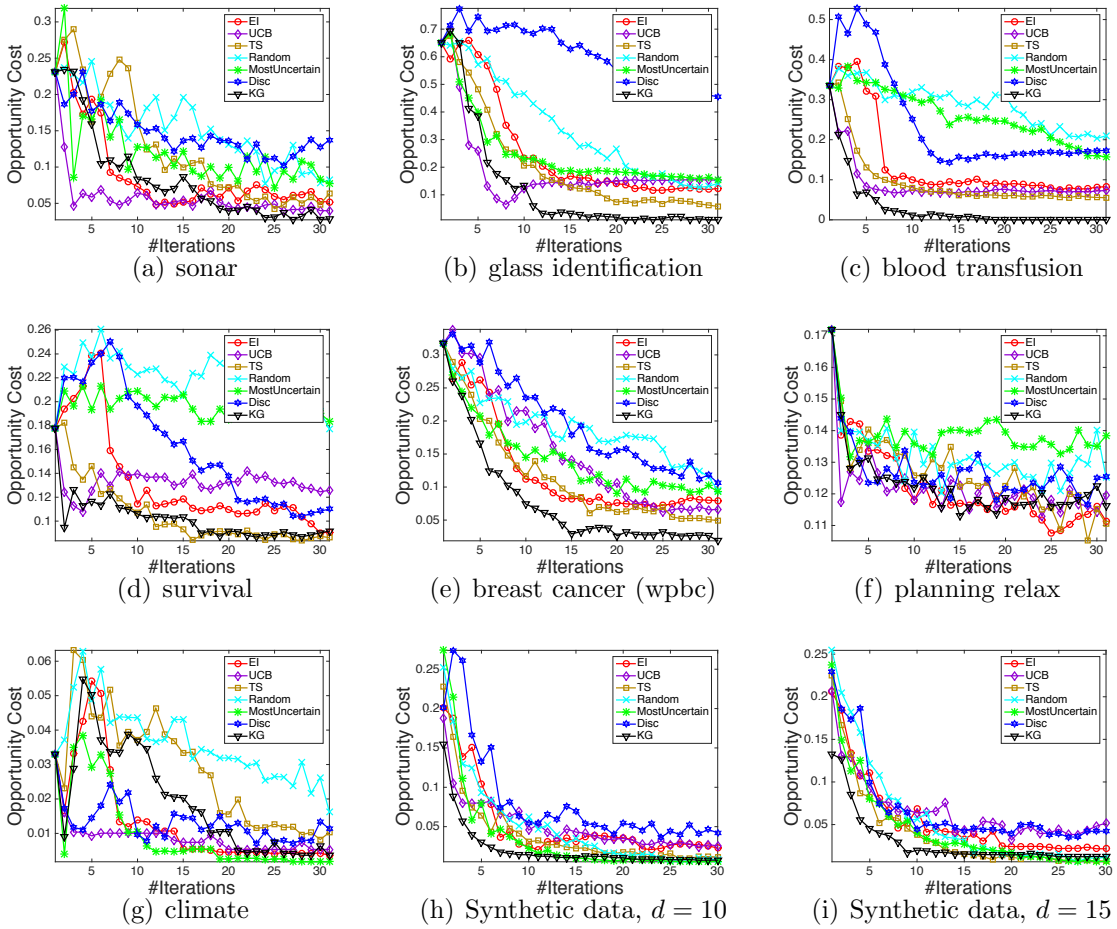
3.6.2 Comparison with Other Policies

Recall that our goal is to maximize the expected response of the implementation decision. We define the Opportunity Cost (OC) metric as the expected response of the implementation decision $\boldsymbol{x}^{N+1} := \arg\max_{\boldsymbol{x}} p(y = +1|\boldsymbol{x}, \boldsymbol{w}^N)$ compared to the true maximal response under weight $\boldsymbol{w}^*$:

$$\text{OC} := \max_{\boldsymbol{x}\in\mathcal{X}} p(y = +1|\boldsymbol{x}, \boldsymbol{w}^*) - p(y = +1|\boldsymbol{x}^{N+1}, \boldsymbol{w}^*).$$

Note that the opportunity cost is always non-negative and the smaller the better. To make a fair comparison, on each run, all the time-N labels of all the alternatives are randomly pre-generated according to the weight vector $\boldsymbol{w}^*$ and shared across all the competing policies. We allow each algorithm to sequentially measure N = 30 alternatives.

We compare with the following state-of-the-art active learning and Bayesian optimization policies that are compatible with logistic regression: Random sampling (Random), a myopic method that selects the most uncertain instance each step (MostUncertain), discriminative batch-mode active learning (Disc) (Guo and Schuurmans, 2008) with batch size equal to 1, expected improvement (EI) (Tesch et al., 2013) with an initial fit of 5 examples and Thompson sampling (TS) (Chapelle and Li, 2011). Besides, as upper confidence bounds (UCB) methods are often considered



(a) sonar (b) glass identification (c) blood transfusion

(d) survival (e) breast cancer (wpbc) (f) planning relax

(g) climate (h) Synthetic data, $d = 10$ (i) Synthetic data, $d = 15$

Figure 3.4: Opportunity cost on UCI and synthetic datasets.

in bandit and optimization problems, we compare against UCB on the latent function $\boldsymbol{w}^T\boldsymbol{x}$ (UCB) (Li et al., 2010) with α tuned to be 1. All the state transitions are based on the online Bayesian logistic regression framework developed in Section 3.4, while different policies provides different rules for measurement decisions at each iteration. The experimental results are shown in Figure 3.4. In all the figures, the x-axis denotes the number of measured alternatives and the y-axis represents the averaged opportunity cost averaged over 100 runs.

It is demonstrated in Figure 3.4 that KG outperforms the other policies in most cases, especially in early iterations, without requiring a tuning parameter. As an unbiased selection procedure, random sampling is at least a consistent algorithm.

Yet it is not suitable for expensive experiments where one needs to learn the most within a small experimental budget. MostUncertain and Disc perform quite well on some datasets while badly on others. A possible explanation is that the goal of active learning is to learn a classifier which accurately predicts the labels of new examples so their criteria are not directly related to maximize the probability of success aside from the intent to learn the prediction. After enough iterations when active learning methods presumably have the ability to achieve a good estimator of $\boldsymbol{w}^*$, their performance will be enhanced. Thompson sampling works in general quite well as reported in other literature (Chapelle and Li, 2011). Yet, KG has a better performance especially during the early iterations. In the case when an experiment is expensive and only a small budget is allowed, the KG policy, which is designed specifically to maximize the response, is preferred.

We also note that KG works better than EI in most cases, especially in Figure 3.4(b), 3.4(c) and 3.4(e). Although both KG and EI work with the expected value of information, when EI decides which alternative to measure, it is based on the expected improvement over current predictive posterior distribution while ignoring the potential change of the posterior distribution resulting from the next (stochastic) outcome y. In comparison, KG considers an additional level of expectation over the random (since at the time of decision, we have not yet observed outcome) binary output y.

Finally, KG, EI and Thompson sampling outperform the naive use of UCB policies on the latent function $\boldsymbol{w}^T\boldsymbol{x}$ due to the errors in the variance introduced by the nonlinear transformation. At each time step, the posterior of $\log \frac{p}{1-p}$ is approximated as a Gaussian distribution. An upper confidence bound on $\log \frac{p}{1-p}$ does not translate to one on p with binary outcomes. In the meantime, KG, EI and Thompson sampling make decisions in the underlying binary outcome probability space and find the right balance of exploration and exploitation.

3.7 Conclusion

Motivated by real world applications, we consider binary classification problems where we have to sequentially run expensive experiments, forcing us to learn the most from each experiment. With a small budget of measurements, the goal is to learn the classification model as quickly as possible to identify the alternative with the highest probability of success. Due to the sequential nature of this problem, we develop a fast online Bayesian linear classifier for general response functions to achieve recursive updates. We propose a knowledge gradient policy using Bayesian linear classification belief models, for which we use different analytical approximation methods to overcome computational challenges. We further extend the knowledge gradient to the contextual bandit settings. We show that the maximum likelihood estimator based on the adaptively sampled points by the KG policy is consistent and asymptotically normal. We show furthermore that the knowledge gradient policy is asymptotic optimal. We demonstrate its efficiency through a series of experiments.

Chapter 4

Bayesian Contextual Bandits for Personalized Health Care

According to the Centers for Medicare and Medicaid Services (CMS), U.S. health care expenditures grew 5.3 percent in 2014, reaching \$3 trillion or $\$9,523$ per person. As a share of the nation's Gross Domestic Product, health spending accounted for 17.5 percent. Rising health care costs have become a major concern for hospital chains (Bodenheimer, 2005; Ginsburg, 2008; Wennberg et al., 2008; Lehnert et al., 2011), which increasingly have to deliver the best care possible within a given budget.

Common practice is to assign patients to medical professionals (general practitioners, specialists, nurse practitioners) on a first-available basis. This ignores special expertise with particular medical conditions, as well as the past performance of the physician or facility. At the same time, physicians may face choices in terms of how to treat a condition, which tends to be guided in part by the past experiences of each physician. Thus, we have choices of physician (or type of physician), care facility, and specific treatments. The best choices depend on a combination of the characteristics of the patient, the physician, the facility, and the treatment plan. We address the problem of how to make the best decisions in the presence of imperfect (and some-

times highly imperfect) understanding of the relationship between patient attributes, medical decisions and medical outcomes.

We are particularly interested in total knee replacement (Callahan et al., 1994; Buckwalter and Lohmander, 1994), a common operation for people with osteoarthritis of the knee which affects more than 27 million people in the U.S. according to the Arthritis Foundation. More than 600,000 knee replacements are performed each year in the U.S., which also leads to extensive post-operative rehabilitation which varies widely from one patient to the next. In 2013, the cost is more than $7 billion for hospitalization alone. In order to promote a health care system that provides better care and spends health care dollars more wisely, in 2016, the United States Department of Health and Human Services (HHS) proposed the Comprehensive Care for Joint Replacement (CJR) where the hospital may be required to repay Medicare for a portion of the cost of a knee replacement episode if the cost and quality fall outside of specified ranges (HHS, 2015).

We focus on the case of a single decision and take an online view, continuously using accumulating data to modify aspects of the treatment regime as new patients come in. We adopt a Bayesian approach where a prior on the response is used to help balance making the best decision now, while maximizing the value of information from each episode to make better decisions in the future.

In this chapter, we consider a binary feedback (success/failure) model where if the post-operative cost is below some threshold, the episode of care (spanning initial diagnosis and testing, inpatient treatment and outpatient care) is said to be successful; otherwise it is treated as failure. The aim is to decide the most appropriate physicians and caregivers for each individual patient and maximize the success rate over time. This is an example of the broader area of personalized medicine, which formalizes clinical decision making as a function that maps individual patient information (including measures of disease stage severity, medical history, clinical diagnosis, genomic infor-

mation and environmental information) to a recommended treatment. Our work is part of a growing trend toward personalized care where medical decisions are tuned to the characteristics of each patient. This approach, however, introduces considerable uncertainty in the identification of the best treatments since there is very little data describing patients with the same (or even similar) attributes. As a result, there is considerable uncertainty in models of the relationship between medical decisions (and patient attributes) with treatment outcomes. This motivates our work that addresses the problem of balancing between making what appears to be the best decisions, and learning to make better decisions in the future.

We formalize personalized medicine as a Bayesian contextual bandit problem. We encounter two challenges. First, there are very few patients with the same characteristics, which means that it is unlikely that an individual physician sees a sufficient number of eligible patients to produce statistically reliable performance measurements on medical outcomes. We overcome this situation using a parametric belief model that allows us to learn relationships across a wide range of patients and health providers. This is different from the earlier multi-armed bandit formulations in clinical trials (Lai, 1987; Lai and Liao, 2012) which ignore the attributes of individual patients, effectively assuming that patients are homogeneous. In this paper, we use context information to explicitly model the heterogeneity in need and responses. The context-specific best action, which requires finding a function (that is, a policy), is dramatically more difficult computationally than finding the best single action required by the context-free case. Second, testing a treatment decision is expensive. This puts us in the setting of optimal learning where we need to learn the best treatment as fast as possible. This represents a distinctly different learning environment than what has traditionally been considered using popular policies such as upper confidence bounding (UCB) which have proven effective in settings with high sampling rates such as learning ad-clicks or the doubly robust estimation which is trained on historical data

(Dudík et al., 2014; Zhang et al., 2012). We therefore adopt a Bayesian approach and the knowledge gradient policy which takes advantage of domain knowledge to produce rapid learning, and which maximizes success rates for both the current and future patients.

In what follows, we describe a methodology for quickly learning a contextual, binary response model for personalized healthcare. We introduce a two-step Bellman's equation for Bayesian contextual bandits and develop an optimal learning policy to guide the treatment assignment by maximizing the expected value of information. We use modularity detection and LASSO to help overcome the intrinsic sparsity of health datasets, and show that the method significantly increases the results of knee replacement surgeries through careful selection of physicians.

4.1 Literature Review

There is a variety of new methods to aid in the search for the optimal treatment regime, where a single decision or a series of sequential decisions may be involved, including sequential multiple assignment randomized trials (SMART) (Almirall et al., 2012; Murphy, 2005), doubly robust estimators (Murphy et al., 2001; Zhang et al., 2012; Brinkley, 2014), Q-learning (Laber et al., 2015; Qian and Murphy, 2011), adaptive strategies (Lavori and Dawson, 2000) and other dynamic treatment regimes studied at length by Robins and colleagues (Murphy, 2003; Murphy and Collins, 2007; Robins, 2004, 1993). Much of the work is trained on past observational data and the treatment regime is not updated with new patient responses. Yet when working with historical data, the result is less objective and can be biased substantially due to the differences, for example, in patient populations and in medical facilities or the evolution of the diseases. This is known as offline settings where we are not punished

for errors incurred during training and where we are only concerned with the final treatment regime after the offline training phases.

Our work also bears similarity to adaptive designs (Ashby, 2006; Berry et al., 2010; Jack Lee and Chu, 2012; Chow, 2014; Bather, 1985; Yin et al., 2012). Decisions are made adaptively throughout the running of the trial. The sequence 1) make an observation, 2) update your knowledge and 3) decide what information to collect next, is the basis of the scientific method, following the guidelines given by Thall and Simon (1994). Both our setting and adaptive designs face the same exploration/exploitation dilemma: (1) treat current patients as effectively as possible while (2) learning to improve treatments in the future. Making what we think is currently the best decision may not be the best given the uncertainty in our model, forcing us to recognize that we have to learn to make better decisions in the future.

4.2 Problem Definition

Personalized medicine takes into consideration the heterogeneity in needs and responses of different patients. It aims to effectively integrate and analyze healthcare data to provide the right intervention to the right patient at the right time. With the popularization of the electronic health record (EHR), defined as a systematic collection of electronic health information about individual patients or populations, it is easier for us to make data-driven decisions that facilitates optimized patient-centered and automated healthcare delivery.

4.2.1 Personalized Medicine

A patient can be characterized by a set of unique characteristics such as measures of disease stage severity, medical history, clinical diagnosis, genomic information, and environmental information, with a health complaint that requires medical intervention.

We use this information as a basis for decisions about medical treatment, including choice of physician, tests, drugs, surgery, and rehabilitation/follow up.

The challenge lies in the sequential nature of the problem which can happen both across different patients and within a patient. For example, patients come to the hospital one by one. For each patient visit, a medical decision is made from which we learn a response and update our model, allowing us to make better decisions for the next upcoming patient. However, we need to use this knowledge in the context of new patients, each described by their own attributes, leading to different medical decisions. The second sequential nature involves development of dynamic treatment regimes. In treating each individual, there are usually multiple hospital visits which correspond to different stages of the health conditions of each patient. In this case, different decisions may be needed for different stages depending on patient attributes at that time.

In this work, for simplicity, we only consider the case where there is a single medical decision to be made for each patient. After the medical decision is made, at the end of a treatment episode, we observe a dichotomous health outcome (e.g. whether cost is below a Medicare-specified threshold, whether the treatment is effective), which is then used to update our understanding of relationships between medical decisions and health outcomes for a patient with a specific set of characteristics. This model can be easily extended to account for dynamic treatment regimes by gathering patient features of each hospital visit. It is worth mentioning that not only the patients are the beneficiaries, it potentially benefits all the components of a healthcare system, including the healthcare provider, individual patient, policy maker and management.

4.2.2 The Contextual Model

For the nth patient x^n, the learner or the decision maker is presented with a context vector $\boldsymbol{\phi}^X(x^n) = \left(\phi_f^X(x^n)\right)_{f\in\mathcal{F}^X}$ and a set of actions $a \in \mathcal{A}$ (doctors, treat-

ments, rehabilitation). Here a learner can be a doctor or a hospital administrator, the context vector is the characteristics of the nth patient x^n, and each action can be different treatments, different doctors or rehabilitation facilities. Each action a is associated with a feature vector $\boldsymbol{\phi}^A(a)$. For example, a drug can be represented by a binary vector with each item representing the presence of a specific molecular compound. A treatment decision can also be handled with indicator variables (such as $\mathbb{I}(a, \text{“doctor”}) = 1$ if an action a refers to assigning a particular doctor). For any context x and action a, there is an unknown function $p : \mathcal{X} \times \mathcal{A} \mapsto [0, 1]$ which represents the underlying binomial probability of success of an experiment. After choosing an action a, a response of patient $y^{n+1} \in \{-1, +1\}/$ {failure, success} for the action a is revealed, but the rewards of other actions are not observable. The “success” or “failure” depends stochastically on x and a. This setting is also known as *contextual bandits*.

A policy π or a treatment regime is a function mapping from any context information x to an action a. We denote the “patient horizon” as N which is the number of present and future patients who will be treated with one of the treatment in $\mathcal{A}$. It is worth noting that N need not be known beforehand and may depend on the pattern of an emerging disease, the performance of current treatments, the emergence of new treatments, and may be infinite for recurring conditions such as knee replacements. Chapter 3 has proposed optimal learning methods for sequential decision making with stochastic binary feedbacks for context-free Bayesian optimization. In terms of personalized medicine scenarios, it is equivalent to assuming that all patients are homogeneous. In this paper, context information explicitly models the heterogeneity in needs and responses and the context-specific best action is a more demanding benchmark than the best action identification in the context-free case.

We adopt probabilistic modeling for the unknown probability of success. Specifically, for any continuous-valued latent function $f(x, a)$, the corresponding probability

density over class probability functions can be obtained by mapping a model with range of $(-\infty, \infty)$ to an output in $[0, 1]$ using sigmoid response functions σ. For example, given a linear regression model $f = \boldsymbol{w}^T \boldsymbol{\phi}$, the predicted probability of class $+1$ is $\sigma(\boldsymbol{w}^T \boldsymbol{\phi})$. The link function $\sigma(b)$ is often chosen as the logistic function $\sigma(b) = \frac{1}{1+\exp(-b)}$ or probit function $\sigma(b) = \Phi(b) = \int_{-\infty}^{b} \mathcal{N}(s|0, 1^2)\mathrm{d}s$.

We encode assumptions about the smoothness of the latent function by modeling it as a sample from a Gaussian process defined over the joint context-action space. Specifically, this paper, we assume that the latent function $f : \mathcal{X} \times \mathcal{A} \mapsto \mathbb{R}$ is a sample from a known Gaussian process (GP) distribution. A Gaussian process is a statistical distribution of dependent random variables such that every finite collection of those random variables has a multivariate normal distribution. We use $\mathcal{Z} = \mathcal{X} \times \mathcal{A}$ to refer to the set of all context-action pairs. A GP is fully specified with its mean function $\boldsymbol{\mu}$ and covariance function (kernel) $\boldsymbol{\Sigma}$ which can often be naturally decomposed into the kernels on actions and contexts. For example, in our knee replacement experiments (described in Section 4.5), we choose the kernel function for actions $\Sigma_A = \boldsymbol{I}$ as the $|\mathcal{A}| \times |\mathcal{A}|$ identity matrix, and Σ_X as the linear kernel on the patient features.

We use K^n to denote the "state of knowledge" which captures the uncertainty in our system at time n. In the case of Gaussian processes for classification, K^n represents the posterior distribution over the latent function $\mathcal{N}(\boldsymbol{\mu}^n, \boldsymbol{\Sigma}^n)$. We then define $S^n = (K^n, x^n)$ to be the state of the system at time n. Each of the past observations are made of triplets (x^n, a^n, y^{n+1}), assuming labels y are generated independently. Let $\mathcal{D}^n = \{(x^i, a^i, y^{i+1})\}_{i=0}^{n-1}$ denote the previous measured data set for any $n = 0, \ldots, N-1$. Note that the notation here is slightly different from the (passive) PAC learning model where the data are i.i.d. and are denoted as $\{(x_i, y_i)\}$. Yet in our (adaptive) sequential decision setting, a decision a^n depends on the state S^n, while Y^{n+1} is a random variable that has not been observed at time n. This notation with superscript indexing time stamp is standard, for example, in control

theory, stochastic optimization and optimal learning. A history of the process can be represented using

$$h^n = (K^0, x^0, S^0, a^0, Y^1, K^1, x^1, S^1, a^1, Y^2, ..., a^{n-1}, Y^n, K^n, x^n, S^n).$$

The goal is to find a policy that selects actions such that the cumulative reward is as large as possible over time, or equivalently, treatment on patients is as effective as possible:

$$\max_{\pi} \mathbb{E}\Big[\sum_{n=0}^{N-1} Y^{n+1}\Big(S^n, A^{\pi}(S^n)\Big)|S^0\Big], \tag{4.1}$$

where Y denotes the random variable of the patient response, A^{π} denotes the treatment recommended by the dynamic treatment regime π.

4.3 Gaussian Process Classification

Since we do not observe values of the latent function f_z directly, the inference step for our GP conditioning on the previous observations $\mathcal{D}$ involves computing the following posterior distribution at any points $z = (x, a)$:

$$p(\boldsymbol{f}|\mathcal{D}) = \frac{1}{Z} p(\mathcal{D}|\boldsymbol{f}) p(\boldsymbol{f}),$$

where $p(\boldsymbol{f})$ is the prior distribution in our GP and the normalization constant Z is the unknown evidence.

Unfortunately, exact Bayesian inference for GP classifiers are intractable since the evaluation of the posterior distribution comprises a factorial likelihood, which is a product of sigmoid (non-Gaussian) functions; in addition, the integral in the normalization constant is intractable as well. We can either use analytic approximations to the posterior, or solutions based on Monte Carlo sampling, foregoing a closed-

form expression for the posterior. Different approximation methods are discussed by Nickisch and Rasmussen (2008). In this work, for algorithmic simplicity, we adopt a linear kernel on the latent function f. To be more specific, we use $\boldsymbol{\phi}(x, a)$ to denote the vector of features for each (context, action) pair. This can be generated by the context vector $\boldsymbol{\phi}^X(x)$ and the action feature vector $\boldsymbol{\phi}^A(a)$. Then the latent function is represented by a linear regression model $f = \boldsymbol{w}^T\boldsymbol{\phi}$.

Observations come one by one due to the sequential nature of our problem setting. After each new observation, retraining the GP classifier using all the previous data is computationally inefficient in terms of both time and space complexity. To this end, we use the online Bayesian linear classification algorithm proposed in Section 3.4.2 to handle recursive updates with each patient response with an independent normal prior distribution of the unknown parameter w is $\mathcal{N}(0, \lambda^{-1}\boldsymbol{I})$, which is known to be equivalent to l_2 regularization.

In our setting, the use of this convenient approximation of the posterior is twofold. It first serves as a prior on the weights to update the model when a new patient response becomes available. Second, it defines the belief states in the Bayesian policies, for example, the knowledge gradient policy and Thompson sampling that we introduce later. Starting from Gaussian priors $\mathcal{N}\big(w_j|m_j^0, (q_j^0)^{-1}\big)$ over w_j with mean m_j^0 and variance $(q_j^0)^{-1}$, after the first n patient responses, the Laplace-approximated posterior distribution is $p(w_j|\mathcal{D}^n) \approx \mathcal{N}\big(w_j|m^n, (q_j^n)^{-1}\big)$. At the nth time step, we find the MAP solution (3.8) to the posterior after the new information (x^n, a^n, y^{n+1}) by a one-dimensional bisection method:

$$\boldsymbol{m}^{n+1} = \arg\max_{\boldsymbol{w}} \frac{1}{2}\sum_{j=1}^{d} q_i^n(w_i - m_i^n)^2 + \log\Big(1 + \exp\big(-y^{n+1}\boldsymbol{w}^T\boldsymbol{\phi}^n\big)\Big), \tag{4.2}$$

where $\boldsymbol{\phi}^n$ is a compact notation for $\boldsymbol{\phi}(x^n, a^n)$. The inverse variance of each weight w_j is given by the curvature at the mode as:

$$q_j^{n+1} = q_j^n + \zeta^n(1-\zeta^n)(\boldsymbol{\phi}_j^n)^2, \tag{4.3}$$

where $\zeta^n = \left(1 + \exp\left((\boldsymbol{w}^{n+1})^T \boldsymbol{\phi}^n\right)\right)^{-1}$.

4.4 Knowledge Gradient Policy with Contextual Information

The knowledge gradient (KG) policy, first proposed for offline (context-free) ranking and selection problems, maximizes the value of information from a decision. In ranking and section problems, the performance of each alternative is represented by a (non-parametric) lookup table model. Although originally developed for offline learning (where we do not pay attention to successes while we are learning), it is easily adapted to online learning where we seek to maximize the cumulative number of successes. This is particularly well suited to personalized medicine where we want to learn as fast as possible from each patient response so as to provide better treatment on the upcoming patients. In comparison, policies like upper confidence bounding (UCB) are known to explore more than necessary, which is particularly undesirable in a medical setting.

While knowledge gradient policies have been designed for a variety of belief models, as of this writing it has not been adapted to the setting of contextual bandits. Chapter 3 has proposed optimal learning methods for sequential decision making with stochastic binary feedbacks for context-free Bayesian optimization, e.g. all patients are homogeneous. In this section, we use context information to explicitly model the heterogeneity in needs and responses of different patients. The context-specific best

action, which requires finding a function (that is, a policy), is dramatically more difficult computationally than finding the best single action required by the context-free case, comparable to the difference between static stochastic optimization and fully sequential dynamic programming.

4.4.1 Markov Decision Process Formulation

We define the knowledge state in our setting as $K^n = (\boldsymbol{m}^n, \boldsymbol{q}^n)$. For the context-free case, the state of the system S^n is identical to the state of knowledge K^n. In a dynamic program, the value function is defined as the value of the optimal policy given a particular state $S^n \in \mathcal{S}$ at time n, and may also be determined recursively through Bellman's equation. At time N, we should simply choose the alternative that looks the best given everything we have learned, because there are no longer any future decisions that might benefit from learning. Since the goal of personalized medicine is to maximize the cumulative success of treatment, the terminal value function $V^N : \mathcal{S} \mapsto \mathbb{R}$ is given by

$$V^N(s) = \max_a \mathbb{E}_Y \left[Y^{N+1}(s,a) \big| s \right] = \max_a p(y = +1|s,a), \forall s \in \mathcal{S}.$$

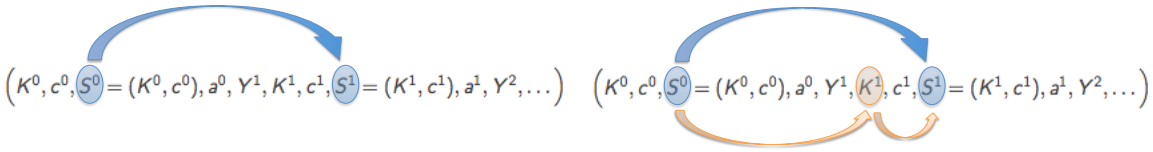
The value function at any other time $n = 0, 1, ..., N-1$ is given recursively by Bellman's equation for dynamic programming:

$$V^n(S^n) = \max_a \mathbb{E}_{x,Y} \left[Y^{n+1}(S^n, a) + V^{n+1}(S^{n+1}) | S^n \right], \tag{4.4}$$

where $S^{n+1} = (K^{n+1}, x^{n+1})$ is the random state of the system and the expectation is over the context x^{n+1} and the outcome Y^{n+1}, which means over the Bernoulli random outcome given the model, and the uncertainty in the model.

All the previous optimal learning literature considers transitions from S^n to S^{n+1} (as in Figure 4.1(a)) and develops algorithms based on the value at state S^n. Yet in

contextual bandits, the contexts are given as arbitrary, meaning that we do not explicitly model the distribution of the context information. Hence, we cannot compute the expectation over the context vector x in Eq. (4.4). In what follows, we break down the Bellman's equations into two steps, and provide the first formal mathematical model for the contextual bandits from a perspective of dynamic programming.



(a) One-step transition. (b) Two-step transition.

Figure 4.1: Illustrations of dynamic programming and Bellman's equation.

To be more specific, we write Bellman's equation in its standard form (Eq. (4.4)) by writing the sequence of states, actions and information using:

$$(S^0, a^0, Y^1, S^1, a^1, Y^2, \ldots, S^n, a^n, Y^{n+1}, \ldots)$$

When we include contextual information, this sequence would be written as

$$\Big(K^0, x^0, S^0 = (K^0, x^0), a^0, Y^1, K^1, x^1, S^1 = (K^1, x^1), a^1, Y^2, \ldots\Big)$$

Here, the knowledge state K^n is the *post-observation* state, while S^n is the *pre-decision* state, representing the state after a patient has arrived. Instead of relating $V^n(S^n)$ to $V^{n+1}(S^{n+1})$ as is classically done in Bellman's equation, we break the recursion into two steps: from S^n to K^{n+1}, and then from K^{n+1} to S^{n+1} (see Figure 4.1(b)), giving us the following two-step version of Bellman's equation.

$$\begin{aligned} V^n(S^n) &= \max_a \mathbb{E}\Big\{ Y^{n+1}(S^n, a) + V^{k,n+1}\Big(K^{n+1}\Big(Y^{n+1}(S^n, a)\Big)\Big) | S^n \Big\}, \\ V^{k,n+1}(K^{n+1}) &= \mathbb{E}_x\Big[V^{n+1}(S^{n+1}) | K^{n+1} \Big]. \end{aligned}$$

Bellman's equation works well for problems with small state and action spaces, and where the transition matrix can be easily computed. But in personalized health care, the context information x^n can only be observed (the distribution is unknown). The context information can be an arbitrary sequence which is fixed beforehand or stored in historical data, or it can be non-stochastically chosen by an adversary. The attractiveness of the post-observation state that the maximum and the expectation over context x are interchanged, giving us computational advantages to use a simulation-based approach without probabilistic modeling of the contextual information and by treating the contextual information as arbitrarily given by the oracle.

4.4.2 Knowledge Gradient Policy with Contextual Information

In order to approximately solve the Bellman's equations in the previous section, we for the first time propose to develop the knowledge gradient policy around the values in *post-observation* states K^n. The knowledge gradient $\nu_a^{\mathrm{KG},n}(S^n)$ of measuring an action a in state $S^n = (K^n, x^n)$ is defined as the single-step expected improvement in value if action a is taken.

$$\nu_a^{\mathrm{KG},n}(S^n) := \mathbb{E}_Y\left[V^{k,N}\Big(K^{n+1}\big(Y^{n+1}(S^n,a)\big)\Big) - V^{k,N}(K^n)|S^n\right], \tag{4.5}$$

where $K^{n+1}\big(Y^{n+1}(S^n,a)\big)$ is the next stochastic state of knowledge if we choose treatment $a^n = a$ right now, allowing us to observe the stochastic patient response Y^{n+1}. This allows us to update $\boldsymbol{m}^n$ and $\boldsymbol{q}^n$ according to Eq. (4.2) and Eq. (4.3), transitioning to the next state of knowledge K^{n+1}. The knowledge gradient policy then balances the treatment that appears to be the current best and the one that learns

the most by choosing an action a at time n as:

$$A^{\mathrm{KG},n}(S^n) = \arg\max_a p(y=+1|S^n,a) + \tau\nu_a^{\mathrm{KG},n}(S^n),$$

where τ reflects a planning horizon that captures the value of the information we have gained on future patients. If the time horizon N is known beforehand, τ can be chosen as $N-n$. Otherwise, we can also treat τ as a tunable parameter.

Given a knowledge state $k=(\boldsymbol{m},\boldsymbol{q})$, or equivalently $p(w_j|k)=\mathcal{N}(w_j|m_j,q_j^{-1})$, the predictive Bernoulli distribution of patient response y for a context x and a treatment a can be found by marginalization over $\boldsymbol{w}$,

$$p(y=+1|x,a,k) = \int p(y=+1|x,a,\boldsymbol{w})p(\boldsymbol{w}|k)\mathrm{d}\boldsymbol{w} = \int \sigma\big(\boldsymbol{w}^T\boldsymbol{\phi}(x,a)\big)p(\boldsymbol{w}|k)\mathrm{d}\boldsymbol{w} \quad (4.6)$$

For notational simplicity, we drop $\boldsymbol{\phi}$'s dependence on x and a. Denoting $b=\boldsymbol{w}^T\boldsymbol{\phi}$ and $\delta(\cdot)$ the Dirac delta function, we have $\sigma(\boldsymbol{w}^T\boldsymbol{\phi}) = \int \delta(b-\boldsymbol{w}^T\boldsymbol{\phi})\sigma(b)\mathrm{d}b$. Hence we have

$$\int \sigma(\boldsymbol{w}^T\boldsymbol{\phi})p(\boldsymbol{w}|k)\mathrm{d}\boldsymbol{w} = \int \sigma(b)p(b)\mathrm{d}b,$$

where $p(b)=\int \delta(b-\boldsymbol{w}^T\boldsymbol{\phi})p(\boldsymbol{w}|k)\mathrm{d}\boldsymbol{w}$. Since the delta function imposes a linear constraint on $\boldsymbol{w}$ and $p(\boldsymbol{w}|k)$ is Gaussian, the marginal distribution $p(b)$ is also Gaussian. We can evaluate $p(b)$ by calculating the mean and variance of this distribution. We have

$$\begin{aligned}
\mu_b &= \mathbb{E}[b] = \int p(b)b\ \mathrm{d}b = \int p(\boldsymbol{w}|k)\boldsymbol{w}^T\boldsymbol{\phi}\ \mathrm{d}\boldsymbol{w} = \boldsymbol{m}^T\boldsymbol{\phi}, \\
\sigma_b^2 &= \mathrm{Var}[b] = \int p(\boldsymbol{w}|k)\big((\boldsymbol{w}^T\boldsymbol{\phi})^2 - (\boldsymbol{m}^T\boldsymbol{\phi})^2\big)\ \mathrm{d}\boldsymbol{w} = \sum_{j=1}^d q_j^{-1}\phi_j^2.
\end{aligned}$$

Thus $\int \sigma\Big(\boldsymbol{w}^T\boldsymbol{\phi}(x,a)\Big)p(\boldsymbol{w}|k)\mathrm{d}\boldsymbol{w} = \int \sigma(b)p(b)\mathrm{d}b = \int \sigma(b)\mathcal{N}(b|\mu_b,\sigma_b^2)$. Since the convolution of a Gaussian $\mathcal{N}(b|\mu_b,\sigma_b^2)$ with a logistic function $\sigma(b)$ cannot be evaluated

analytically, we apply the approximation $\sigma(b) \approx \Phi(\alpha b)$ with $\alpha = \pi/8$. Denoting $\kappa(\sigma^2) = (1 + \pi\sigma^2/8)^{-1/2}$, we have

$$p(y = +1|x, a, k) = \int \sigma\Big(\boldsymbol{w}^T \boldsymbol{\phi}(x, a)\Big) p(\boldsymbol{w}|k) \mathrm{d}\boldsymbol{w} \approx \sigma(\kappa(\sigma_b^2)\mu_b).$$

We are now ready to compute the knowledge gradient value $\nu_a^{\mathrm{KG},n}(S^n)$ in Eq. (4.5). First, $V^{k,N}(K^n)$ is deterministic at time n. Since the patient response Y^{n+1} is not known at the time of selection, the expectation is computed over the Bernoulli distribution and the current belief model specified by K^n. Specifically, the expectation can be obtained by averaging the two possible responses $+1/-1$ as follows:

$$\begin{aligned}
& \mathbb{E}\Big[V^{k,N}\big(K^{n+1}(Y^{n+1}(S^n, a))\big)|S^n\Big] \\
= \;& p(y = +1|x, a, K^n) V^{k,N}\Big(K_{+1}^{n+1}(S^n, a)\Big) + p(y = -1|x, a, K^n) V^{k,N}\Big(K_{-1}^{n+1}(S^n, a)\Big) \\
= \;& p(y = +1|x, a, K^n) \cdot \max_{a'} p\Big(y = +1|x, a', K_{+1}^{n+1}(S^n, a)\Big) \\
& + p(y = -1|x, a, K^n) \cdot \max_{a'} p\Big(y = +1|x, a', K_{-1}^{n+1}(S^n, a)\Big),
\end{aligned}$$

where $K_Y^{n+1}(x, a)$ denotes the next belief state given outcome Y according to Eq. (4.2) and Eq. (4.3).

It can be seen that the Laplace approximation and the recursive update make the computation of the knowledge gradient tractable by analytically approximating the value in the next state $V^{n+1}(S^{n+1})$ and offering computational simplicity with Gaussian distributions.

4.5 Cost Reduction of Knee Replacement

We obtained knee replacement datasets from major hospital chains in New York and New Jersey. To make a fair statement of the costs, we selected 26,735 structured claim records that are obtained from the same health care provider. Each record

includes age, gender, episode identifier, episode start data, claim line paid amount, diagnosis codes, procedure codes, attributed physician identifiers and so on. Specifically, for example, the diagnosis is represented with ICD-9 codes which is the ninth revision of the International Classification of Diseases, a hierarchical terminology of diseases, signs, symptoms, abnormal findings, complaints, social circumstances, and external causes of injury maintained by the World Health Organization (WHO). In the procedure type record, each health care procedure is encoded by the Healthcare Common Procedure Coding System (hcpcs) based on the American Medical Association's Current Procedural Terminology (cpt). The episode ID, beneficiary ID and claim ID have been replaced with randomly assigned numbers for anonymization.

4.5.1 Data Description

All the patients in the knee replacement dataset have undergone knee replacement surgery which is represented by the hcpcs code 27447. After the knee replacement surgery, different patients have been involved in different lengths of rehabilitations and incurred a wild range of post-operative costs. In this work, we want to understand the effect of different physicians and/or facilities on the post-operative costs of each individual, and provide guidelines on how to more effectively assign different physician/facility to each patient based on patient attributes.

To this end, we are interested in the following data fields. The first one is the demographic information, including age and gender. The next category is the episode information, including the Episode_id and the Claim_date. We will later use these to group different claim records by episode ID and use time stamps to distinguish between pre-operative or post-operative clinical visits. We use Procedure_code and Diagnosis_code to later construct the patient feature vector. Attributed_phys_npi indicates the attributed physician for an episode and Rndrg_prvdr_npi_num represents the specific provider delivering a service. We are also interested in the amount paid for

Variable	Variable Definition
Episode_id	Episode identifier of each claim record
Age	Age
Gender	Gender
Claim_date	The date of the service
Attributed_phys_npi	Attributed physician for episode
Rndrg_prvdr_npi_num	The specific provider delivering a service, e.g. physical therapist
Claim_line_paid_amount	Amount paid for service
Procedure_code	Procedure hcpcs codes
Diagnosis_code	Diagnosis icd-9 codes

Table 4.1: Data description of the knee replacement.

service Claim_line_paid_amount in order to optimize on the post-operative costs. Table 4.1 summarizes the selected variables.

The key response variable in our analysis is the post-operative cost. In order to obtain the post-operative cost for each episode, we first group all the claim records based on their Episode_id and sort the claim records within each episode chronologically. This results in a total of 211 different episodes. We then locate the surgery with hcpcs code of 27447. By summing up the Claim_line_paid_amount of the records that happened after the surgery, we get the post-operative cost for each episode. We see a variety of different post-operative costs of 211 episodes, ranging from \$1787.69 to \$11571.44, as depicted in Figure 4.2. Based on the Comprehensive Care for Joint Replacement (CJR) project, the threshold is set to be \$5878 for this region. Since the hospital is required to repay Medicare for a portion of the cost of a knee replacement episode if the cost is higher than the specified threshold, from the perspective of hospital administration, we say that if the cost is smaller than the specified threshold, we think it as acceptable, otherwise we treat it as failure. Then the binary response (successes/failure) is the dependent variable in our prediction model.

For predictors, we first translate the claim records into matrix-based attributes. Specifically, we create a set of columns consisting of every diagnosis (icd-9 diagnosis code) and/or pre-operative procedure (hcpcs/cpt procedure code) that has appeared

in the dataset. For each patient, mark the value of that column as 1 if the patient has a diagnosis or uses the procedure in his/her historical medical records, otherwise mark as 0. Beyond this binary valued vector that encodes diagnoses and pre-operative procedures, we also include the demographic characteristics (sex and age) to constitute the feature vector of a patient.

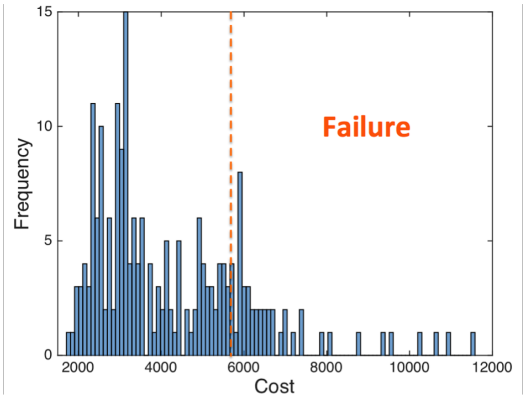


Figure 4.2: Post-operative cost distribution.

Not surprisingly, there are a huge number of features associated with each patient, e.g. 979 columns of possible diagnoses and 772 columns of pre-operative procedures. These features are typically sparse – compared to nearly 2000 features, the number of 1's for each patient ranges from 8 to 110, with an average of 31. For example, Figure 4.3 is a snapshot of the binary representation of patient information for 50 different episodes. The y-axis depicts around 40 distinct diagnosis icd-9 codes. In the construction of the matrix, if a patient has a specific diagnosis, we will mark that location. We can see from this figure that the patient feature matrix is extremely sparse, with a density of only 3%. If we directly use these features, the sparsity and the relative small number of patients makes learning more difficult and is computationally expensive. Besides, simplification of models can make them easier to interpret by researchers and enhance generalization by reducing overfitting. To this end, we instead find the lower dimension feature representation as explained in the next section.

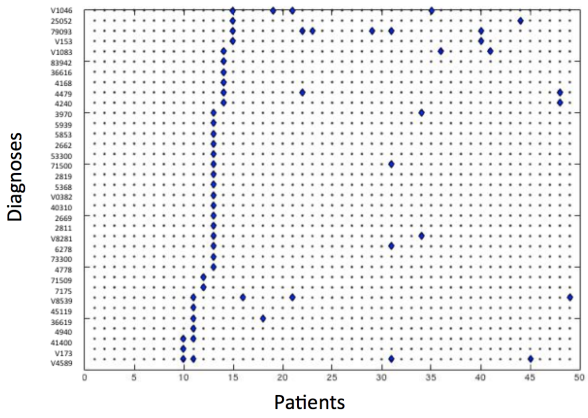


Figure 4.3: Matrix-based patient representation.

4.5.2 Feature Selection

There are several ways to look for more compact representations, such as principal component analysis (PCA), singular value decomposition (SVD) or auto encoder to perform the dimension reduction. However, the features from the non-linear dimension reduction lose the original meanings of the health terminologies. Yet in health care analytics, interpretability of the resulting feature subspace is desired. For example, it is interesting to know whether age or malignant essential hypertension affect the cost or quality of total knee replacement.

Due to the nature of health analytics, many features tend to happen at the same time. For example, in terms of diagnoses, obesity and hypertension are linked, with obese patients having higher rates of hypertension than normal-weight individuals (Chiang et al., 1969). In addition, certain tests are often run together. In order to capture this characteristic, we first cluster the diagnoses into groups based on their occurrences. We construct an undirected network to represent the relationship of different diagnoses as follows. We treat each icd9 diagnosis code as a node in the network. We measure the occurrence similarity of any diagnosis pair (d_1, d_2) by their intersection angle. Specifically, each diagnosis is represented by a 211 dimensional binary vector indicating whether each patient has that diagnosis. We then set a

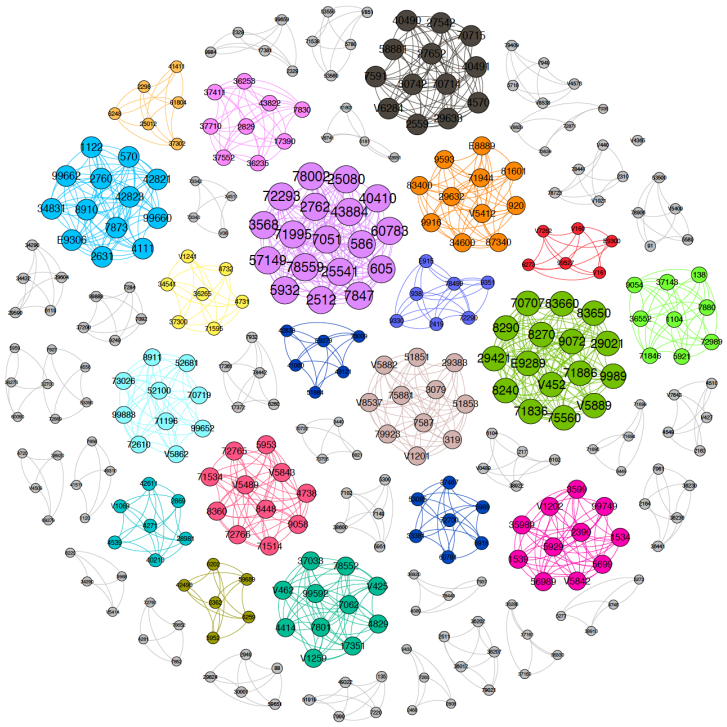


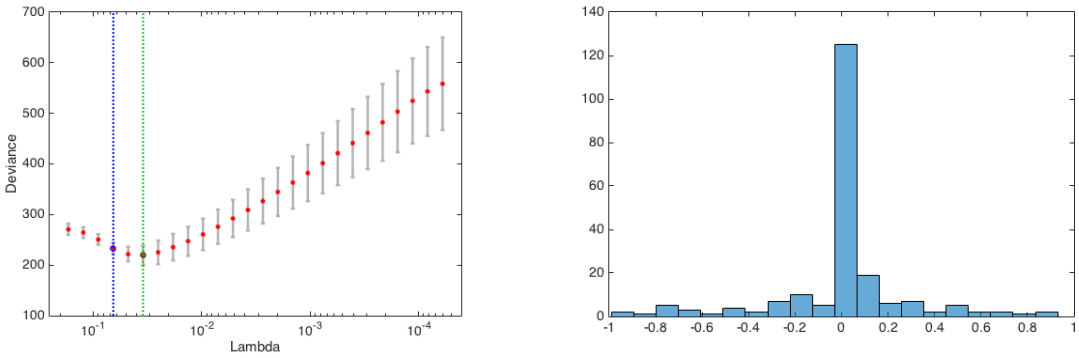
Figure 4.4: Cluster of the diagnoses.

threshold of the cosine of the intersection angle $\frac{\langle d_1, d_2 \rangle}{\|d_1\| \|d_2\|}$. For a diagnosis pair (d_1, d_2), if the cosine of the intersection is larger than the threshold, we draw an edge between them. It is worth noting that if we set the threshold to be 1, it is equivalent to saying that d_1 and d_2 always happen at the same time across all the patients if there is an edge between them. When we set the threshold to 0.8, the presence of an edge means that d_1 and d_2 are recorded together 80 percent of the time.

After the construction of the network, we find the clusters/groups by detecting the weakly connected components in the network. In Figure 4.4, each node is labeled with its icd9 diagnosis code with the size of the node corresponding to its degree. Different colors represent different groups/cliques. The nodes with degree less than 3 are filtered. After clustering, 979 diagnoses have been grouped into 608 components. For example, the red group on the upper right consists of icd9 diagnosis code V160 (Family history of malignant neoplasm of gastrointestinal tract), V161 (Family history of malignant neoplasm of trachea, bronchus, and lung), 99527 (Other drug allergy),

6273 (Postmenopausal atrophic vaginitis), E9300 (Penicillins causing adverse effects in therapeutic use), V7262 (Laboratory examination ordered as part of a routine general medical examination).

Instead of using individual diagnosis codes as the features for the patients, we first use the clusters as features. We further conduct feature selection by selecting a subset of relevant features for use in model construction. Specifically, we use the l_1 regularized (LASSO) logistic regression to yield the sparse solutions of selecting a subset of relevant features using 30 regularization parameter (Lambda) values and 10-fold cross validation on the patient datasets. Figure 4.5(a) identifies the minimum-deviance point with a green circle and dashed line as a function of the regularization parameter. The blue circled point has minimum deviance plus no more than one standard deviation. Under the lambda value (0.0652) at the blue circled point, 25 features are selected in the sparse model and the deviance of the LASSO fit is 233.2678, where the deviance is the value of negative log-likelihood averaged over the validation folds in the cross-validation procedure. In the meantime, it yields a deviance of 99.763 on the entire dataset with a p-value $\ll 0.001$. The corresponding residuals of the sparse model for the 211 episodes are depicted in Figure 4.5(b). In words, the 25 selected features provide reasonable explanatory power for the binary response variable with statistical significance. We then use the 25 selected features as the set of relevant patient attributes to proceed our contextual bandit learning in the next section. Many other interesting statistical questions can be asked regarding this dataset, e.g. feature importance, best prediction model or statistical significance of each explanatory variable. Yet they are not the main focus of this work which addressed the optimal learning challenge with stochastic binary feedback.



(a) Cross-validated deviance of Lasso fit.

(b) Histogram of residuals.

Figure 4.5: Results of Lasso fit.

4.5.3 Community Detection of Caregivers

We discovered from data that not surprisingly there is a number of caregivers (with national provider identifier NPIs) performing the rehabilitation. The caregivers should be divided naturally into communities or modules. Different people doing rehab on the same patient will belong to the same facility. Since some facilities keep patients longer than other ones, in this case, what is important is the facility, not the individual caregiver. If we can detect and characterize this community structure, this should give us a set of "facilities" (in the form of these clusters). To this end, we propose the next hypothesis:

HYPOTHESIS 4.5.1. *There is presence of community structure: different people performing rehabilitation on the same patient belong to the same rehabilitation facility.*

We use the same idea as in the previous section to construct a network that represents the relationships between different caregivers. Each node is one caregiver. We construct the caregiver representation vector as follows: we build an N-dimensional feature vector for each caregiver with each item standing for the frequency of that caregiver giving services to the nth patient. Since two caregivers from the same facility do not always work with the same patients, we lower the threshold of the cosine of their intersection angle to 0.5 to capture this reality.

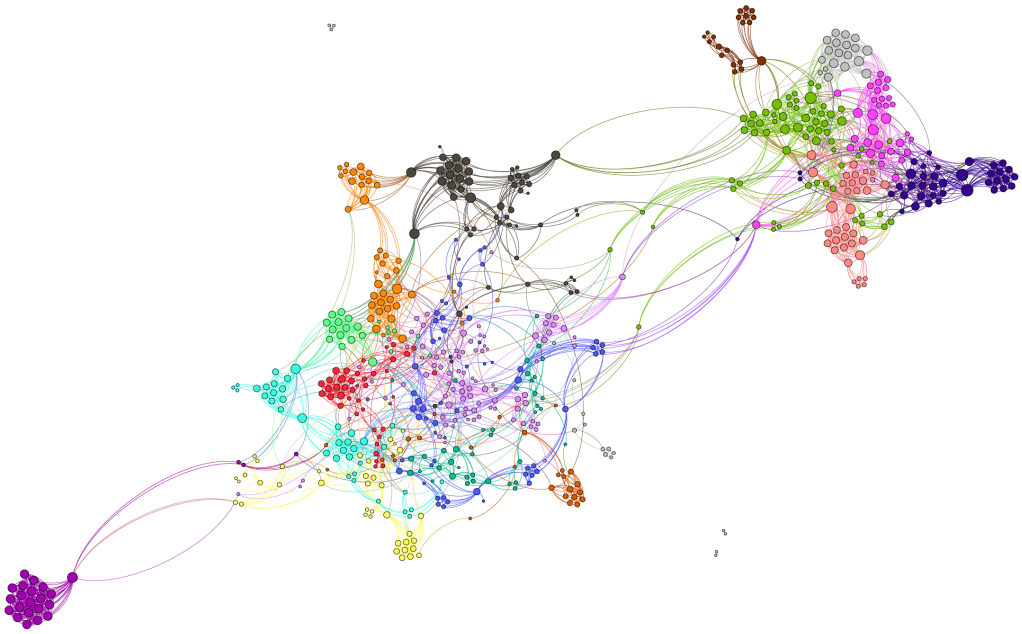


Figure 4.6: Clustering of the caregivers.

In the previous section, when the threshold is 1, the edge represents an equivalence relation (reflexive, symmetric and transitive), making each weakly connected components a clique. Yet when lowering the threshold, it is less meaningful to treat weakly connected components as a group. We instead use the concept of *modularity*, which is the number of edges falling within groups minus the expected number in an equivalent network with edges placed at random (Newman, 2006). A positive modularity value indicates the possible presence of community structure. We detect the modularity of the total 768 caregivers using spectral community detection (Newman, 2006), which maximizes the modularity defined as:

$$Q = \frac{1}{m} \sum_{i,j} \left(A_{ij} - \frac{k_i k_j}{2m} \right) (s_i s_j + 1).$$

Here, for any division of the network into two groups, $s_i = 1$ if vertex i belongs to group one. The number of edges between two vertices i, j is A_{ij}, k_i is the degrees of vertex i, and the total number number edges in the network is defined as $m = \frac{1}{2} \sum_i k_i$.

Figure 4.6 depicts the modular classes (58 in total) with different classes drawn in different colors. The community detection yields a modularity score around 0.5. Network visualization provides strong hints of connectional relationships. We can see from the figure that nodes within each modular class have dense connections with each other while inter-modular-class connections are sparse. This provides strong evidence of the existence of the rehabilitation facilities.

4.5.4 Personalized Physicians and Caregivers Assignment

We now study the problems of how physicians and facilities (rather than each caregiver) affect the post-operative cost of each patient. We use Gaussian process classification, as described in Section 4.3, to model the probability of the post-operative cost below the threshold given a patient feature vector $\boldsymbol{\phi}^X(x)$: the probability of success of each physician and/or facility a is $\sigma\big(\boldsymbol{w}_a^T \boldsymbol{\phi}^X(x)\big)$ for some unknown weight vector $\boldsymbol{w}_a$ to be learned. We have a different weight vector for each physician and/or facility, which is affordable since the number of physicians and/or facilities and patient features are both small, i.e. in our experiments the number of physicians and/or facilities is 20 and the dimension of patient attributes is 25 after clustering and LASSO. This is equivalent to decomposing the kernel $\boldsymbol{\Sigma}$ of the Gaussian process into a product kernel $\boldsymbol{\Sigma} = \boldsymbol{\Sigma}_X \otimes \boldsymbol{\Sigma}_A$, where $(\boldsymbol{\Sigma}_X \otimes \boldsymbol{\Sigma}_A)\big((x,a),(x',a')\big) = \Sigma_X(x,x')\Sigma_A(a,a')$, and choosing $\Sigma_A = \boldsymbol{I}$ as the $|\mathcal{A}| \times |\mathcal{A}|$ identity matrix and Σ_X as the linear kernel on the patient features. For each patient, one and only one physician (or facility) can be assigned.

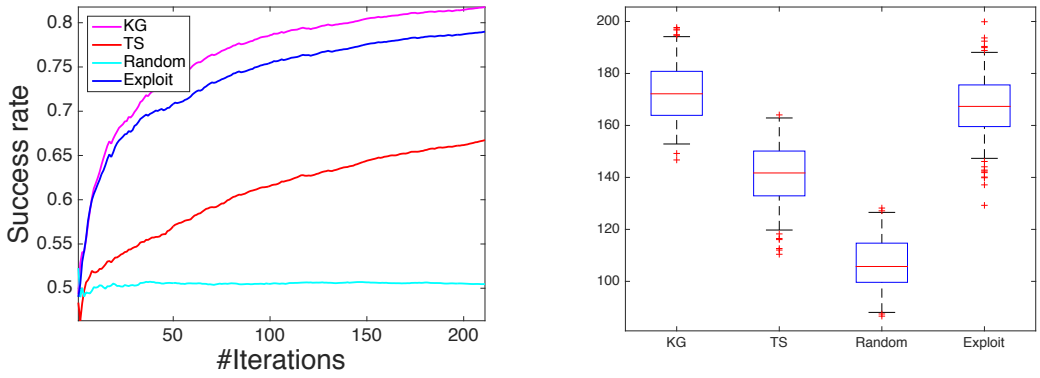
We sort the patient data chronologically. For each patient visit, based on the patient attributes x, we assigned a physician for the surgery and/or a facility for the rehab. We then receive a payoff of whether it is success or failure. The goal is to maximize the number of successes across all 211 patients. There is a fundamental exploration vs. exploitation tradeoff: in order to learn the success rate of each physi-

cian/facility, it needs to be tried for long time benefit, leading to a potential drop in the short-term performance.

Evaluating an exploration/exploitation policy is difficult since we do not know the outcome for physicians and facilities that were not chosen for a particular patient in the record data (Chapelle and Li, 2011). Using real world context and patient features in the knee replacement dataset, we instead simulate the true outcomes using a weight vector $\boldsymbol{w}^*$. This weight vector could be chosen arbitrarily, but it was in fact a perturbed version of some weight vector learned from real data. Although we have modeled the choice of physician and rehabilitation facilities, these are handled in an identical way as in $\sigma\big(\boldsymbol{w}_a^T \boldsymbol{\phi}^X(x)\big)$, and as a result we focused just on the choice of physician.

The number of available physicians is $M = 20$. The experimental results are reported on 100 repetitions of each algorithm. The only difference is the way each policy selects the actions; all the rest, including the model updates, is identical as described in Section 4.3.

We compare our policy with pure exploitation (which assigns the physician that seems to be the best), pure exploration (which randomly assign a physician, as would happen if you assigned the first available physician) and Thompson sampling (Thompson, 1933). Pure exploration is the most used strategy in nowadays health systems. Thompson sampling has been successfully applied to two-treatment adaptive designs (Berry and Eick, 1995; Hu and Rosenberger, 2006) and other applications (Graepel et al., 2010; Agrawal and Goyal, 2012). In our personalized health settings, at each time step n, given the patient information x^n, it first draws a sample $\hat{\boldsymbol{w}}$ according to the posterior distribution $p(\boldsymbol{w}|\mathcal{D}^n)$. It then selects a treatment a^n (that is, the physician) that maximizes the probability of success under the sample parameter value $a^n = \arg\max_a p(y = +1|x^n, a, \hat{\boldsymbol{w}})$.



(a) Cumulative success rate.

(b) Distribution of successes.

Figure 4.7: Comparison of different algorithms on the knee replacement dataset.

We report the number of cumulative success divided by the number of treated patients after each of the 211 patient visits in Figure 4.7(a). We also report the distribution of the number of successes produced by each policy after last patient visit in Figure 4.7(b). On each box, the central red line is the median, the edges of the box are the 25th and 75th percentiles, and outliers are plotted individually.

We can see from the figures that the KG policy yields the best performance. Pure exploitation also does comparatively well. This seems at first a bit odd given that the system has no prior knowledge about the true parameter value $\boldsymbol{w}^*$. A possible explanation is that the change in context induces some level of exploration. In the meantime, KG policy is more robust than pure exploitation with smaller variance and fewer outliers. Random assignment does not perform well since it is not learning from its past experiences while other policies use their past observations to guide the next assignment. We conclude that even though the data is sparse (no two patients are alike), through careful selection of physicians we can improve success rates by around 60 percent over current strategy (pure exploration) used by many health systems.

The real value of the knowledge gradient is its rapid learning, which is especially important in a health setting since if we can learn faster, we can benefit more patients. The knowledge gradient correctly captures the full value of information, prop-

erly balancing exploitation (doing well now) and exploration (learning to do well in the future). Thompson sampling captures the exploration-exploitation tradeoff only approximately. The appeal of Thompson sampling is the ease of computation which is useful in high-frequency internet applications. Other optimizing policies such as pure exploitation (a greedy policy based on the prior) or Bayes greedy (a greedy policy based on the posterior, similar to Thompson sampling) do not accurately capture the value of information which requires capturing the value of reducing the uncertainty in the belief.

In the meantime, we also report the sampling frequency of each physician on three randomly chosen runs in Figure 4.8. In the "Truth" figure, it depicts the number of times each of the $M = 20$ physicians is the actual best physician for the 211 patients, under the underlying true parameter value $\boldsymbol{w}^*$ (which is unknown to the learner). The x-axis is corresponding to the 20 physicians. For example, in the first row, physician #13 is in general a very good physician who is the best for around 70 different patients. The histogram of each policy illustrates the number of time the policy chose to assign each physician. For example, in the first run, KG policy assigned physician #13 for around 95 times in a total of 211 episodes. We can see from the figures that the KG policy manages to achieve a sampling pattern very similar to the true physician distribution. It largely concentrates on the best physician for each patient with moderate exploration. As seen before, pure exploration learns nothing from the past and it fails to infer anything about the performance of each physician. Thompson sampling is also discovered to explore more than necessary, which explains the lower success rate by waisting time on unpromising physicians. It is known that pure exploitation is a non-consistent policy and will generally become stuck without further improvement. In our case, pure exploitation concentrates samples on a very limited portion of physicians without even trying others once. This will lead to a biased estimation of the performance of each physician.

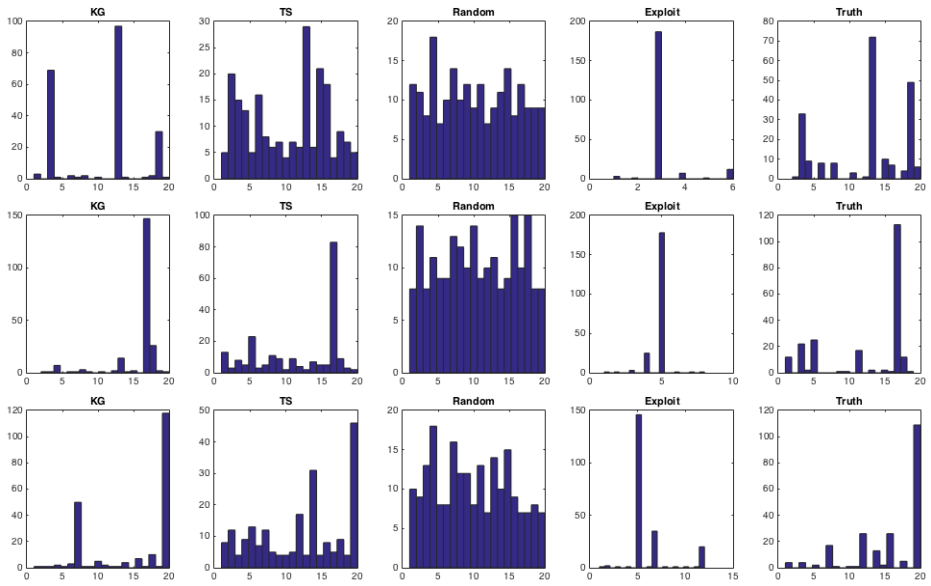


Figure 4.8: Sampling frequency of each physician.

Table 4.2: Summarized statistics on the number of times each policy assigned the actual best physician.

	KG	TS	Random	Exploit
Mean	55.8200	31.5500	10.8100	40.1800
Standard deviation	30.2188	13.8290	3.0771	33.6048

We also summarize the sampling statistics over 100 runs in Table 4.2. It reports the number of times the physician assigned by each policy is actually the best physician for each patient and its standard deviation, averaged over 100 runs. We can see that KG yields the highest number of correctness, which is consistent with the previous discovery on the sampling pattern.

In sum, beyond the fact that KG achieves the highest cumulative success rate, it also rapidly learns the underlying distribution of the physicians, which means that KG well balances exploitation v.s. exploration and it is expected to have a reasonable generalization on the entire population.

4.6 Conclusion

In this chapter, we consider the problem of personalized medicine which formalizes clinical decision making as a function that maps individual patient information to a recommended treatment. The learner is rewarded by "successes" and "failures" which can be predicted through an unknown relationship that depends on the patient information and the selected treatment. Each experiment is expensive, forcing us to learn the most from each experiment. The goal is to treat current patients as effectively as possible and correctly identify the better treatment as quickly as possible. We adopt a Bayesian approach both to incorporate possible prior information and to update our treatment regime continuously as information accrues, with the potential to allow smaller yet more informative trials and for patients to receive better treatment. We formulate the problem as contextual bandits which use context information to explicitly model the heterogeneity in needs and responses of different patients. We for the first time introduce a two-step Bellman's equation, based on which a knowledge gradient policy is developed for Bayesian contextual bandits. The context-specific best action is a more demanding benchmark than the best action identification in the context-free case. We provide a detailed study on the problem of how sequentially assignment of physicians/facilities to individual patients can reduce the health care cost. We use modularity detection and LASSO to deal with the intrinsic sparsity in health datasets. We show experimentally that even though the problem is sparse, through careful selection of physicians (versus picking them at random), we can significantly improve the success rates.

Chapter 5

Ensemble Bayesian Optimization for Sequential Information Processes

The existing literature on Bayesian optimization, multi-armed bandits and optimal learning assumes a single prediction model for describing the performance of each alternative. For example, different belief models have been studied under the name of contextual bandits, including linear models (Chu et al., 2011a) and Gaussian processes (Krause and Ong, 2011). There is a literature on Bayesian optimization (He et al., 2007; Chick, 2001; Powell and Ryzhov, 2012). EGO (and related methods such as SKO (Jones et al., 1998; Huang et al., 2006)) assumes a Gaussian process belief model which does not scale to the higher dimensional settings that we consider. Others assume lookup table, or low-dimensional parametric methods (e.g. response surface/surrogate models (Gutmann, 2001; Jones, 2001; Regis and Shoemaker, 2005)). We are particularly interested in a knowledge gradient policy that maximizes the value of information, since this approach is particularly well suited to problems where observations are expensive. After its first appearance for ranking and selection

problems (Frazier et al., 2008), KG has been extended to various other belief models (e.g. hierarchical belief model in Mes et al. (2011), linear belief model in Negoescu et al. (2011), logistic regression in Wang et al. (2016)).

An open question, however, is the identification of which belief model is most appropriate for a given problem. For example, a patient can have a number of attributes, spanning the usual age, weight, gender, ethnicity, body type, to the information about their condition (diagnoses), to their medical history. Yet if we directly use these features, the sparsity and the relatively small number of patients makes learning more difficult and is computationally expensive. Besides, simplification of models can make them easier to interpret by researchers and enhance generalization by reducing overfitting. We could instead find the lower dimension feature representation or perform dimensionality reduction based on prior knowledge such as previously learned patient profiles. Yet if a patient deviates from stereotypical patients, then a reduced space may not include enough explanatory power. One question is how to appropriately choose the explanatory variables. Another essential question is what type of prediction model should be chosen among many competing models, such as perceptron, support vector machines (SVM), and decision trees. Ensemble learning is of vital importance in these cases. Ensemble learning is the process by which multiple models, such as classifiers or experts, are strategically generated and combined to solve a particular computational intelligence problem. For example, we ask the opinions of different doctors before deciding on a medical procedure, and we ask the views of parents, friends and advisors before we make a life decision. In each case, the primary goal of ensemble learning is to minimize the incorrect selection of a particularly poorly performing prediction model.

Boosting is one of the most powerful ensemble learning techniques in batch learning. It aims to convert weak learners to strong ones. The theoretical perspective of boosting in the batch setting has been studied extensively (Schapire and Freund,

2012), with a huge practical impact (He and Thiesson, 2007; Ferreira and Figueiredo, 2012). Yet sequential decision problems does not require a batch of training data beforehand but processes streaming examples one by one. Although there are relatively few existing studies on online boosting algorithms, as opposed to their offline counterparts, there is an increasing interest in the realm of online learning (Beygelzimer et al., 2015; Chen et al., 2012; Oza, 2005; Babenko et al., 2009; Lin et al., 2014).

In this chapter, similar to the idea of online boosting, we use Bayesian learning with expert advise as the belief model for sequential decision making problems for statistical, computational and representational reasons (Dietterich, 2000), aiming to improve the prediction of the performance of each alternative overall, so as to spend the limited measurement budget more wisely. To the best of our knowledge, this work is the first attempt to use an online boosting framework as the prediction model in Bayesian optimization and multi-armed bandit literature. We use logistic learners as an illustration of the base models and derive an efficient and practical algorithm for ensemble sequential decision making. Synthetic experiments and real-world experiments on a knee replacement dataset demonstrate the effectiveness of our proposed algorithms.

5.1 Problem Formulation

We assume that we have a finite set of alternatives $\boldsymbol{x} \in \mathcal{X} = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_m\}$ and an unknown function $\mu : \mathcal{X} \mapsto \mathbb{R}$ which represents the underlying performance of an experiment. The learner sequentially chooses a series of alternatives $\{\boldsymbol{x}^0, \boldsymbol{x}^1, \cdots | \boldsymbol{x}^i \in \mathcal{X}\}$ to measure. After choosing each of the point $\boldsymbol{x}^n$, the learner will receive a feedback y^{n+1}. The choice of $\boldsymbol{x}^n$ can be made based on past observations $\{\boldsymbol{x}^0, y^1, \boldsymbol{x}^1, y^2, \cdots, \boldsymbol{x}^{n-1}, y^n\}$. Any strategy that specifies the choice of measurement at each time step is called a policy π. Under a limited measurement

budget N, the goal of the learner is to recommend an implementation decision $\boldsymbol{x}^N$ that maximizes the performance $\mu(\boldsymbol{x})$, or equivalently, minimizing the simple regret, $\max_{\boldsymbol{x}\in\mathcal{X}} \mu(\boldsymbol{x}) - \mu(\boldsymbol{x}^N)$. There are two optimization problems involved: 1) we would like to optimize a prediction model to most accurately describe the performance of each alternative; 2) We need to spend our limited measurement budget wisely in order to find out the best alternative that has highest expected performance.

5.2 Bayesian Learning with Expert Advice

Boosting is one of the most powerful ensemble learning techniques in batch learning. It aims to convert weak learners to strong ones. In batch learning settings, the whole set S of training examples are given beforehand. In each round n, it chooses a distribution p^n over the training set and feeds p^n and S to a weaker learner which in return produces a weak hypothesis h^n. After N rounds, N weak hypotheses are combined to produce a final hypothesis of the form $H^N(\boldsymbol{x}) = \sum_{n=1}^{N} \alpha^n h^n(\boldsymbol{x})$, where $\alpha^n \in \mathbb{R}$ is the voting schema of h^n.

In our sequential decision problem, the examples only become available one at a time. Thus, the online boosting algorithm must fix the number of K weak learners before it starts (Chen et al., 2012). In each round n, a new example is fed to K weak learners and they return K updated weak hypotheses h_k^{n+1}, and the boosting algorithm predicts the next example $\boldsymbol{x}^{n+1}$ by $H^{n+1}(\boldsymbol{x}) = \sum_{k=1}^{K} \alpha_k^{n+1} h_k^{n+1}(\boldsymbol{x})$, where $\alpha_k^{n+1} \in \mathbb{R}$ is the voting weight of h_k^{n+1}.

Similarly to the case of online boosting, we fix the number of base (prediction) models as K. Each of the base model $h_k(\cdot;\boldsymbol{\theta}_k)$ is parameterized by some vector $\boldsymbol{\theta}_k$. The final model is an ensemble system obtained by $H(\boldsymbol{x}) = \sum_{k=1}^{K} \alpha_k h_k(\boldsymbol{x};\boldsymbol{\theta}_k)$.

5.2.1 Generalized Weighted Majority

In sequential decision making problems, each data point is actively selected by some policy. This is very different from the PAC learning model whose key assumption is that the distribution over data points is fixed over time, both for training and test points, and that samples are i.i.d. In contrast, in our case, no distributional assumption is made. In this regard, we borrow the ideas from online learning and treat each base model as an *expert*.

The generalized weighted majority algorithm works as follows (Algorithm 6). First, set initial weights to each expert $w_i = 1/K$, where K is the number of base models. At each time step n, select expert ε_i in proportion to the normalized weights $\frac{w_i^n}{\sum_j w_j^n}$ and predict the same as what expert ε_i predicts. The learner receives the actual outcome and each expert incur a loss $l^n(\varepsilon_i) \in [0, 1]$. Adjust the weight of each expert by $w_i^{n+1} = w_i^n e^{-\alpha l^n(\varepsilon_i)}$.

Algorithm 6: Generalized Weighted Majority Algorithm

Initialize $w_i^0 = 1/K$, for all i
for $n = 1$ *to* N **do**
  Receive example x^n
  Predict expert ε_i in proportion to the normalized weights, $w_i^n / \sum_j w_j^n$
  Receive loss: $l^n(\varepsilon_i)$ for all i
  Update weights of experts: $w_i^{n+1} = w_i^n e^{-\alpha l^n(\varepsilon_i)}$, $\forall i$, with $l^n(\varepsilon_i) \in [0, 1]$
end

5.2.2 A Bayesian Interpretation

Many loss functions can be represented as log likelihoods. To this end, suppose each expert ε_k's prediction is a probability distribution $p_k(y|x)$. This is natural either for continuous or discrete responses, for example, in Bayesian optimization where a Gaussian additive noise is added to the observation $y = f(x) + \epsilon$. Or, as in logistic regression, the probability of class +1 can be written as $\sigma(\boldsymbol{\theta}^T \boldsymbol{x})$. Standard loss for

making a probabilistic prediction is log-loss, which is corresponding to log likelihoods:

$$l^n(\varepsilon_k) = \log\big(p_k(y^{n+1}|x)\big), \tag{5.1}$$

where y^{n+1} is the true observation. Plugging the log-loss into the weight update rule, we have

$$w_k^{n+1} = w_k^n e^{\alpha \log\left(p_k(y^{n+1}|x)\right)} = w_k^n \big(p_k(y^{n+1}|x)\big)^\alpha.$$

In the meantime, at time step n, w_k^n can be interpreted as a prior on experts $p(\varepsilon_k)$. Now, the posterior distribution given a new data point (x^n, y^{n+1}) can be obtained by Bayes' theorem:

$$w_k^{n+1} = p(\varepsilon_k|y^{n+1}) \propto p(y^{n+1}|\varepsilon_k)p(\varepsilon_k) = w_k^n p_k(y^{n+1}|x).$$

We can see that when setting $\alpha = 1$ in weighted majority algorithm (Eq. (5.1)), the update of the weights becomes Bayes' rule. This interpretation gives us great flexibility to design Bayesian policies based on the value of information.

5.3 Bayesian Optimal Learning with Ensembles

As of this writing, there is no KG variant designed for ensemble models. In what follows, we first formulate our learning problem as a Markov decision process and then extend the KG policy for ensemble systems.

5.3.1 Markov Decision Process Formulation

We assume our base models are Bayes' learners. To be more specific, each of the models $h_k(\cdot;\boldsymbol{\theta})$ is parameterized by some parameters $\boldsymbol{\theta}_k$. Each observation is modeled using some known likelihood function $P\big(y|h_k(\boldsymbol{x},\boldsymbol{\theta}_k)\big)$. For a Bayes' learner, $\boldsymbol{\theta}_k$ is a random variable, and each of the learners will consequently update their beliefs on

the unknown parameters through Bayesian inference. Given some prior distribution $P(\boldsymbol{\theta}_k)$ on these parameters, the posterior distribution for one observed data point $(\boldsymbol{x}, y)$ can be obtained by Bayes' theorem,

$$q(\boldsymbol{\theta}_k) := P\big(\boldsymbol{\theta}_k|\boldsymbol{x}, y\big) \propto P\big(y|h_k(\boldsymbol{x}, \boldsymbol{\theta}_k)\big)P(\boldsymbol{\theta}_k).$$

Our learning problem is thus a dynamic program that can be formulated as a Markov decision process as follows. First, we can define the state space $\mathcal{S}$ as the cross-product of $[0,1]^K$ and the space of all possible predictive distributions for $\boldsymbol{\theta}_k$ of each Bayes' learner. This captures the probability distributions that describe the uncertainty about unknown parameters. We will write $S^n = \big((w_1^n, \cdots, w_K^n), q^n(\boldsymbol{\theta}_1), \cdots, q^n(\boldsymbol{\theta}_K)\big)$ to refer to the state at time n after previous n observations D^n, where $q^n(\cdot)$ is defined as the posterior distribution $q^n(\boldsymbol{\theta}_k) := P\big(\boldsymbol{\theta}_k|D^n\big)$ at time n. Recall that the way the Generalized Weighted Majority algorithm works can be interpreted as Bayes' rule. The weight update can be obtained by marginalizing over the parameters $\boldsymbol{\theta}$. We can summarize the Bayesian inference of the ensemble system as follows, for $\forall k = 1, \cdots, K$:

$$\begin{aligned} w_k^{n+1} := P(\varepsilon_k|y^{n+1}, D^n) &\propto P(y^{n+1}|\varepsilon_k, D^n)P(\varepsilon_k|D^n) \\ &= w_k^n \int_{\boldsymbol{\theta}_k} P(y^{n+1}|\varepsilon_k, \boldsymbol{\theta}_k, D^n)P(\boldsymbol{\theta}_k|D^n)\mathrm{d}\boldsymbol{\theta}_k \\ &= w_k^n \int_{\boldsymbol{\theta}_k} P\big(y^{n+1}|h_k(\boldsymbol{x}^n, \boldsymbol{\theta}_k)\big)q^n(\boldsymbol{\theta}_k)\mathrm{d}\boldsymbol{\theta}_k, \end{aligned} \tag{5.2}$$

$$\begin{aligned} q^{n+1}(\boldsymbol{\theta}_k) := P\big(\boldsymbol{\theta}_k|y^{n+1}, D^n\big) &\propto P(y^{n+1}|\boldsymbol{\theta}_k)P(\boldsymbol{\theta}_k|D^n) \\ &= P\big(y^{n+1}|h_k(\boldsymbol{x}^n, \boldsymbol{\theta}_k)\big)q^n(\boldsymbol{\theta}_k). \end{aligned} \tag{5.3}$$

We can thus define the transition function $T : \mathcal{S} \times \mathcal{X} \times \mathcal{Y}$ as:

$$\begin{aligned} & T\Big(\big((w_1, \cdots, w_K), q(\boldsymbol{\theta}_1), \cdots, q(\boldsymbol{\theta}_K)\big), \boldsymbol{x}, y\Big) \\ = \; & \Big(\big[w_k \int_{\boldsymbol{\theta}_k} P\big(y|h_k(\boldsymbol{x}, \boldsymbol{\theta}_k)\big) q(\boldsymbol{\theta}_k) \mathrm{d}\boldsymbol{\theta}_k / Z_w\big]_{k=1}^K, \big[P\big(y|h_k(\boldsymbol{x}, \boldsymbol{\theta}_k)\big) q(\boldsymbol{\theta}_k) / Z_{\boldsymbol{\theta}_k}\big]_{k=1}^K\Big), \end{aligned} \tag{5.4}$$

where $\mathcal{Y}$ is the domain of the observations, $Z_w = \sum w_k \int_{\boldsymbol{\theta}_k} P\big(y|h_k(\boldsymbol{x}, \boldsymbol{\theta}_k)\big) q(\boldsymbol{\theta}_k) \mathrm{d}\boldsymbol{\theta}_k$ and $Z_{\theta_k} = \int_{\boldsymbol{\theta}_k} P\big(y|h_k(\boldsymbol{x}, \boldsymbol{\theta}_k)\big) q(\boldsymbol{\theta}_k) \mathrm{d}\boldsymbol{\theta}_k$ are the normalizing factors, so that $S^{n+1} = T(S^n, \boldsymbol{x}, Y^{n+1})$ with Y^{n+1} as the unobserved outcome random variable at time n.

In a dynamic program, the value function is defined as the value of the optimal policy given a particular state S^n at time n, and may be determined recursively through Bellman's equation. For any state $s \in \mathcal{S}$, since the goal is to maximize $\mu(\boldsymbol{x}^N)$, if there is no measurement left, the value of each state is the maximal value we can get under our current belief. Thus, for our ensemble system, by the law of total expectation, the terminal value function $V^N : \mathcal{S} \mapsto \mathbb{R}$ is given by

$$\begin{aligned} V^N(s) = \max_{\boldsymbol{x}} \mathbb{E}[Y|s, x] &= \max_{\boldsymbol{x}} \sum_{k=1}^K \mathbb{E}[Y|s, \boldsymbol{x}, \varepsilon = \varepsilon_k] P(\varepsilon_k|s) \\ &= \max_{\boldsymbol{x}} \sum_{k=1}^K w_k \int_{\theta_k} q(\theta_k) \mathbb{E}[Y|h_k(\boldsymbol{x}, \boldsymbol{\theta}_k)] \mathrm{d}\boldsymbol{\theta}_k, \end{aligned} \tag{5.5}$$

for any $s = \big((w_1, \cdots, w_K), q(\boldsymbol{\theta}_1), \cdots, q(\boldsymbol{\theta}_K)\big) \in \mathcal{S}$.

The dynamic programming principle tells us that the value at any other time $n = 1, \cdots, N$, V^n is given recursively by

$$V^n(s) = \max_{\boldsymbol{x}} \mathbb{E}[V^{n+1}(T(s, \boldsymbol{x}, Y^{n+1}))|\boldsymbol{x}, s], \forall s \in \mathcal{S}.$$

Since the curse of dimensionality on the state space $\mathcal{S}$ makes direct computation of the value function intractable, we develop a knowledge gradient type policy for our ensemble system in the next section.

5.3.2 Knowledge Gradient with Ensembles

The concept of knowledge gradient is the expected improvement in value of measuring an alternative $\boldsymbol{x}$ (Frazier et al., 2008):

DEFINITION 5.3.1. *The knowledge gradient of measuring an alternative $\boldsymbol{x}$ while in state s is*

$$\nu_{\boldsymbol{x}}^{KG}(s) := \mathbb{E}\big[V^N\big(T(s,\boldsymbol{x},Y)\big) - V^N(s)|\boldsymbol{x},s\big].$$

In the case of ensemble systems, given any state $s = \big((w_1,\cdots,w_K), q(\boldsymbol{\theta}_1),\cdots,q(\boldsymbol{\theta}_K)\big)$, the outcome y of an alternative $\boldsymbol{x}$ is a random variable with a predictive distribution, marginalized over the posterior,

$$p(y|\boldsymbol{x},s) = \sum_{k=1}^{K} w_k \int_{\boldsymbol{\theta}_k} P(y|h_k(\boldsymbol{x},\boldsymbol{\theta}_k))q(\boldsymbol{\theta}_k)\mathrm{d}\boldsymbol{\theta}_k = Z_w.$$

Hence by some algebra and recalling from Eq. (5.4) and Eq. (5.5) the definition of $T(s,\boldsymbol{x},y)$ and $V^N(s)$, we have

$$\begin{aligned}
& \mathbb{E}\big[V^N\big(T(s,\boldsymbol{x},Y)\big)|\boldsymbol{x},s\big] \\
= & \int_y p(y|\boldsymbol{x},s)V^N(T(s,\boldsymbol{x},y))\mathrm{d}y \\
= & \int_y p(y|\boldsymbol{x},s)\cdot\max_{\boldsymbol{x}'}\sum_{k=1}^{K}\frac{w_k\int_{\boldsymbol{\theta}_k}P(y|h_k(\boldsymbol{x},\boldsymbol{\theta}_k))q(\boldsymbol{\theta}_k)\mathrm{d}\boldsymbol{\theta}_k}{p(y|\boldsymbol{x},s)}\int_{\boldsymbol{\theta}_k}\frac{P\big(y|h_k(\boldsymbol{x},\boldsymbol{\theta}_k)\big)q(\boldsymbol{\theta}_k)}{Z_{\boldsymbol{\theta}_k}}\mathbb{E}[y|h_k(\boldsymbol{x}',\boldsymbol{\theta}_k)]\mathrm{d}y \\
= & \int_y \max_{\boldsymbol{x}'}\sum_{k=1}^{K} w_k\int_{\boldsymbol{\theta}_k}\mathbb{E}[y|h_k(\boldsymbol{x}',\boldsymbol{\theta}_k)]P\big(y|h_k(\boldsymbol{x},\boldsymbol{\theta}_k)\big)q(\boldsymbol{\theta}_k)\mathrm{d}\boldsymbol{\theta}_k\mathrm{d}y. \qquad (5.6)
\end{aligned}$$

Since under current belief state s, $V^N(s)$ does not depend on the next measurement, the knowledge gradient policy for ensemble belief model can thus be defined at each time step, choosing the alternative that yields the largest expected incremental

value:

$$\begin{aligned} X^{\pi,\mathrm{KG}}(s) &:= \arg\max_{\boldsymbol{x}} \nu_{\boldsymbol{x}}^{\mathrm{KG}}(s) \\ &= \arg\max_{\boldsymbol{x}} \int_y \max_{\boldsymbol{x}'} \sum_{k=1}^{K} w_k \int_{\boldsymbol{\theta}_k} \mathbb{E}[y|h_k(\boldsymbol{x}',\boldsymbol{\theta}_k)] P\big(y|h_k(\boldsymbol{x},\boldsymbol{\theta}_k)\big) q(\boldsymbol{\theta}_k) \mathrm{d}\boldsymbol{\theta}_k \mathrm{d}y. \end{aligned}$$

We next show the positive value of information, which is the benefit of measurement, meaning that the more measurement, the higher the objective function value. This is essential for studying the theoretical behavior of the KG policy in that if an alternative cannot provide any more information, we will never choose it again unless the performance of all the alternatives are perfectly learnt.

LEMMA 5.3.2. *The knowledge gradient of measuring any alternative $\boldsymbol{x}$ while in any state $s \in \mathcal{S}$ is nonegative, $\nu_{\boldsymbol{x}}^{KG}(s) \geq 0$.*

Proof. By Jensen inequality, for any $\boldsymbol{x}$,

$$\begin{aligned} \mathbb{E}\big[V^N\big(T(s,\boldsymbol{x},Y)\big)|\boldsymbol{x},s\big] &= \int_y \max_{\boldsymbol{x}'} \sum_{k=1}^{K} w_k \int_{\boldsymbol{\theta}_k} \mathbb{E}[y|h_k(\boldsymbol{x}',\boldsymbol{\theta}_k)] P\big(y|h_k(\boldsymbol{x},\boldsymbol{\theta}_k)\big) q(\boldsymbol{\theta}_k) \mathrm{d}\boldsymbol{\theta}_k \mathrm{d}y \\ &\geq \max_{\boldsymbol{x}'} \int_y \sum_{k=1}^{K} w_k \int_{\boldsymbol{\theta}_k} \mathbb{E}[y|h_k(\boldsymbol{x}',\boldsymbol{\theta}_k)] P\big(y|h_k(\boldsymbol{x},\boldsymbol{\theta}_k)\big) q(\boldsymbol{\theta}_k) \mathrm{d}\boldsymbol{\theta}_k \mathrm{d}y \\ &= \max_{\boldsymbol{x}'} \sum_{k=1}^{K} w_k \int_y P\big(y|h_k(\boldsymbol{x},\boldsymbol{\theta}_k)\big) \mathrm{d}y \int_{\boldsymbol{\theta}_k} \mathbb{E}[y|h_k(\boldsymbol{x}',\boldsymbol{\theta}_k)] q(\boldsymbol{\theta}_k) \mathrm{d}\boldsymbol{\theta}_k \\ &= \max_{\boldsymbol{x}'} \sum_{k=1}^{K} w_k \int_{\boldsymbol{\theta}_k} \mathbb{E}[y|h_k(\boldsymbol{x}',\boldsymbol{\theta}_k)] q(\boldsymbol{\theta}_k) \mathrm{d}\boldsymbol{\theta}_k = V^N(s). \end{aligned}$$

Since $V^N(s)$ is deterministic under current state s, we have

$$\nu_{\boldsymbol{x}}^{\mathrm{KG}}(s) = \mathbb{E}\big[V^N\big(T(s,\boldsymbol{x},Y)\big)|\boldsymbol{x},s\big] - V^N(s) \geq 0.$$

□

5.3.3 Derivation for Bayesian Logistic Learners

In general, the prior and the likelihood are not necessarily conjugate to each other, so that there is no compact representation of the state space since the posterior distributions are not usually in the same family as the prior probability distribution. In the meantime, the knowledge gradient value requires the computation of the expectation of a maximization which is a difficult computational challenge, and a closed-form solution can be seldom obtained.

In this section, we consider the case of binary outcomes and our goal is to maximize the probability of success. Many real world applications easily fit into the success/failure model. For example, in online advertisement, a user feedback is whether or not he/she clicks on the displayed ad. In health care, the outcome is whether a treatment is successful. In loan applications, a lending company needs to minimize the default rate.

We assume that we have a finite set of alternatives $\boldsymbol{x} \in \mathcal{X} = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_m\}$ and an unknown function $\mu : \mathcal{X} \mapsto [0, 1]$ which represents the underlying binomial probability of success of an experiment. After choosing each of the point $\boldsymbol{x}^n$, the learner will receive a binary feedback y^{n+1} with the probability of $y^{n+1} = 1$ as $\mu(\boldsymbol{x}^n)$. We choose Bayesian logistic regression as base models. That is, for each base model k, the posterior probability of class $+1$ can be written as a link function acting on a linear function of the feature vector

$$\Pr(y = +1|\boldsymbol{x}) = \sigma(\boldsymbol{\theta}_k^T \boldsymbol{x}_{[k]}),$$

with the link function $\sigma(a)$ often chosen as the logistic function $\sigma(a) = \frac{1}{1+\exp(-a)}$ and $\boldsymbol{x}_{[k]}$ can be different sets of features for each base model k, allowing individual classifiers to generate different decision boundaries. Ideally, if proper diversity is achieved, a different error is made by each base model, strategic combination of

which can then reduce the prediction error. For example, in personalized healthcare, a patient can have a number of attributes, including age, weight, gender, diagnoses, and medical history. Directly using these features would cause sparsity and computational inefficiency. Besides, simplification of models can make them easier to interpret by researchers and enhance generalization by reducing overfitting. We could instead find the lower dimension feature representation based on prior knowledge such as historical patient profiles. Yet a reduced space may not include enough explanatory power. In this case, we can develop base learners using multi-level feature hierarchies in order to strike a balance between coarse-to-fine feature representations.

Adapting the concept of Gaussian processes, for each base learner $h_k(\cdot;\boldsymbol{\theta}_k)$ we start with a multivariate prior distribution for the unknown parameter vector $\boldsymbol{\theta}_k$, $\boldsymbol{\theta}_k \sim \mathcal{N}\big(\boldsymbol{m}_k^0, (\boldsymbol{\beta}_k^0)^{-1}\big)$, where $\beta_{k,j}$ is the inverse of variance of $\theta_{k,j}$. Note here a diagonal covariance matrix is adopted due to simplicity and its equivalence to l_2 regularization. The likelihood function $P(y|h_k(\boldsymbol{x},\boldsymbol{\theta}_k))$ in this case is expressed as $\sigma(y\cdot\boldsymbol{\theta}_k^T\boldsymbol{x}_{[k]})$.

For any given data point $(\boldsymbol{x}, y)$, according to Eq. (5.2) and (5.3), the posterior can be calculated as follows:

$$\begin{aligned} w_k^{n+1} &\propto w_k^n \int_{\boldsymbol{\theta}} \sigma(y\cdot\boldsymbol{\theta}_k^T\boldsymbol{x}_{[k]})\mathcal{N}\big(\boldsymbol{m}_k^n, (\boldsymbol{\beta}_k^n)^{-1}\big)\mathrm{d}\boldsymbol{\theta}_k \\ &\approx w_k^n \sigma\big(y\cdot\kappa(\sigma_k^2)\mu_k\big), \\ q^{n+1}(\boldsymbol{\theta}_k) &\propto P(y|\boldsymbol{\theta}_k)q^n(\boldsymbol{\theta}_k) \\ &= \sigma(y\cdot\boldsymbol{\theta}_k^T\boldsymbol{x}_{[k]})\mathcal{N}\big(\boldsymbol{m}_k^n, (\boldsymbol{\beta}_k^n)^{-1}\big), \end{aligned}$$

where $\mu_k = (\boldsymbol{m}_k^n)^T\boldsymbol{x}_{[k]}$ and $\sigma_k = \sum_{j=1}^d (\beta_{k,j}^n)^{-1}x_{[k],j}^2$ (Bishop et al., 2006). Unfortunately, exact Bayesian inference for linear classifiers is intractable since the evaluation of the posterior distribution comprises a product of sigmoid functions; in addition, the integral in the normalization constant is intractable as well. Following the work in Chapelle and Li (2011) and Wang et al. (2016), we use a Laplace approximation

of $q^{n+1}(\boldsymbol{\theta}_k)$, yielding an approximated posterior in the form of $\mathcal{N}\big(\boldsymbol{m}_k^{n+1}, (\boldsymbol{\beta}_k^{n+1})^{-1}\big)$ as follows: for any measurement $\boldsymbol{x}$ and potential outcome $y = +1/-1$,

$$\begin{aligned}
\Psi_k^n(\boldsymbol{\theta}_k, y) &:= -\frac{1}{2}\sum_{j=1}^{d} \beta_{k,j}^n(\theta_{k,j} - m_{k,j}^n)^2 - \log(1+\exp(-y\cdot\boldsymbol{\theta}_k^T\boldsymbol{x}_{[k]})), && (5.7)\\
\boldsymbol{m}_{k\pm}^{n+1} &= \arg\max_{\boldsymbol{w}} \Psi_k^n(\boldsymbol{\theta}_k, \pm 1), && (5.8)\\
\beta_{k\pm,j}^{n+1} &= \beta_{k,j}^n + \sigma\big((\boldsymbol{m}_{k\pm}^{n+1})^T\boldsymbol{x}_{[k]}\big)\Big(1-\sigma\big((\boldsymbol{m}_{k\pm}^{n+1})^T\boldsymbol{x}_{[k]}\big)\Big)x_{[k],j}^2. && (5.9)
\end{aligned}$$

Since each of the posterior is approximated as a normal distribution, the state space can be compactly represented by $s = \Big((w_1,\cdots,w_K),(\boldsymbol{\theta}_1,\boldsymbol{\beta}_1),\cdots,(\boldsymbol{\theta}_K,\boldsymbol{\beta}_K)\Big)$. For the case when y is discrete, recalling from Eq. (5.6), we have,

$$\begin{aligned}
&\mathbb{E}\big[V^N\big(T(S^n,\boldsymbol{x},Y)\big)|\boldsymbol{x},S^n\big]\\
=\ &\max_{\boldsymbol{x}'}\sum_{k=1}^{K} w_k^n\int_{\boldsymbol{\theta}_k}\mathbb{E}[y=+1|h_k(\boldsymbol{x}',\boldsymbol{\theta}_k)]P\big(y=+1|h_k(\boldsymbol{x},\boldsymbol{\theta}_k)\big)q^n(\boldsymbol{\theta}_k)\mathrm{d}\boldsymbol{\theta}_k\\
&+\max_{\boldsymbol{x}'}\sum_{k=1}^{K} w_k^n\int_{\boldsymbol{\theta}_k}\mathbb{E}[y=-1|h_k(\boldsymbol{x}',\boldsymbol{\theta}_k)]P\big(y=-1|h_k(\boldsymbol{x},\boldsymbol{\theta}_k)\big)q^n(\boldsymbol{\theta}_k)\mathrm{d}\boldsymbol{\theta}_k\\
=\ &\max_{\boldsymbol{x}'}\sum_{k} w_k^n\int_{\boldsymbol{\theta}_k}\sigma(\boldsymbol{\theta}_k^T\boldsymbol{x}'_{[k]})\sigma(\boldsymbol{\theta}_k^T\boldsymbol{x}_{[k]})q^n(\boldsymbol{\theta}_k)\mathrm{d}\boldsymbol{\theta}_k\\
&+\max_{\boldsymbol{x}'}\sum_{k} w_k^n\int_{\boldsymbol{\theta}_k}\sigma(\boldsymbol{\theta}_k^T\boldsymbol{x}'_{[k]})\sigma(-\boldsymbol{\theta}_k^T\boldsymbol{x}_{[k]})q^n(\boldsymbol{\theta}_k)\mathrm{d}\boldsymbol{\theta}_k.
\end{aligned}$$

In order to compute this, one can use numerical integration. Yet since $\boldsymbol{\theta}_k$ can be high dimensional and the numerical integration needs to be repeated for any pair of alternatives for each time step, the computation is very time consuming. We instead develop a general approximation schema for $\int\sigma(\boldsymbol{\theta}^T\boldsymbol{x}')\sigma(\boldsymbol{\theta}^T\boldsymbol{x})\mathcal{N}(\boldsymbol{m},\boldsymbol{\beta}^{-1})\mathrm{d}\boldsymbol{\theta}$ to overcome the computational hurdle. First, by linear algebra, we have,

$$\begin{aligned}
&\int_{\boldsymbol{\theta}_k}\sigma(\boldsymbol{\theta}_k^T\boldsymbol{x}'_{[k]})\sigma(\boldsymbol{\theta}_k^T\boldsymbol{x}_{[k]})q^n(\boldsymbol{\theta}_k)\mathrm{d}\boldsymbol{\theta}_k\\
=\ &\int_{\boldsymbol{\theta}_k}\sigma(\boldsymbol{\theta}_k^T\boldsymbol{x}'_{[k]})\frac{\sigma(\boldsymbol{\theta}_k^T\boldsymbol{x}_{[k]})q^n(\boldsymbol{\theta}_k)}{\int_{\boldsymbol{\theta}'_k}\sigma\big((\boldsymbol{\theta}'_k)^T\boldsymbol{x}_{[k]}\big)q^n(\boldsymbol{\theta}'_k)\mathrm{d}\boldsymbol{\theta}'_k}\mathrm{d}\boldsymbol{\theta}_k\cdot\int_{\boldsymbol{\theta}'_k}\sigma\big((\boldsymbol{\theta}'_k)^T\boldsymbol{x}_{[k]}\big)q^n(\boldsymbol{\theta}'_k)\mathrm{d}\boldsymbol{\theta}'_k.
\end{aligned}$$

We can see that $\frac{\sigma(\boldsymbol{\theta}_k^T \boldsymbol{x}_{[k]}) q^n(\boldsymbol{\theta}_k)}{\int_{\boldsymbol{\theta}'_k} \sigma\big((\boldsymbol{\theta}'_k)^T \boldsymbol{x}_{[k]}\big) q^n(\boldsymbol{\theta}'_k) \mathrm{d}\boldsymbol{\theta}'_k}$ is the posterior distribution $q^{n+1}(\boldsymbol{\theta}_k)$ over $\boldsymbol{\theta}_k$ after an observed point $(\boldsymbol{x}, +1)$. Again based on Laplace approximation (Eq. (5.8)(5.9)), the posterior distribution $q^{n+1}(\boldsymbol{\theta}_k)$ can be approximated as a normal distribution $\mathcal{N}\big(\boldsymbol{m}_{k+}^{n+1}, (\boldsymbol{\beta}_{k+}^{n+1})^{-1}\big)$. Then making use of the standard approximation of the convolution of a logistic function and a normal density function (Bishop et al., 2006), we have

$$\begin{aligned}\int_{\boldsymbol{\theta}_k} \sigma(\boldsymbol{\theta}_k^T \boldsymbol{x}'_{[k]}) \frac{\sigma(\boldsymbol{\theta}_k^T \boldsymbol{x}_{[k]}) q^n(\boldsymbol{\theta}_k)}{\int_{\boldsymbol{\theta}'_k} \sigma\big((\boldsymbol{\theta}'_k)^T \boldsymbol{x}_{[k]}\big) q^n(\boldsymbol{\theta}'_k) \mathrm{d}\boldsymbol{\theta}'_k} \mathrm{d}\boldsymbol{\theta}_k &\approx \int_{\boldsymbol{\theta}_k} \sigma(\boldsymbol{\theta}_k^T \boldsymbol{x}'_{[k]}) \mathcal{N}\big(\boldsymbol{m}_{k+}^{n+1}, (\boldsymbol{\beta}_{k+}^{n+1})^{-1}\big) \mathrm{d}\boldsymbol{\theta}_k \\ &\approx \sigma\bigg(\kappa\Big(\tilde{\sigma}^2(\boldsymbol{\beta}_{k+}^{n+1}, \boldsymbol{x}'_{[k]})\Big) \mu(\boldsymbol{m}_{k+}^{n+1}, \boldsymbol{x}'_{[k]})\bigg),\end{aligned}$$

where $\mu(\boldsymbol{m}, \boldsymbol{x}) := \boldsymbol{m}^T \boldsymbol{x}$ and $\tilde{\sigma}^2(\boldsymbol{\beta}, \boldsymbol{x}) := \sum_{j=1}^d \beta_j^{-1} x_j^2$. At the same time,

$$\begin{aligned}\int_{\boldsymbol{\theta}'_k} \sigma\big((\boldsymbol{\theta}'_k)^T \boldsymbol{x}_{[k]}\big) q^n(\boldsymbol{\theta}'_k) \mathrm{d}\boldsymbol{\theta}'_k. &= \sigma\big((\boldsymbol{\theta}'_k)^T \boldsymbol{x}_{[k]}\big) \mathcal{N}(\boldsymbol{m}_k^n, \boldsymbol{\beta}_k^n) \mathrm{d}\boldsymbol{\theta}'_k \\ &\approx \sigma\bigg(\kappa\Big(\tilde{\sigma}^2(\boldsymbol{\beta}_k^n, \boldsymbol{x}_{[k]})\Big) \mu(\boldsymbol{m}_k^n, \boldsymbol{x}_{[k]})\bigg).\end{aligned}$$

A similar approximation derivation can be conducted for the case of

$$\int \sigma(\boldsymbol{\theta}^T \boldsymbol{x}') \sigma(-\boldsymbol{\theta}^T \boldsymbol{x}) \mathcal{N}(\boldsymbol{m}, \boldsymbol{\beta}^{-1}) \mathrm{d}\boldsymbol{\theta}.$$

As a result, for Bayesian logistic learners, the knowledge gradient value can be effectively calculated as

$$\begin{aligned}&\mathbb{E}\big[V^N\big(T(S^n, \boldsymbol{x}, Y)\big) | \boldsymbol{x}, S^n\big] \\ \approx\ &\max_{\boldsymbol{x}'} \sum_k w_k^n \sigma\bigg(\kappa\Big(\tilde{\sigma}^2(\boldsymbol{\beta}_k^n, \boldsymbol{x}_{[k]})\Big) \mu(\boldsymbol{m}_k^n, \boldsymbol{x}_{[k]})\bigg) \sigma\bigg(\kappa\Big(\tilde{\sigma}^2(\boldsymbol{\beta}_{k+}^{n+1}, \boldsymbol{x}'_{[k]})\Big) \mu(\boldsymbol{m}_{k+}^{n+1}, \boldsymbol{x}'_{[k]})\bigg) \\ &+ \max_{\boldsymbol{x}'} \sum_k w_k^n \sigma\bigg(-\kappa\Big(\tilde{\sigma}^2(\boldsymbol{\beta}_k^n, \boldsymbol{x}_{[k]})\Big) \mu(\boldsymbol{m}_k^n, \boldsymbol{x}_{[k]})\bigg) \sigma\bigg(\kappa\Big(\tilde{\sigma}^2(\boldsymbol{\beta}_{k-}^{n+1}, \boldsymbol{x}'_{[k]})\Big) \mu(\boldsymbol{m}_{k-}^{n+1}, \boldsymbol{x}'_{[k]})\bigg),\end{aligned}$$

leading to an efficient and practical algorithm for Bayesian optimization using an ensemble system. The resulting knowledge gradient policy at time step n is

$$X^{\pi,\mathrm{KG}}(S^n) = \arg\max_{\boldsymbol{x}} \mathbb{E}\big[V^N\big(T(S^n, \boldsymbol{x}, Y)\big)|\boldsymbol{x}, S^n\big].$$

5.4 Experimental Results

In this section, we first propose empirical experiments to illustrate the behavior of the knowledge gradient policy with ensemble logistic belief models. We then compare our proposed algorithm with other policies in a real-world knee replacement dataset.

5.4.1 Computational Analysis

There are many real world applications that involves high dimensional feature vectors, but only a few of them have actual explanatory power. For example, as we mentioned previously, modern health data acquisition routinely produces massive amounts of high dimensional datasets, including various patient attributes, medical history, medical images, genetic information, behavior data and unstructured clinical notes. Another example arises in movie recommendation where not only user information can be high dimensional (including watch history, age, geographic information and visited pages), standard technique (feature hashing) also maps movie attributes, e.g. cast, popularity, genre, premiere time, to sparse binary vector of dimension 2^{24}. In batch learning settings, LASSO or dimension reduction can be adopted to achieve feature selection while in our sequential decision making problems, without seeing the remaining examples, it does not seem easy to determine a good set of explanatory variables. To this end, in this section, we consider a set of synthetic experiments with a sparse feature structure.

We randomly sample a set of 300 alternatives $\boldsymbol{x}$ with each alternative as a 100-dimensional feature vector from $[0, 1]$. We assume that the true model is a logistic regression with the weights of only the first 25 predictors non-zero. We randomly generate the weight for each non-zero feature from the distribution $\theta_i^* \sim \mathcal{N}(0, 1)$. The $+1$ label for each alternative $\boldsymbol{x}$ is simulated with probability $\sigma(\theta_0^* + \sum_{j=1}^{25} \theta_j^* x_j)$. Note that each alternative can be measured more than once, yielding different observations. To begin with, we randomly pre-sample 200 data points $(\boldsymbol{x}, y)$ from the true model and performed l_1 regularized logistic regression on the pre-generated batch data set. By changing the value of regularization parameter Lambda, we can obtain a hierarchical representation of the alternatives. We next examine the behavior of our knowledge gradient policy with an ensemble system of different logistic base models with different sets of explanatory variables.

Figure 5.1 illustrates the behavior of the KG policy with different feature hierarchies. The results are reported on 100 different random runs. The first row depicts the changing values of the weights on each base model, averaged over 100 different replicas. For example, in the left-most figure, we have four base models which have 5,10, 14 and 22 selected features as the explanatory variables in logistic regression, respectively. The first figure illustrates the intuitive property that the weights on the most aggregate level are highest when there are only a few observations, with a shift to the more disaggregate level as more data points are acquired. This is a very important behavior when approximating functions recursively. With a few data points, it is simply not possible to produce good function approximations, so it is reasonable to use simple functions.

At the same time, the behavior in the middle column seems contradictory to the previous observation as the most aggregate level yields the highest weights all the time. One explanation is that since the actual model has only 25 non-zero features, the more complex model is largely overfitting and does not have a good generalization

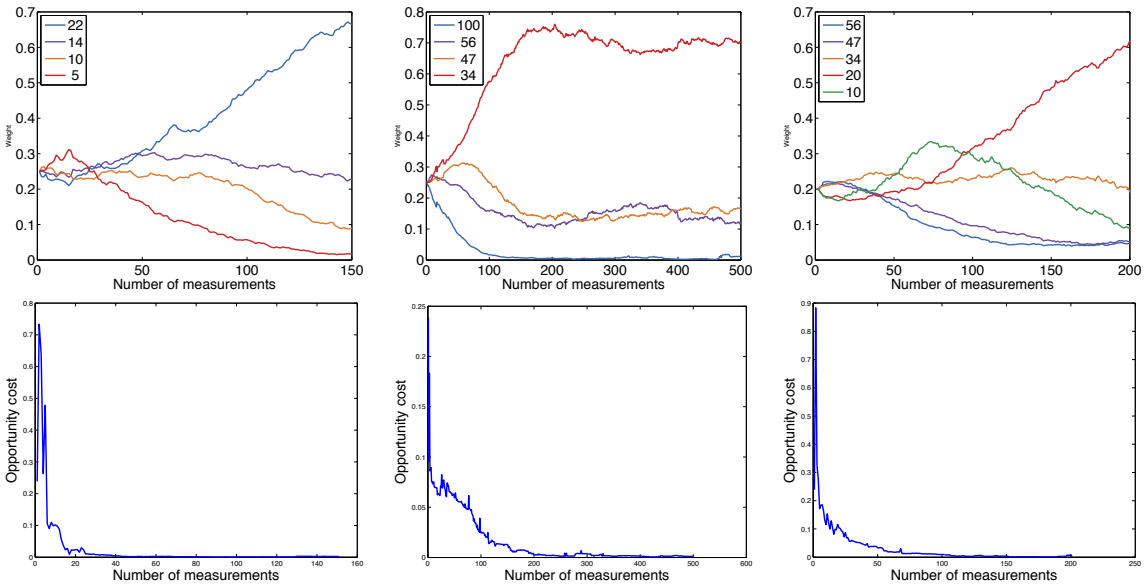


Figure 5.1: Behavior of the KG policy with feature hierarchies.

on the upcoming data points, leading to a higher loss as the algorithm processes. In the right figure where there are both models simpler than the true model and those more complex, the model that is most similar to the true model is gradually winning over others. It means that even though the experiments are guided by the KG policy rather than sampled i.i.d., it manages to identify the best base model that yields both good accuracy and generalization.

Recall that our goal is to maximize the expected response of the implementation decision. We define the Opportunity Cost (OC) metric as the expected response of the implementation decision $\boldsymbol{x}^{N+1} := \arg\max_{\boldsymbol{x}} p(y = +1|\boldsymbol{x}, S^N)$ compared to the true maximal response under weight $\boldsymbol{\theta}^*$:

$$\text{OC} := \max_{\boldsymbol{x}\in\mathcal{X}} p(y = +1|\boldsymbol{x}, \boldsymbol{\theta}^*) - p(y = +1|\boldsymbol{x}^{N+1}, \boldsymbol{\theta}^*).$$

The second row in Figure 5.1 depicts the opportunity cost of the KG policy with different feature hierarchies, averaged over 100 runs. Note that the opportunity cost is always non-negative and the smaller the better. We can see that in any case, the

OC quickly decays as the number of measurements increases, showing that the KG policy learns rapidly the location of the maximal probability of success.

5.4.2 Personalized Healthcare

In this section, we consider a real-world personalized healthcare problem. We obtained knee replacement datasets from major hospital chains in New York and New Jersey. To make a fair statement of the costs, we selected 211 episodes with 26,735 structured claim records that are obtained from the same health care provider. Each record includes age, gender, episode identifier, episode start data, claim line paid amount, diagnosis codes, procedure codes, attributed physician identifiers and so on. All the patients in the knee replacement dataset have undergone knee replacement surgery. After the knee replacement surgery, different patients have been involved in different lengths of rehabilitations and incurred a wild range of post-operative costs. We study a success/failure model where if the post-operative cost is below a Medicare specified threshold, then the episode is said to be successful. In this work, we want to understand the effect of different physicians and/or facilities on the post-operative costs, and provide guidelines on how to more effectively assign different physicians to each patient based on patient attributes.

The additional challenge in this problem is that the "success" or "failure" can be predicted through an unknown relationship that depends on a **partially** controllable vector of attributes for each instance. For example, the patient attributes are given which we do not have control over. We can only choose a medical decision, and the outcome is based on both the patient attributes and the selected medical decision. This problem is known as a contextual bandit (Langford and Zhang, 2008; Li et al., 2010; Krause and Ong, 2011). At each round n, the learner is presented with a context vector $\boldsymbol{c}_n$ (patient attributes) and a set of (vectorized) actions $\boldsymbol{a} \in \mathcal{A}$ (medical decisions). After choosing an alternative $\boldsymbol{a}$, we observe an outcome y^{n+1}. The goal

is to find a policy that selects actions such that the cumulative reward is as large as possible over time. In contrast, the previous section considers a *context-free* scenario with the goal of maximizing the probability of success (terminal reward) after the offline training phase so that the error incurred during the training is not punished.

Each of the past observations are made of triplets $(\boldsymbol{c}^n, \boldsymbol{a}^n, y^{n+1})$. We consider the success/failure model with an unknown function $\mu : \mathcal{C} \times \mathcal{A} \mapsto [0, 1]$ representing the underlying binomial probability of success of an experiment. Each base model h_k is chosen as a logistic regression where the binomial outcome $p(y = +1|\boldsymbol{c}, \boldsymbol{a})$ is predicted though $\sigma\big(F(\boldsymbol{c}_{[k]}, \boldsymbol{a})\big)$,

$$F(\boldsymbol{c}_{[k]}, \boldsymbol{a}) = \theta_0 + (\boldsymbol{\theta}^C)^T \boldsymbol{c}_{[k]} + (\boldsymbol{\theta}^A)^T \boldsymbol{a}. \tag{5.10}$$

At each round n, the model updates can be slightly modified based on the observation triplet $(\boldsymbol{c}^n, \boldsymbol{a}^n, y^{n+1})$ by treating $1||\boldsymbol{c}||\boldsymbol{a}$ as the alternative $\boldsymbol{x}$, where $\boldsymbol{u}||\boldsymbol{v}$ denotes the concatenation of the two vectors $\boldsymbol{u}$ and $\boldsymbol{v}$.

The knowledge gradient $\nu^{\text{KG}}_{\boldsymbol{a}|\boldsymbol{c}}(s)$ of measuring an action $\boldsymbol{a}$ given context $\boldsymbol{c}$ can be defined as follows:

DEFINITION 5.4.1. *The knowledge gradient of measuring an action $\boldsymbol{a}$ given a context $\boldsymbol{c}$ while in state s is*

$$\nu^{KG}_{\boldsymbol{a}|\boldsymbol{c}}(s) := \mathbb{E}\Big[V^N\Big(T\big(s, 1||\boldsymbol{c}||\boldsymbol{a}, y\big)\Big) - V^N(s)|\boldsymbol{c}, \boldsymbol{a}, s\Big]. \tag{5.11}$$

The calculation of $\nu^{\text{KG}}_{\boldsymbol{a}|\boldsymbol{c}}(s)$ can be modified based on Section 5.3.3 by replacing $\boldsymbol{x}$ as $1||\boldsymbol{c}||\boldsymbol{a}$ throughout. Since the objective in the contextual bandit problems is to maximize the cumulative number of successes, the knowledge gradient policy developed in Section 5.3.3 for stochastic binary feedback can be easily extended to online learning following the "stop-learning" (SL) policy adopted by (Ryzhov et al., 2012). The action $X^{\text{KG},n}_{\boldsymbol{c}}(s^n)$ that is chosen by KG at time n given a context $\boldsymbol{c}$ and a knowledge

state s^n can be obtained as:

$$X_{\boldsymbol{c}}^{\mathrm{KG},n}(S^n) = \arg\max_{\boldsymbol{a}} p(y = +1|\boldsymbol{c}, \boldsymbol{a}, S^n) + \tau \nu_{\boldsymbol{a}|\boldsymbol{c}}^{\mathrm{KG},n}(S^n),$$

where τ reflects a planning horizon.

After a pre-processing of feature selection, we use 161 selected patient features as the true model, out of which 54 of them are chosen by LASSO with minimum deviation via 25 fold cross-validation as non-zero features. To make a fair comparison, on each run, all the time-N labels of all the alternatives are randomly pre-generated according to the weight vector $\boldsymbol{\theta}^*$ and shared across all the competing policies. In our experiments, we have $M = 20$ different physicians and treat each physician $\boldsymbol{a}$ as an indicator variable as follows in each base model:

$$F(c, p) = \theta_0 + (\boldsymbol{\theta}^c)^T \boldsymbol{c}_{[k]} + \sum_{m=1}^{M} \theta_m^A \mathbb{I}(p, p_m).$$

For each patient, one and only one p (or f) can be assigned such that exactly one $\mathbb{I}(p, p_m)$ is 1.

We sort the patient data chronologically. For each patient visit, based on the patient attributes x, we assigned a physician for the surgery and/or a facility for the rehab. We then receive a payoff of whether it is success or failure. Evaluating an exploration/exploitation policy is difficult since we do not know the outcome for physicians and facilities that were not chosen for a particular patient in the record data. Based on the real world context and patient features in the knee replacement dataset, we instead simulate the true outcomes using a weight vector $\boldsymbol{\theta}^*$.

The experimental results are reported on 100 repetitions of each algorithm. The only difference is the way each policy selects the actions; all the rest, including the model updates, is identical as described in Section 5.3.3.

We compare our ensemble model with simple logistic models. We compare our policy (hier_KG) with pure exploitation (which assigns the physician that seems to be the best), pure exploration (which randomly assign a physician, as would happen if you assigned the first available physician) and the KG policy for a single level of logistic regression (see Section 4).

Since binary feedbacks are inherently noisy, we learn very little from a single outcome. Now consider what happens if we decide to test, say, k patients. The value of information from $k = 1$ patients may be quite low, but the value can grow nonlinearly (specifically, in the shape of an S-curve), producing much greater value if we are willing to consider the combined information learned from, say, $k = 20$ patients (Frazier and Powell, 2010). As a result of this non-concavity in the value of information, we propose to boost the performance of KG without ensembles by considering the impact of **posterior reshaping**. In our simulations, we have tried to reshape the covariance matrix $\mathbf{\Sigma}^n$ to $\eta^2\mathbf{\Sigma}^n$. This only affects the calculation of the knowledge gradient and does not change the model updates.

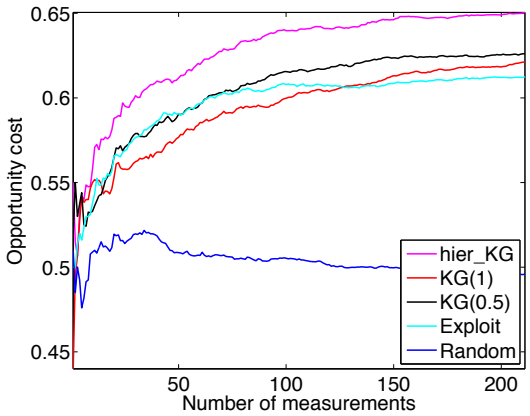


Figure 5.2: Comparison of different algorithms on the knee replacement dataset.

We report the number of cumulative success divided by the number of treated patients after each of the 211 patient visits in Fig. 5.2. We can see from the figure that the KG policy with feature hierarchies yields the best performance. The knowledge gradient policy with a posterior reshaping parameter value of 0.5 is the best among

non-hierarchical policies. Pure exploitation also does well. This seems at first a bit odd given that the system has no prior knowledge about the true parameter value $\boldsymbol{\theta}^*$. A possible explanation is that the change in context induces some level of exploration. Random assignment does not perform well since it is not learning from its past experiences while other policies use their past observations to guide the next assignment. We conclude that even though the data is sparse and high-dimensional, an ensemble optimal learning method can additionally improve success rates by around 10 percent.

5.5 Conclusion

There is variety of belief models for sequential decision making problems, e.g. look-up table model, or a specified non-parametric/parametric model. However, given the unknown structural of the underlying problem, it can be hardly decided what is the most appropriate belief model/prediction model. In this work, we purpose to develop belief models based on ensemble system, in which multiple base models can be strategically generated and combined. The primary goal is to minimize the unfortunate selection of a particularly poorly performing prediction model. Similar to the idea of online boosting, we develop Bayesian optimal learning methods with ensembles, aiming to improve the prediction of the performance of each alternative overall, so as to spend the limited measurement budget more wisely. We use logistic learners as an illustration of the base models and derive an efficient and practical algorithm for ensemble sequential decision making. Synthetic experiments and real-world experiments on a knee replacement dataset demonstrate the effectiveness of our proposed algorithms.

Chapter 6

Parallel Knowledge Gradient Method for Nested-batch Bayesian Optimization

Our work is motivated by problems in the laboratory sciences where we have to select a series of parameters (e.g. size, shape, density and concentration) that guide the design of a material where we are trying to achieve a particular goal (e.g. maximum strength, conductivity, or reflexivity). For example, in this chapter we are interested in identifying the density, size and type of nanostructures on the surface of a photoactive device that maximizes output current (see Section 6.2 for more details). The number of potential parameter settings is much larger than we can explore experimentally, especially when we consider that an experiment can take hours or even days. This is exacerbated by the complication that certain parameters may be more difficult to vary than others in a serial fashion.

There are several factors contributing to this. First is the curse of dimensionality, in which the set of potential experiments (identified by a selection of tunable parameters) increases exponentially with the number of tunable parameters. Second is the

continuous nature of certain parameters. For example, the density or concentration of a solute in solution may often be varied within several orders of magnitude, and yet the optimum selection of density could occur within a small window of values. This problem of separation of scales may be naively dealt with by using a refined discretization, which results in a large number of experimental alternatives. Third is the fact that physically, varying one parameter may be more difficult than another. For example, in our reflexivity problem (see Section 6.2), a selection of "type" and size of nanostructure (e.g. nanorods, nanodots or some other geometrical shape) entails chemically synthesizing such a structure, which may take a day in the laboratory. Contrast this with a selection of density, which can be varied more readily by an appropriate choice of solution concentration.

This sequential design of experiments considered in previous chapters fails to account for the realities encountered by experimentalists, who may be able to run several parallel experiments in batches. For example, an experimenter can easily vary nanoparticle density over a sample, effectively performing parallel, batch experiments through a single sample. While the idea of batch experimentation is well established throughout all of the physical sciences, recently new tools have provided experimentalists the ability to vary parameters such as surface feature lengths and areas on the nanometer length scale (Huo et al., 2008; Liao et al., 2013; Eichelsdoerfer et al., 2013). As a second constraint, it may be difficult or expensive for a scientist to explore the set of experiments in the order prescribed by the knowledge gradient policy, which often suggests consecutively vastly different experiments. For example, the choice of nanoparticle size described above cannot be readily changed between experiments since their synthesis is expensive. The choice of nanoparticle size and density can therefore be modeled as a nested decision in which a nanoparticle size is first selected, and several densities are chosen to maximize the marginal value of infor-

mation given the fixed nanoparticle size. Such batch and nested batch experimental modes must be taken into consideration in designing a sequence of experiments.

In this chapter, we extend the knowledge gradient concept to handle both batch experiments, as well as nested experiments that are performed within a batch. We derive the marginal value of information for each possible experiment which the scientists can use as a guide.

6.1 Literature Review

Commonly used sequential decision making policies (Auer et al., 2002; Bubeck and Cesa-Bianchi, 2012; Frazier et al., 2008; He et al., 2007) allocate only one alternative at a time and are not directly applicable to the above mentioned batch and nested batch experimental setting. More relevant to this setting, there exists literature on stochastic and/or adversarial bandit problems addressing the problem of multi-plays (playing several alternatives at the same time), which can be viewed as batch-mode decision making (Agrawal et al., 1990; Audibert et al., 2013; Uchiya et al., 2010; Kale et al., 2010; Cesa-Bianchi and Lugosi, 2012; Radlinski et al., 2008; Gopalan et al., 2014). However, the bandit objective is to maximize the cumulative rewards over time, which is not suitable for our laboratory setting where the objective is to find the controllable parameters that maximizes some utility function. Chen and Krause (2013) have studied batch mode active learning and more general information-parallel stochastic optimization problems. But their objective is to let a set function exceed a threshold value while at the same time minimizing the number of items allocated. Moreover, the proposed algorithm in Chen and Krause (2013) is specifically designed for batch-mode active learning and cannot be generalized to other information-parallel stochastic optimization problems.

The most related models are the stochastic subset selection problems introduced in Ryzhov and Powell (2009a,b), where the choice in each round is a subset of alternatives while the objective is to find the set of alternatives that maximizes some function on such sets. This differs fundamentally from finding one alternative that maximizes some utility function through batch measurements, as is the case in our setting. In Ryzhov and Powell (2009a,b), the way to recommend a set of alternatives in each round is to treat each subset of alternatives as a singly super alternative in the space of subsets and construct beliefs over the set function values rather than the function values of the alternatives. The number of subsets with B elements out of M elements is $\binom{M}{B}$, which grows exponentially with the number of alternatives. As the number of alternatives increases, even storing and updating the requisite $\binom{M}{B} \times \binom{M}{B}$ covariance matrix becomes problematic. For example, the size of the choice set considered in Ryzhov and Powell (2009b) was $\binom{10}{5} = 252$. Instead, to address this, we derive the policies presented in this paper with beliefs on the function values of the alternatives whose number is far smaller than the super alternatives.

6.2 Motivating Application

As a motivating example and as discussed briefly above, we consider a photoactive device in which anisotropic gold nanoparticles (NPs) are immobilized on the surface of the device. The immobilization is performed using a DNA-mediated approach using thiol-gold chemistry in which both the surface and the NPs are functionalized with complimentary DNA strands that subsequently bind to hold the particles onto surface (Senesi et al., 2013). The NP's role is to enhance the photocurrent of the device via photonic and plasmonic phenomena, thereby potentially increasing the photoelectric efficiency of such a device. Understanding the particular configuration of the device

that yields optimal photoactivity is desirable in applications such as efficient solar cells.

Among the tunable parameters that describe the device's configuration are NP size and the density of NPs functionalized onto the surface of the device. Synthesis of NPs of a particular size is done via solution-phase chemistry, and requires several hours to days to complete (Millstone et al., 2009; Langille et al., 2012). In contrast, once NPs are synthesized, it is straightforward to immobilize them onto the device at some prescribed density (Senesi et al., 2013; Park et al., 2008), and often several such densities can be considered in parallel. Therefore, in selecting which configurations (i.e. a choice of NP size and density) to experimentally test, we are naturally led to a nested, batch decision. Different densities may be run in a batch setting, provided that the size of the NPs is the same within the batch.

While the exact mapping between the tunable parameters of NP size and density and the response output current is not well established, we may make some qualitative statements using domain expert prior knowledge. Specifically, both NP size and density affect the phenomena of surface plasmon resonance, photon absorption, and scattering, which subsequently influence output current. The full description of the effect of the parameters on these physical phenomena and output current is beyond the scope of this paper (see e.g. Djurii and Leung (2006); Gehr and Boyd (1996); Flory and Berginc (2011)) for a general treatment of discussions on optical and electrical properties of nanostructured devices). However, we state that due to competing effects, there exists a critical value of both NP size d and the logarithm of NP density ρ that optimizes output current $I(d, \rho)$, and further assume that there exists a single such extrema in the domain of interest. Our task is to find this critical value under uncertainty of the true physics of the system. In what follows, we describe the technique employed to adaptively and iteratively select those configurations to test in order to maximize output current in a nested batch setting.

6.3 From Sequential Decision Making to Nested-Batch-Mode Decision Making

We first recall the definition of the ranking and selection (R&S) problem introduced in Section 2.1.1. For correlated normal beliefs, by Bayes' rule and the Sherman-Morrison formula, the update equations (see Eq. (2.3) and (2.4)) can be written as

$$\theta^{n+1} = \theta^n + \frac{W^{n+1} - \theta^n_x}{\lambda^W + \Sigma^n_{xx}} \Sigma^n e_x, \tag{6.1}$$

$$\Sigma^{n+1} = \Sigma^n - \frac{\Sigma^n e_x (e_x)^T \Sigma^n}{\lambda^W + \Sigma^n_{xx}}, \tag{6.2}$$

where e_x is a vector with 1 at index x and zeros everywhere else.

Alternatively, we can formulate the problem within a dynamic programming framework (Frazier et al., 2009). Define the state space $\mathcal{S}$ to be the cross-product of $\mathbb{R}^M$ and the space of positive semi-definite matrices. We next define the transition function from the updating equations (6.1) (6.2). Define a vector valued function $\tilde{\sigma}$ as

$$\tilde{\sigma}(\Sigma, x) = \frac{\Sigma e_x}{\sqrt{\lambda^W + \Sigma_{xx}}}, \tag{6.3}$$

where Σ is any covariance matrix. Next define the random variable

$$Z^{n+1} = \frac{W^{n+1}_{x^n} - \theta^n_{x^n}}{\sqrt{\mathrm{Var}[W^{n+1}_{x^n} - \theta^n_{x^n} | \mathcal{F}^n]}},$$

which is a one-dimensional standard normal random variable when conditioned on $\mathcal{F}^n$.

We can write (6.1) as

$$\theta^{n+1} = \theta^n + \tilde{\sigma}(\Sigma^n, x^n) Z^{n+1}. \tag{6.4}$$

Update (6.2) can also be rewritten as

$$\Sigma^{n+1} = \Sigma^n - \tilde{\sigma}(\Sigma^n, x^n)\big(\tilde{\sigma}(\Sigma^n, x^n)\big)^{\mathrm{T}}. \tag{6.5}$$

Now we can define the transition function.

DEFINITION 6.3.1. *The transition function* $T : \mathcal{S} \times \mathcal{X} \times \mathbb{R}$ *is defined as*

$$T\bigg((\theta, \Sigma), x, z\bigg) := \bigg(\theta + \tilde{\sigma}(\Sigma, x)z, \Sigma - \tilde{\sigma}(\Sigma, x)\big(\tilde{\sigma}(\Sigma, x)\big)^{\mathrm{T}}\bigg), \tag{6.6}$$

so that $S^{n+1} = T(S^n, x^n, Z^{n+1})$. *Here* θ *is a vector,* Σ *is a covariance matrix,* $z \in \mathbb{R}$ *and* Z^{n+1} *is a one-dimensional standard normal random variable.*

In what follows, we will first give the formal model for batch learning and then we will extend it to nested batch mode decision making.

6.3.1 Batch Mode Learning Model

In real world applications, it often occurs that information collectors do not simply take one measurement at a time. For example, in a pharmaceutical company, researchers might test the efficiency of a medicine by taking measurements of five different concentrations simultaneously, observing all the outcomes, and then measuring the next five concentrations. Or in the motivating application, if we fix a NP size, then we can test on different densities simultaneously. This leads us to the idea of batch measurements.

Suppose we have a collection $\mathcal{X} = \{1, 2, ..., M\}$ of M alternatives. Instead of sequentially measuring some alternatives to estimate the constant but unknown underlying mean μ_x, we can measure a batch of alternatives simultaneously at each step. We begin with a prior multivariate normal distribution of belief about the performance μ_x for each alternative $x \in \mathcal{X}$, $\mu \sim \mathcal{N}(\theta^0, \Sigma^0)$, where $\mu = (\mu_x)_{x\in\mathcal{X}}, \theta^0 = (\theta^0_x)_{x\in\mathcal{X}}$ and

Σ^0 is the covariance in our belief about the alternatives. Denote the batch size by B and the total number of batches by K. Then the total number of measurements allowed is $N = BK$. At the kth batch (starting with $n = 0$), instead of choosing one alternative to measure as in Section 2.1.1, we choose to measure B alternatives $x^{k,0}, x^{k,1}, ..., x^{k,B-1}$. Let ϵ^{k+1} be the measurement error which is assumed to be normally distributed with known variance $\lambda^W = \sigma_W^2$. The resulting observations are $W^{k+1,0} \sim \mathcal{N}(\mu_{x^{k,0}}, \sigma_W), W^{k+1,1} \sim \mathcal{N}(\mu_{x^{k,1}}, \sigma_W), ..., W^{k+1,B-1} \sim \mathcal{N}(\mu_{x^{k,B-1}}, \sigma_W)$.

We modify our notations to fit batch measurements. The superscript (k, b) for some $k = 0, 1, ..., K-1$ and $b = 1, 2, ..., B-1$ should be understood as meaning that we have done k batches and use $x^{k,0}, ..., x^{k,b-1}, W^{k+1,0}, W^{k+1,...,b-1}$ to update our belief. Thus the prior multivariate normal belief can be rewritten as $(\theta^{0,0}, \Sigma^{0,0})$. The new updating equations can be written as

$$\theta^{k,b+1} = \theta^{k,0} + \sum_{j=0}^{b} \frac{W^{k+1,j} - \theta^{k,j}_{x^{k,j}}}{\lambda^W + \Sigma^{k,j}_{x^{k,j}x^{k,j}}} \Sigma^{k,j} e_{x^{k,j}}, \tag{6.7}$$

$$\Sigma^{k,b+1} = \Sigma^{k,b} - \frac{\Sigma^{k,b} e_{x^{k,b}} (e_{x^{k,b}})^T \Sigma^{k,b}}{\lambda^W + \Sigma^{k,b}_{x^{k,b}x^{k,b}}}, \tag{6.8}$$

where $k = 0, 1, ..., K-1$, $b = 0, 1, ..., B-1$, $\theta^{k+1,0} = \theta^{k,B}$ and $\Sigma^{k+1,0} = \Sigma^{k,B}$. It is worth emphasizing that in the batch setting the covariance matrix would be updated within a batch since it is determined by the measurement decisions and is independent of the observations, whereas the mean values θ^n are only updated after the observations are collected for the whole batch. Additionally, the updating formula (6.7) is not affected by whether the observations are obtained sequentially or in batch.

A decision function $X^\pi(S^n)$ is defined as a mapping from the knowledge states to $\mathcal{X}^B$, where S^n is short for $S^{n,0} = (\theta^{n,0}, \Sigma^{n,0})$.

If we are limited to $N = KB$ measurements, the objective is to maximize the expected reward of the final recommended alternative:

$$\max_{\pi \in \Pi} \mathbb{E}\left[\mu_{x^K}\right], \tag{6.9}$$

where $x^K = \arg\max_{x \in \mathcal{X}} \theta_x^K$ and $\{x^{k,0}, ..., x^{k,B-1}\} = X^\pi(S^k)$ for $k = 0, 1, ..., K-1$.

We can also formulate the problem within a dynamic programming framework. We first define the transition function from the updating equations.

For convenience, we introduce the σ-algebras $\mathcal{F}^{k,b}$ for any $b = 0, 1, ..., B-1$ which is formed by the previous k batch measurement outcomes and the first b observations in the current batch. The idea is that even when performing experiments in batch, we can model the updating as if each outcome is collected sequentially. Suppose we are at the $k+1$th batch and have made the measurement decisions for the whole batch. For any $b = 0, 1, ..., B-1$, define the random variable $Z^{k+1,b}$ as

$$Z^{k+1,b} := \frac{W^{k+1,b} - \theta^{k,b}_{x^{k,b}}}{\sqrt{\mathrm{Var}[W^{k+1,b} - \theta^{k,b}_{x^{k,b}} | \mathcal{F}^{k,b}]}}.$$

Since $\theta_x^{k,b} \in \mathcal{F}^{k,b}$,

$$\mathrm{Var}[W^{k+1,b} - \theta^{k,b}_{x^{k,b}} | \mathcal{F}^{k,b}] = \mathrm{Var}[\mu_{x^{k,b}} + \epsilon^{k+1} | \mathcal{F}^{k,b}] = \Sigma^{k,b}_{x^{k,b}x^{k,b}} + \lambda^W.$$

It is important to note that if conditioned on $Z^{k+1,0}, ..., Z^{k+1,b-1}$, or in other words, $\mathcal{F}^{k,b}$, the $Z^{k+1,b}$ is a standard normal distribution.

Recall the definition of a vector valued function $\tilde{\sigma}$ as

$$\tilde{\sigma}(\Sigma, x) = \frac{\Sigma e_x}{\sqrt{\lambda^W + \Sigma_{xx}}}, \tag{6.10}$$

where Σ is any covariance matrix. We can rewrite (6.7) and (6.8) as

$$\begin{aligned} \theta^{k,b+1} &= \theta^{k,0} + \sum_{j=0}^{b} \tilde{\sigma}(\Sigma^{k,j}, x^{k,j}) Z^{k+1,j}, \qquad (6.11) \\ \Sigma^{k,b+1} &= \Sigma^{k,b} - \tilde{\sigma}(\Sigma^{k,b}, x^{k,b}) \big(\tilde{\sigma}(\Sigma^{k,b}, x^{k,b})\big)^{\mathrm{T}}. \qquad (6.12) \end{aligned}$$

Now we can define the transition function for batch mode learning recursively by pretending the outcomes are obtained sequentially.

DEFINITION 6.3.2. *The transition function $T^B : \mathcal{S} \times \mathcal{X}^B \times \mathbb{R}^B$ is defined as*

$$T^B\bigg((\theta, \Sigma), (x_1, ..., x_B), (z_1, ..., z_B)\bigg) := T(...T((\theta, \Sigma), x_1, z_1), .., x_B, z_B), \qquad (6.13)$$

so that $S^{k+1,0} = T^B\big(S^{k,0}, (x^{k,0}, ..., x^{k,B-1}), (Z^{k+1,0}, ..., Z^{k+1,B-1})\big)$. Here θ is a vector, Σ is a covariance matrix, $z^{k+1,j} \in \mathbb{R}$, $Z^{k+1,j}$ is a one-dimensional standard normal random variable and T is the transition function defined in Definition 6.3.1.

We then define the value function $V^{B,k} : \mathcal{S} \mapsto \mathbb{R}$ after k batch measurements at times $k = 0, 1, ..., K-1$ as

$$V^{B,k}(s) := \max_{\pi} \mathbb{E}^{\pi}\big[\max_{x} \theta_x^K | S^k = s\big], \forall s \in \mathcal{S}.$$

By noting that θ^K is deterministic given S^K, the terminal value function $V^{B,K}$ can be computed directly as:

$$V^{B,K}(s) = \max_{x \in \mathcal{X}} \theta_x, \forall s = (\theta, \Sigma) \in \mathcal{S}. \qquad (6.14)$$

The dynamic programming principle tells us that the value function at times $k = 0, 1, ..., K-1$, $V^{B,k}$ is given recursively by :

$$V^{B,k}(s) = \max_{(x_i)_{i=1}^B \in \mathcal{X}^B} \mathbb{E}\big[V^{B,k+1}\big(T^{\mathrm{B}}(s, (x_i)_{i=1}^B, (Z_i)_{i=1}^B\big)\big], s \in \mathcal{S}, \tag{6.15}$$

where Z_i is a one dimensional standard normal variable.

A Knowledge-Gradient policy is provided for batch learning model in section 6.4.

6.3.2 Nested Batch Mode Learning Model

Motivated by the applications given by the real world applications in Section 6.2, right now we have a collection $\mathcal{X}_1 \times \mathcal{X}_2$ of M alternatives, where at each decision step, we choose one $x \in \mathcal{X}_1$ and a set $\mathcal{Y} \in \mathcal{X}_2^B$, constructing B alternatives to measure simultaneously (e.g. design 50nm triangle particles and experiment with densities of 3%, 10%, 27%, 78% and 92% with a batch size $B = 5$).

As before, we begin with a prior multivariate normal distribution of belief about the performance $\mu_{(x,y)}$ for each alternative $x \in \mathcal{X}_1$ and $y \in \mathcal{X}_2$, $\mu \sim \mathcal{N}(\theta^0, \Sigma^0)$, where $\mu = (\mu_{(x,y)})_{(x,y)\in\mathcal{X}_1\times\mathcal{X}_2}$, $\theta^0 = (\theta^0_{(x,y)})_{(x,y)\in\mathcal{X}_1\times\mathcal{X}_2}$ and Σ^0 is a $M \times M$ covariance matrix.

Let K be the total number of batches. At any decision step $k = 0, 1, ..., K-1$ after we make the B measurement decisions $(x^k, y^{k,0}), (x^k, y^{k,1}), ...(x^k, y^{k,B-1})$ and get their outcomes, we can also pretend that the information is collected sequentially. So the updating equations are the same as those in the batch mode model when treating (x, y) as the alternative and replacing $x^{k,j}$ with $(x^k, y^{k,j})$. It is worth noting here, we are not only updating our belief about the alternatives with x^k, but we are also updating our belief about all M alternatives.

A decision function $X^\pi(S^n)$ is defined as a mapping from the knowledge state to $\mathcal{X}_1 \times \mathcal{X}_2^B$. The objective is to maximize the expected reward of the final recommended

alternative:

$$\max_{\pi\in\Pi} \mathbb{E}\left[\mu_{(x^K,y^K)}\right], \tag{6.16}$$

where $(x^K, y^K) = \arg\max_{(x,y)\in\mathcal{X}_1\times\mathcal{X}_2} \theta^K_{(x,y)}$ and $\{x^k, y^{k,0}, ..., y^{k,B-1}\} = X^\pi(S^k)$ for $k = 0, 1, ..., K-1$.

We formulate the problem within a dynamic programming framework. By a similar argument as that in batch mode, we can define the transition function as

DEFINITION 6.3.3. *Define the transition function* $T^{NB} : \mathcal{S} \times (\mathcal{X}_1 \times \mathcal{X}_2^B) \times \mathbb{R}^B$ *as*

$$T^{NB}\Big((\theta,\Sigma),(x,y_1,...,y_B),(z_1,...,z_B)\Big) := T\big(...T\big((\theta,\Sigma),(x,y_1),z_1\big),..,(x,y_B),z_B\big), \tag{6.17}$$

so that $S^{k+1} = T^{NB}\big(S^k,(x^k,y^{k,0},...,y^{k,B-1}),(Z^{k+1,0},...,Z^{k+1,B-1})\big)$. *Here* θ *is a vector,* Σ *is a covariance matrix,* $z^{k+1,j} \in \mathbb{R}$, $Z^{k+1,j}$ *is a one-dimensional standard normal random variable and* T *is the transition function defined in Definition 6.3.1.*

We then define the value function $V^{NB,k} : \mathcal{S} \mapsto \mathbb{R}$ after k nested batch measurements at times $k = 0, 1, ..., K-1$ as

$$V^{NB,k}(s) := \max_\pi \mathbb{E}^\pi\big[\max_{(x,y)} \theta^K_{(x,y)} | S^k = s\big], \forall s \in \mathcal{S}.$$

The terminal value function $V^{NB,K}$ can be computed directly as:

$$V^{NB,K}(s) = \max_{(x,y)\in\mathcal{X}_1\times\mathcal{X}_2} \theta_{(x,y)}, \forall s = (\theta,\Sigma) \in \mathcal{S}. \tag{6.18}$$

The dynamic programming principle tells us that the value function at times $k = 0, 1, ..., K-1$, $V^{B,k}$ is given recursively by :

$$V^{NB,k}(s) = \max_{(x,\mathcal{Y})\in\mathcal{X}_1\times\mathcal{X}_2^B} \mathbb{E}\big[V^{NB,k+1}\big(T^{\mathrm{NB}}(s,(x,y_1,...y_B),(Z_1,...Z_B))\big)\big], s \in \mathcal{S}, \tag{6.19}$$

where Z_i is a one dimensional standard normal variable.

A KG-type policy is provided for nested batch learning in the section 6.5.

6.4 Batch Knowledge Gradient (BKG) Policy

In this section, we extend the original idea of the KG policy for batch mode learning. We first give the formal definition of the batch knowledge gradient policy and then provide a Monte Carlo algorithm for any given batch size.

6.4.1 Definition of BKG Policy

Following the basic idea of the knowledge gradient, we would like to design a policy that seeks to measure the B alternatives that provide the single-period expected increment as a batch. We first define the value of information from measuring a batch of alternatives.

DEFINITION 6.4.1. *The knowledge gradient for measuring a batch of j alternatives $\{x_1, ...x_j\}$ at state s is defined as*

$$\nu^{BKG}_{x_1,...x_j}(s) := \mathbb{E}\left[V^{B,K}\big(T^B(s,(x_1,...x_j),(Z_1,...,Z_j))\big) - V^{B,K}(s)\right], \tag{6.20}$$

where Z_i is a one-dimensional standard normal random variable.

Recall from (6.14) that $V^{B,K}(S^k) = \max_{x\in\mathcal{X}} \theta^k_x$. Thus, suppose we are in knowledge state $S^k = (\theta^k, \Sigma^k) = (\theta^{k,0}, \Sigma^{k,0})$. If we choose to measure $(x^{k,0} = x_1, ..., x^{k,j-1} = x_j)$ right now, allowing us to observe $(W^{k+1,0}_{x^{k,0}}, ..., W^{k+1,j-1}_{x^{k,j-1}})$, then we transition to a new state of knowledge $S^{n+1} = (\theta^{n+1}, \Sigma^{n+1})$. At iteration k, θ^{k+1} is a random vector since we do not yet know what W^{k+1} is going to be. The knowledge gradient of

measuring $(x_1, ..., x_j)$ is then

$$\nu^{\text{BKG}}_{x_1,...x_j}(S^k) = \mathbb{E}[\max_x \theta^{k+1}_x - \max_x \theta^k_x | x^{k,0} = x_1, \cdots, x^{k,j-1} = x_j, S^k]. \tag{6.21}$$

One way to design a policy π' using the knowledge gradient concept is to directly find the $\{x_1, ..., x_j\}$ that maximizes $\nu^{\text{BKG}}_{x_1,...x_j}(S^k)$ subject to $j \leq B$. Since the measurement is noisy, measuring the same x_i several times will most likely give different observations and thus it is meaningful if we measure some alternative x_i more than once within a batch. For example, in the motivating application, we can choose to test on 5 densities $(\rho_1, \rho_1, \rho_3, \rho_3, \rho_7)$ all at once. Thus the batch decision procedure is analogous to multi-set function maximization problems. Let $\mathcal{X}$ be a finite set of M elements. Define the multi-set function $f : \mathbb{N}^{\mathcal{X}} \mapsto \mathbb{R}$. The problem is to find a multi-set A of cardinality less than or equal to some specified number B, such that $f(A)$ is the maximum:

$$\max_{A \subset \mathbb{N}^{\mathcal{X}}} \{f(A) : |A| \leq B\}. \tag{6.22}$$

The problem with π' is that it involves testing all $\sum_{b=0}^{B-1} \binom{b+M-1}{M-1}$ which would be computationally costly when B and M are large. Alternatively, as a common technique to deal with set function maximization problems, we can use a greedy heuristic to start from the null set and add elements one at a time. We first claim that the more measurements, the larger the value of information. Thus, if we are limited to B measurements in a batch, we will indeed measure B alternatives in each batch.

PROPOSITION 6.4.1. **(Benefits of Measurement)**

$\nu^{BKG}_{x_1,...x_{j+1}}(s) \geq \nu^{BKG}_{x_1,...x_j}(s)$ for all $j \geq 0$, $s \in \mathcal{S}$ and $x_i \in \mathcal{X}$.

Proof. In the following proof, we use properties of conditional expectations $\mathbb{E}[\mathbb{E}[U|V]] = \mathbb{E}[U]$ for any random variables U and V.

$$\begin{aligned}
& \nu^{\mathrm{BKG}}_{x_1,\ldots x_{j+1}}(s) - \nu^{\mathrm{BKG}}_{x_1,\ldots x_j}(s) \\
= \;& \mathbb{E}\left[V^{B,K}\left(T^{\mathrm{B}}(s,(x_i)_{i=1}^{j+1},(Z_i)_{i=1}^{j+1})\right) - V^{B,K}\left(T^{\mathrm{B}}(s,(x_i)_{i=1}^{j},(Z_i)_{i=1}^{j})\right)\right] \\
= \;& \mathbb{E}\left[\mathbb{E}\left[V^{B,K}\left(T^{\mathrm{B}}(s,(x_i)_{i=1}^{j+1},(Z_i)_{i=1}^{j+1})\right) - V^{B,K}\left(T^{\mathrm{B}}(s,(x_i)_{i=1}^{j},(Z_i)_{i=1}^{j})\right)|(x_i)_{i=1}^{j},(z_i)_{i=1}^{j}\right]\right] \\
= \;& \mathbb{E}\left[\mathbb{E}\left[V^{B,K}\left(T(s',x_{j+1},Z_{j+1})\right) - V^{B,K}(s')\right]\right],
\end{aligned}$$

where $s' = T^{\mathrm{B}}\left(s,(x_i)_{i=1}^{j},(z_i)_{i=1}^{j}\right)$, $T(s,x,z)$ is the transition function defined in Definition 6.3.1 and in the last equation the first expectation is taken over the random choices of s' or equivalently the choices of $(z_i)_{i=1}^{j}$ and the second expectation is taken over Z_{j+1}. By the definition of T and $V^{B,K}$, we have $V^{B,K}\left(T(s',x_{j+1},Z_{j+1})\right) = \max_{x\in\mathcal{X}}\left(\theta'_x + \tilde{\sigma}_x(\Sigma',x_{j+1})Z_{j+1}\right)$ and $V^{B,K}(s') = \max_x \theta'_x$. By Jensen's inequality, we have

$$\begin{aligned}
\mathbb{E}\left[V^{B,K}\left(T(s',x_{j+1},Z_{j+1})\right)\right] &= \mathbb{E}\left[\max_{x\in\mathcal{X}}\left(\theta'_x + \tilde{\sigma}_x(\Sigma',x_{j+1})Z_{j+1}\right)\right] \\
&\geq \max_{x\in\mathcal{X}}\mathbb{E}\left[\left(\theta'_x + \tilde{\sigma}_x(\Sigma',x_{j+1})Z_{j+1}\right)\right] \\
&= \max_{x\in\mathcal{X}}\theta'_x \\
&= V^{B,K}(s').
\end{aligned}$$

Since this inequality holds for any realization of s', the proposition follows. □

Corollary 6.4.2. *The knowledge gradient of measuring a batch of j alternatives at any state s is always non-negative, $\nu^{BKG}_{x_1,\ldots x_j}(s)$ for all $j \geq 0$, $s \in \mathcal{S}$ and $x_i \in \mathcal{X}$.*

Proof. It follows from Proposition 6.4.1 by noting that $\nu^{\mathrm{BKG}}_{\emptyset}(s) = 0$ for any $s \in \mathcal{S}$. □

Since the more measurements the better, if we are limited to at most B measurements at each time step, we will exactly choose to make B measurements. We thus can define the batch knowledge gradient (BKG) policy that greedily adds in each alternative that maximizes the expected increment of value one at a time until B alternatives are chosen.

DEFINITION 6.4.3. *The Batch Knowledge Gradient (BKG) policy has the decision function*

$$x^{k,b} := X_b^{BKG}(S^k) = \arg\max_{x\in\mathcal{X}} \nu^{BKG}_{x^{k,0},\ldots,x^{k,b-1},x^{k,b}=x}(S^k), \tag{6.23}$$

for any $b = 0, ..., B-1$ *and decision points* $k = 0, 1, ..., K-1$.

The above formulation tells us that we make each measurement decision in the batch by conditioning on the earlier decisions made in the same batch and the state. With (6.11) and (6.21), we can rewrite (6.23) as

$$X_b^{\mathrm{BKG}}(S^k) = \arg\max_{x\in\mathcal{X}} \mathbb{E}\left[\max_{x'}\left(\theta^{k,0} + \sum_{j=0}^{b-1}\tilde{\sigma}(\Sigma^{k,j}, x^{k,j})Z^{k+1,j} + \tilde{\sigma}(\Sigma^{k,b}, x)Z^{k+1,b}\right)\right], \tag{6.24}$$

where $x^{k,j}$, $j \le b$ are fixed when choosing $x^{k,b}$ and $\Sigma^{k,j}$ can be updated within a batch according to (6.8). This formula will be of use in the following computations.

6.4.2 Computation

We notice from (6.23) that at each batch decision point k, we can find the first measurement decision explicitly by carrying out the original KG calculation described since the objective function (6.24) to be maximized for the first decision in the batch is exactly the same as the sequential knowledge gradient policy that was described in Section 2.1.2.

Since an analytic expression for the expected maximization as in (6.24) is unknown, we utilize Monte Carlo sampling to approximate the expectation. After the

first measurement decision $x^{k,0}$ is made, the following decisions are made one at a time to find $x^{k,b}$ according to (6.24) using Monte Carlo Simulation. To be more specific, the second decision is made by randomly generating both $Z^{k+1,0}$ and $Z^{k+1,1}$ for Q times, where $Z^{k+1,i}$ are independent standard normal variables. We then define the second decision $x^{k,1}$ as:

$$\arg\max_{x\in\mathcal{X}} \frac{1}{Q}\sum_{q=1}^{Q}\left[\max_{x'}\left(\theta^{k,0}+\tilde{\sigma}(\Sigma^{k,0},x^{k,0})z_q^0+\tilde{\sigma}(\Sigma^{k,1},x)z_q^1\right)\right],$$

where z_q^0 and z_q^1 are realizations of $Z^{k+1,0}$ and $Z^{k+1,1}$ respectively and $\Sigma^{k,1}$ is updated according to (6.8). We then have $x^{k,0}$ and $x^{k,1}$ fixed, and proceed to find $x^{k,2}$ similarly by sampling $Z^{k+1,0}$, $Z^{k+1,1}$ and $Z^{k+1,2}$ for Q times and finding the alternative that maximizes the analogous expression coming from (6.24) that contains these three random variables. It is worth re-emphasizing here that all three variables are standard normal when we generate their realizations after fixing the previous decisions.

In general, after we get the first b decisions within a batch, we are looking to find the solution to

$$x^{k,b}=\arg\max_{x\in\mathcal{X}} \frac{1}{Q}\sum_{q=1}^{Q}\left[\max_{x'}\left(\theta^{k,0}+\sum_{j=0}^{b-1}\tilde{\sigma}(\Sigma^{k,j},x^{k,j})z_q^j+\tilde{\sigma}(\Sigma^{k,b},x)z_q^b\right)\right],$$

where $\Sigma^{k,j}$ are updated within this batch according to (6.8).

The pseudo-code of the algorithms are presented below. Algorithm 7 is the BKG policy for the kth batch decision, which calls Algorithm 8 to find the next measurement decision for B times.

Algorithm 7: Batch Knowledge Gradient Policy

input : $\theta^{k,0}, \Sigma^{k,0}$ and the number of sample Q for the Monte Carlo simulation

Use the sequential KG policy presented in Section 2.1.2 to find $x^{k,0}$;

$\tilde{\sigma}^0 \leftarrow \tilde{\sigma}(\Sigma^{k,0}, x^{k,0})$;

Update $\Sigma^{k,1}$ according to (6.8);

for $b = 1$ *to* $B-1$ **do**

  Use Algorithm 8 below to find $x^{k,b}$;

  $\tilde{\sigma}^b \leftarrow \tilde{\sigma}(\Sigma^{k,b}, x^{k,b})$;

  Update $\Sigma^{k,b+1}$ according to (6.8);

end

output: batch decisions $x^{k,0}, x^{k,1}, ..., x^{k,B-1}$

Algorithm 8: Monte Carlo Simulation for the $(b+1)$th decision within a batch

input : b, $\theta^{k,0}, \tilde{\sigma}^0, \tilde{\sigma}^1, ..., \tilde{\sigma}^{b-1}, \Sigma^{k,b}$ and Q

for *each* $x \in \mathcal{X}$ **do**

  $\text{sum}_x = 0$;

  for $q = 1$ *to* Q **do**

    for $j = 0$ *to* b **do**

      Generate a realization z_q^j of $Z^{k,j}$;

    end

    $\text{temp} \leftarrow \max_{x'} \left(\theta_{x'}^{k,0} + \sum_{j=0}^{b-1} \tilde{\sigma}_{x'}^j z_q^j + \tilde{\sigma}(\Sigma^{k,b}, x) z_q^b\right)$;

    $\text{sum}_x \leftarrow \text{sum}_x + \text{temp}$;

  end

end

$x^{k,b} = \arg\max_{x \in \mathcal{X}} \text{sum}_x$;

output: the $b+1$th decision $x^{k,b}$ within the batch

6.5 Nested Batch Knowledge Gradient (NBKG) Policy

A nested batch decision may involve the selection of a particular NP size and subsequent selection of several NP densities (given the NP size fixed in the first stage of the decision). In general, we would like to design a policy that seeks to measure the B alternatives $(x, y_1), ..., (x, y_B)$ that provide the largest single period value of information. We first define the knowledge gradient of measuring a nested batch of alternatives.

DEFINITION 6.5.1. *The knowledge gradient of measuring a nested batch of j alternatives $\{(x, y_1), ..., (x, y_j)\}$ for any $x \in \mathcal{X}_1$ and $y_i \in \mathcal{X}_2$ at state s is defined as*

$$\nu^{NBKG}_{x;y_1,...y_j}(s) := \mathbb{E}\left[V^{NB,K}\big(T^{NB}(s, (x, y_1, ...y_j), (Z_1, ..., Z_j))\big) - V^{NB,K}(s)\right], \quad (6.25)$$

where Z_i is a one-dimensional standard normal random variable.

By a similar argument as Proposition 6.4.1, we can show that if we are limited to B measurements in a batch we will indeed evaluate B alternatives.

We define the Nested Batch Knowledge Gradient policy as directly finding out $\{x, y_1, ..., y_B\}$ that maximizes $\nu^{\text{NBKG}}_{x;y_1,...y_j}(s)$ at any decision point $k = 0, 1, ..., K$. For clarity, we use $\mathcal{Y}$ to denote the multi-set $\{y_1, ..., y_B\}$ since the alternatives being measured in each batch are not necessarily distinct.

DEFINITION 6.5.2. *The Nested Batch Knowledge Gradient (NBKG) policy has the decision function*

$$X^{NBKG}(S^k) = \arg\max_{(x,\mathcal{Y})} \nu^{NBKG}_{x;\mathcal{Y}}(S^k), \quad (6.26)$$

for any decision points $k = 0, 1, ..., K - 1$.

We can show analytically that

$$\begin{cases} x^* & = \arg\max_x \left(\max_{\mathcal{Y}} \nu^{\text{NBKG}}_{x;\mathcal{Y}} \right) \\ \mathcal{Y}^* & = \arg\max_{\mathcal{Y}} \nu^{\text{NBKG}}_{x^*;\mathcal{Y}} \end{cases}$$

is a solution to the optimization problem (6.26). This gives us a two-stage decision process. At the first step, for each $x \in \mathcal{X}_1$, find the multi-set (a batch) $\mathcal{Y}_x$ that gives the most value of information; i.e. $\max_{\mathcal{Y}} \nu^{\text{NBKG}}_{x;\mathcal{Y}}$. This can be done by using the Batch Knowledge Gradient policy for each fixed x with the value function $\nu^{\text{NBKG}}_{x;y_1,\ldots,y_B}$ instead of ν^{BKG}. Namely, for example, when calculating a similar expression as (6.21):

$$\nu^{\text{NBKG}}_{x;y_1,\ldots y_j}(S^k) = \mathbb{E}[\max_{(x',y')} \theta^{n+1}_{(x',y')} - \max_{(x',y')} \theta^n_{(x',y')} | x^k = x, y^{k,0} = y_1, \cdots, y^{k,j-1} = y_j, S^k], \tag{6.27}$$

it should be noted that even though the BKG is constructed for each $x \in \mathcal{X}_1$, when taking the maximization inside the expectation, x', y' should include all the choices in the domain $\mathcal{X}_1 \times \mathcal{X}_2$. Since calculating the expected maximum is needed to make the decision, Monte Carlo sampling is used as in Algorithm 7 to approximate the expectation.

We next define the nested knowledge gradient ν^{NKG}_x for each $x \in \mathcal{X}_1$ at state s in the nested dimensions as

$$\nu^{\text{NKG}}_x(s) = \max_{\mathcal{Y}} \nu^{\text{NBKG}}_{x;\mathcal{Y}}(s). \tag{6.28}$$

If the set function maximization problem and the expectation of maximization can be solved exactly (without using greedy heuristic and Monte Carlo sampling), the asymptotic convergence of both the nested batch and batch knowledge gradient policy follows from the sequential KG cases (Frazier et al., 2009; Frazier and Powell, 2011). Specifically, it can be shown that the knowledge gradient policy will measure

each alternative infinitely often as the (nested) batch measurement budget goes to infinity and will discover which alternative is the best regardless of the imperfection of the prior.

6.6 Numerical Experiments on NBKG and Optimizing Photocurrent

In this section, we present simulation results for the material science application described in Section 6.2: optimizing the photocurrent of a photoactive device that has anisotropic nanoparticles immobilized onto its surface. We wish to maximize the output current $I(d, \rho)$ with respect to size d and the logarithm of NP density ρ. In the physical setting, preparing NPs of a particular size is expensive, while varying the density of a NP can be done easily and in parallel experiments. Therefore, we model the choice of experiment to perform as a nested-batch decision, and apply the NBKG policy toward finding the optimal choice of size and density.

6.6.1 Prior Generation

There are many ways to incorporate domain expert's knowledge about the role of NP size and density on output current. Here we use a simplified linear model purely for the purpose of initializing a prior, but the belief model is still represented as a lookup table. Once we have initialized our prior on function values $I(d, \rho)$ (as a lookup table), the experiments are used to update our belief, which will move the posterior away from our initial linear estimate. To this end, we consider a third-order polynomial approximation of the output current $I(d, \rho)$ with respect to size d and the logarithm of NP density ρ:

$$I(d, \rho) = c_1 + c_2 d + c_3 \rho + c_4 d^2 + c_5 d\rho + c_6 \rho^2 + c_7 d^3 + c_8 d^2 \rho + c_9 d\rho^2 + c_{10} \rho^3. \quad (6.29)$$

This polynomial regression model is meant as a third-order local approximation to the true response function. A cubic polynomial was specifically selected to provide a balance between the accuracy of this approximation without containing too many terms, which would expose the model to overfitting noisy measurements.

To generate a prior distribution on the values of the regression coefficients, we incorporate the following observations, which reflect a domain expert's prior knowledge about the role of NP size and density on output current. Nanoparticles typically have physical dimensions in the range of 10 to 100 nm, but it is more instructive to parameterize them through the wavelength of light that they interact with most strongly. In the case of metal nanoparticles, this is related to their localized surface plasmon resonance, which is in turn dictated by properties including their size, shape, and aspect ratio. For the purposes of this example, we simply designate a NP geometry using the wavelength of light that they absorb most strongly. First, the experimental range of size was assumed to be between 550 nm to 1300 nm (wavelength), while the range for NP density was assumed to be between 1 NP/m^2 to 10^{15} NP/m^2. When $d = 550$ nm and $\rho = 0$, the output current is simply the output current of the photoactive device non-functionalized with NPs. We presume that this current is scaled to 1 nanoamp (nA). For extreme values where $d = 1300$ nm and $\rho = 15$, we assume the current is a nominally small value 0.001 nA. Lastly, we presume that for points away from the extremes, the current has moderate values between 1 and 20 nA.

Prior generation was performed by uniformly sampling values

$$d_1 = 550 \leq d_2 \leq d_3 \leq d_4 = 1300 \text{ nm},$$

and

$$\rho_1 = 0 \leq \rho_2 \leq \rho_3 \leq \rho_4 = 15.$$

Sixteen points of the form $(d_i, \rho_j, I(i) + I(j))$ were calculated, where

$$I(i) = \begin{cases} 0.5 & i = 1; \\ 1 & i = 2, 3; \\ 0.0005 & i = 4. \end{cases}$$

We then computed the least-squares fit of the polynomial model in Equation (6.29) to these points, and obtained an instance of the regression parameters c_i. This procedure was repeated several times, resulting in an empirical distribution on c, which we use as the regression parameters' prior distribution. From this, we obtain the induced prior distribution on function values (as a lookup table) that incorporate the domain expert's prior (albeit limited) knowledge about the behavior of the photocurrent with respect to NP size and density. Figure 6.1 plots several instances of $I(d, \rho)$ obtained in the above manner.

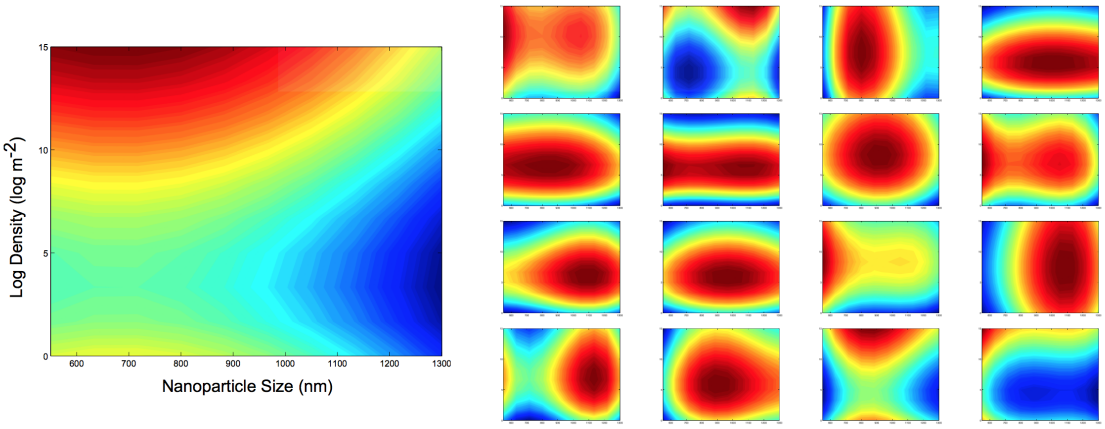


Figure 6.1: Example plots of photocurrent $I(d, \rho)$ obtained from the procedure outlined above.

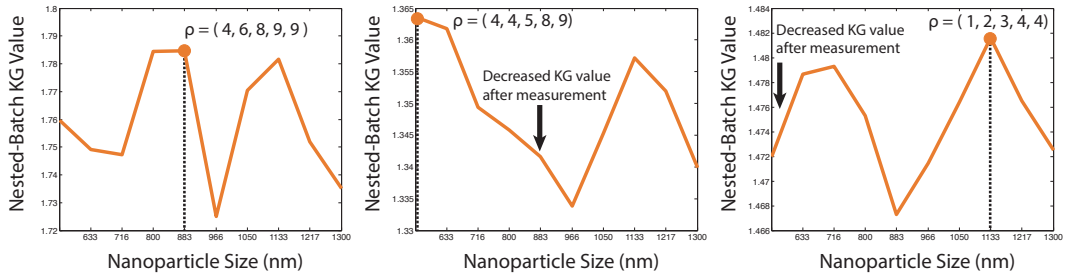
6.6.2 Performance of NBKG

In order to assess the performance of the NBKG policy, we performed several numerical experiments in which the decision-measurement-update loop was simulated over several batch measurements and over several trials. For each simulation trial, a true value of the regression parameters (and hence a true response surface) was fixed, but unknown to the simulation.

Illustration on NBKG Policy

We first illustrate how NBKG works under a measurement noise of 30% of the function range. At each iteration, a NBKG value was calculated for each choice of NP size. Example NBKG values are depicted in Figure 6.2, which is an example of NBKG values after zero, one and two measurements, respectively. The optimal NP size and corresponding batch of log density values are given at each step. The figure also illustrates a key feature of the KG policy, as shown by a marked decrease in the relative KG value of a NP size after it has been measured. Due to correlation, the values of measuring adjacent alternatives also drops since they roughly provide similar information. As shown in Figure 6.2, the KG value for NP size = 800nm drops after measuring NP size = 883nm. This gives the KG policy the ability to explore parameter space during the initial set of measurements. From the KG values, the optimal NP size and five corresponding NP densities were selected in the nested-batch method outlined above. After a noisy measurement is made from the true surface, the posterior distribution on μ is calculated according to Equations (6.1). This process is repeated until 15 batch measurements are made.

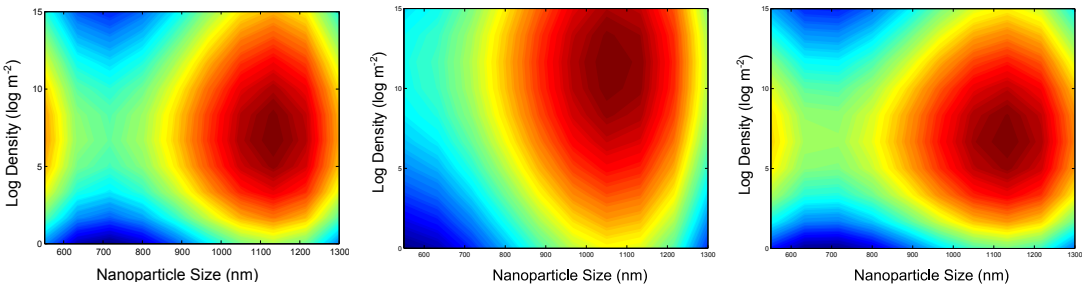
Figure 6.3 together with Figure 6.4 shows an example of the prior and posterior estimates of the true photocurrent function for a particular simulation. In Figure 6.3, the leftmost figure depicts the true photocurrent function values. The middle and rightmost figures demonstrate the prior and posterior estimates of the true function



(a) 0 batch measurements (b) 1 batch measurements (c) 2 batch measurements

Figure 6.2: NBKG values before and after 3 batch measurements. The optimal NP size at each step is indicated by the dashed line, and the corresponding optimal batch of densities are also shown. The arrows indicate the decrease in KG value for the NP size that was previously measured.

surface after 0 and 15 batch measurements, respectively, using the NBKG policy. Also depicted in Figure 6.4 is the residual error, which is the difference between the estimate and true function values. The residual errors are calculated after 0, 5, 10 or 15 batch measurements. By examining the residual error plot after 15 measurements, we see that the function value at the true maximum alternative is well approximated, while moderate error in the estimate is located away from this region of interest.

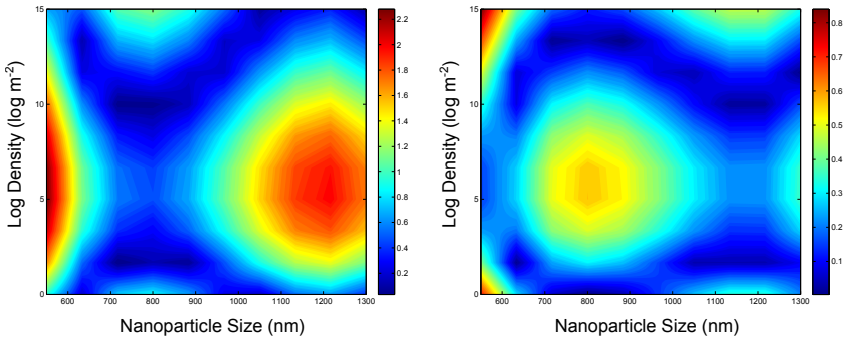


(a) True function values (b) Prior estimation (c) Posterior estimation

Figure 6.3: Prior and posterior estimates of the true function surface after 0 and 15 batch measurements, using the NBKG policy.

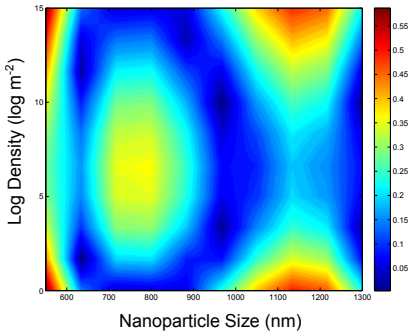
Computational Analysis

In this section, we analyze the performance of NBKG as parameters vary. As a more quantitative measure of the performance of NBKG, we consider the opportunity cost

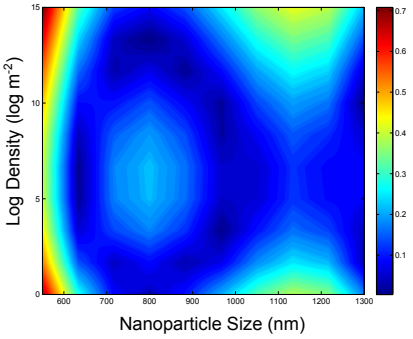


(a) 0 batch measurements

(b) 5 batch measurements



(c) 10 batch measurements



(d) 15 batch measurements

Figure 6.4: Prior and posterior estimates of the true function surface after 0, 5, 10 and 15 batch measurements, using the NBKG policy. For each choice of number of measurements, the plot shows the residual error between this estimate and the true function.

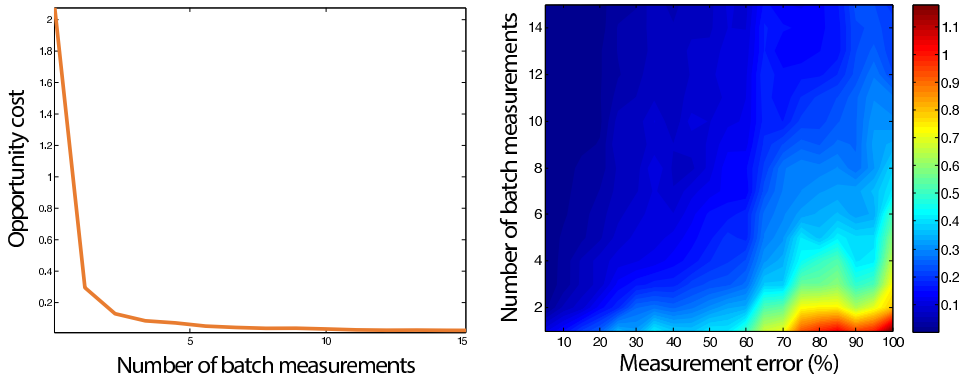
(OC) as a function of the number of batch measurements K:

$$\mathrm{OC}^K = \max_{(x,y)} \mu_{(x,y)} - \mu_{(x^K,y^K)}, \tag{6.30}$$

where $(x^K, y^K) = \arg\max_{(x,y)} \theta^K_{(x,y)}$.

Figure 6.5 shows the mean OC versus number of batch measurements, averaged over 500 simulation trials. In Figure 6.5(a), we see this plot for the case when the measurement error is 30% of the true function's range (as before). We observe that the OC quickly decays as the number of measurements increases, showing that the NBKG value rapidly finds the location of the maximal photocurrent. Figure 6.5(b) shows the mean OC versus the number of batch measurements and measurement

error. We observe that the OC increases with increasing error, as expected. Such a plot is meaningful in experimental budgeting, and shows the requisite number of measurements needed to obtain a certain level of optimality for a particular level of noise. This plot can suggest to the experimenter the amount of measurement precision needed in order to achieve a desired level of optimality as measured by opportunity cost.



(a) OC under 30% measurement error

(b) OC vs. number of batch measurement and measurement error

Figure 6.5: Opportunity cost

We may also assess the performance of NGKB as the problem size increases. We experiment with different batch sizes $B = 1, 2, 3, 4, 5$ and report in Figure 6.6 the mean opportunity cost after each batch measurement ranging from 0 to 15, averaged over 500 runs. In order to make a fair comparison, all the observations are pre-generated and shared for simulations with different batch sizes. We observe that no matter which batch size it uses, the OC quickly decays as the number of batch measurements increases. Since a larger batch size means more measurements at each iteration, thus providing more information and yielding more precise estimation. This intuition is also verified in Proposition 6.4.1 (Benefits of measurement). We see from the figure that for any measurement budget K, larger batch sizes yield lower OC, as expected.

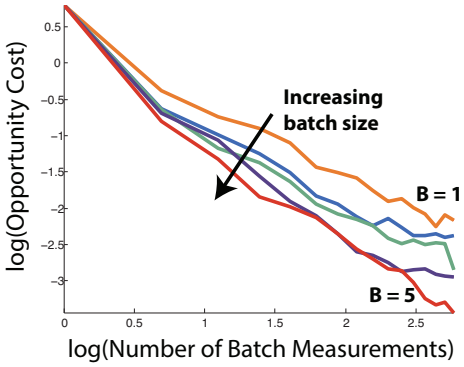


Figure 6.6: Performance of NGKB as K, B changes. Horizontal axis denotes the logarithm of the number of batch measurement $K = 0, 1, ..., 15$. Vertical axis is the logarithm of mean opportunity cost. Lines with different colors correspond to different simulations with different batch sizes $B = 1, 2, ..., 5$.

6.6.3 Comparison with Other Policies

In this section, we consider the performance of NBKG in comparison to other policies.

We consider the following policies:

1. **Nested-Batch KG**: The policy described in the paper.

2. **Sequential KG**: The basic, sequential KG policy as described in Frazier et al. (2009).

3. **Sequential Exploration**: The pure exploration policy, which chooses an alternative uniformly at random.

4. **Nested-Batch Exploration**: A random NP size is selected, and then B NP densities are selected in batch.

5. **Sequential Exploitation**: The pure exploitation policy, which chooses the alternative x^n corresponding to the maximum value, $\max_x \theta_x^n$.

6. **Nested-Batch Exploitation**: Select the batch of experiments

$$\{(d, \rho_1), \ldots, (d, \rho_B)\},$$

that maximizes

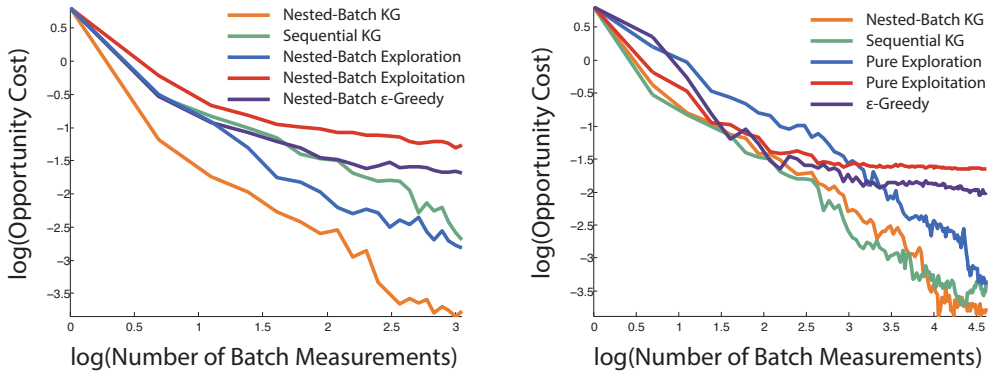
$$\mathcal{I}(d, \rho_1, \ldots, \rho_B) = \sum_{i=1}^{B} \theta^n_{(d,\rho_i)}.$$

7. **Sequential ϵ-Greedy**: A sequential policy that provides a mixture between the pure exploration and exploitation policy. The alternative x^n selected at time n is obtain by choosing between pure exploration with probability ϵ^n and pure exploitation with probability $(1 - \epsilon^n)$, where $\epsilon^n = 0.9/n$.

8. **Nested-Batch ϵ-Greedy**: Similar to the sequential ϵ-greedy policy, but chooses between the nested-batch versions of exploration and exploitation with probability ϵ^n and $(1 - \epsilon^n)$, respectively.

Figure 6.7(a) plots the mean opportunity cost for the nested-batch policies as a function of the number of batch measurements, averaged over 200 independent simulations and plotted in log scale for clarity. We observe that NBKG outperforms all the nested-batch policies. Also included in the figure is the opportunity cost for the sequential KG policy. In the nested-batch setting, the sequential KG does not take advantage of batch experiments, opting instead of performing the single experiment with largest KG value, effectively using a batch size of $B = 1$. We note that NBKG outperforms the sequential KG policy, as illustrated in Figure 6.6. The comparison between NBKG and sequential KG exhibits the experimental savings to be gained in performing experiments in batch mode. Figure 6.7(b) compares NBKG versus the sequential policies in a sequential experiment setting. In this context, we equate one batch measurement performed using the NBKG policy with B sequentially measurements for comparison. The sequential policies are more adaptive than NBKG in this manner, as they can incorporate information obtained from experiments one at a time, while NBKG only updates the state of knowledge after B measurements. Nevertheless, we observe that for a large number of measurements, NBKG outperforms all sequential policies except for sequential KG. Between sequential and NBKG,

we observe similar performance, hinting that while NBKG has a delay in updating information, the effect of this delay is minimal.



(a) Nested-batch experiments (b) Sequential experiments

Figure 6.7: A comparison of policy performance. The graphs show mean opportunity cost versus the number of measurement for the policies outlined above. (a) Nested-batch experiments, in which a policy may perform several experiments in parallel, varying NP density, provided that the NP size is the same between the parallel experiments. Sequential policies use a batch size of $B = 1$. (b) Sequential experiments, in which experiments must be performed one at a time. Here we equate 1 batch measurement with B sequential measurements.

6.7 Conclusion

In this chapter, motivated by several applications, we extended the sequential ranking and selection problem into a general framework for batch-mode learning and nested-batch-mode learning. By formulating the problem within a dynamic programming framework, we derived the Knowledge-Gradient variants to tackle both batch and nested-batch measurements. Since the Knowledge-Gradient variants require computing expectations which may be intractable, a Monte Carlo sampling procedure was applied. We empirically demonstrate the effectiveness of the NBKG policy on the immobilized nanoparticles design problem. We see that NGKB is competitive with a fully sequential strategy and significantly outperforming pure exploration, pure exploitation and ϵ-greedy strategies for the model application presented in this chapter.

Chapter 7

MOLTE: a Modular Optimal Learning Testing Environment

Since the seminal paper by Lai and Robbins (1985), there has been a long history in the optimal learning literature of proving some sort of bound, supported at times by relatively thin empirical work by comparing a few policies on a small number of randomly generated problems (Audibert and Bubeck, 2010; Cappé et al., 2013; Srinivas et al., 2009; Auer et al., 2002; Audibert et al., 2009). The problem, of course, is that compiling a library of test problems, and then running an extensive set of comparisons, is difficult. In the last chapter, we address the relative paucity of empirical testing of learning algorithms (of any type) by introducing a new public-domain, Modular, Optimal Learning Testing Environment (**MOLTE**) for Bayesian ranking and selection problem, stochastic bandits or sequential experimental design problems. The Matlab-based simulator allows the comparison of a number of learning policies (represented as a series of .m modules) in the context of a wide range of problems (each represented in its own .m module) which makes it easy to add new algorithms and new test problems. State-of-the-art policies and various problem classes are provided in the package. The choice of problems and policies is guided through a spreadsheet-

based interface. Different graphical metrics are included. **MOLTE** is designed to be compatible with parallel computing to scale up from local desktop to clusters and clouds. We demonstrate the capabilities of **MOLTE** through a series of comparisons of policies on a starter library of test problems. We also address the problem of tuning and constructing priors that have been largely overlooked in optimal learning literature. We envision **MOLTE** as a modest spur to provide researchers an easy environment to study interesting questions involved in optimal learning.

Similar libraries have been proposed for Bayesian optimization in different programming languages with different metrics and visualizations, for example, **BayesOpt** (Martinez-Cantin, 2014) and **Spearmint** (Snoek et al., 2012). Yet the uniqueness of **MOLTE** lies in its design goal to facilitate comprehensive comparisons, on a broader set of test problems and a broader set of policies (which is not to restricted to Bayesian algorithms), rather than just a code library. With its unique modular design, **MOLTE** allows users to easily specify their own problems or their own algorithms without limitation as long as they follow the general function interface. The choice of problems and policies is guided through a spreadsheet-based interface. Since many of the algorithms have tunable parameters, we include the feature that the user can easily indicate in the spreadsheet to specify the value of the tunable parameter, or ask the package to optimize the tunable parameter. We have designed various (graphical) comparison metrics in order to gain a comprehensive understanding of different policies from different perspective. **MOLTE** is also designed to be compatible with parallel computing to scale up from local desktop to clusters and clouds. We offer **MOLTE** as an easy-to-use tool for the research community that provides a highly flexible environment for testing a range of learning policies on a library of test problems, so that researchers can more easily draw insights into the behavior of different policies in the context of different problem classes.

MOLTE is designed for problems where decisions can be represented as a set of discrete alternatives. These might be materials, drug combinations, features in a product, and medical decisions. They might also be discretized continuous decisions such as temperatures, pressures, concentrations, length and time (e.g. how long a material is soaked in a bath).

7.1 Software Implementation

In this section, we describe the implementation of **MOLTE**[1] that is designed to test a variety of different learning policies on a library of test problems. The architecture makes it particularly easy for researchers to add new policies, and new problems.

7.1.1 Structural Overview

MOLTE is a Matlab-based modular architecture, where policies and problems are captured in a set of .m files, which makes it easy for researchers to add new policies and new problems. `MOLTE.m` compares the polices specified in an Excel spreadsheet for each problem class for `numP` times. Each time the simulator is run, it generates `numTruth` different sample paths, shared between all the policies, computes the value of the objective function for each sample path and then averages the `numTruth` replicas as the expected *terminal reward* or the expected *cumulative rewards*. The user may select in the spreadsheet to evaluate policies using either an online (cumulative reward) objective function Eq. (1.2), or an offline (terminal reward) objective function Eq. (1.1) (ranking and selection, Bayesian optimization).

In order to speed up the comparison, **MOLTE** is specially designed to be compatible with parallel computing to scale up from local desktop to clusters and clouds.

[1]The software is available at `http://www.castlelab.princeton.edu/software.htm`.

This can be achieved by first invoking `matlabpool` to submit a batch job to start a parallel environment and then use `parfor i=1:numP` instead of `for i=1:numP`.

7.1.2 Input Arguments

The input to the simulator is an Excel spreadsheet `ProblemsandAlgorithms.xls` which allows users to specify the problem classes and competing policies, as well as the belief models, the objectives, the prior construction and the measurement budgets. We provide a sample input spreadsheet in Table 7.1. For policies that have tunable parameters, a star included in the parentheses after the policy will initiate an automatic brute force tuning procedure with the optimal value reported in `alpha.txt`. The logic anticipates that tunable parameters may be anywhere from 10^{-5} up to 10^{5}. Whereas the user can also specify the value to be used for the policy in the parentheses.

Table 7.1: Sample input spreadsheet.

Problem class	**Prior**	**Measurement Budget**	**Belief Model**	**Offline/ Online**	**Number of policies**				
Bubeck1	Uninform	10	independent	Online	3	OLKG	IE(*)	UCB	
Branin	MLE	5	independent	Offline	4	UCBE(*)	IE(1.7)	KG	SR
GPR	Default	0.3	correlated	Online	4	KLUCB	EXPL	UCB	TS
NanoDesign	MLE	0.5	correlated	Offline	3	Kriging	EXPT	KG	

Problem class is the name of a pre-coded problem with a specified truth function, the number of alternatives and a default noise level. If it is a user defined problem, the user should write a .m file in the `./problemClasses` folder with the same name as presented in this spreadsheet. Due to the high popularity of Gaussian Process Regression (GPR), we offer the flexibility of directly specify the values of the parameter of GPR in the spreadsheet. For example, GPR($\sigma, \beta; M$) specifies the value of the parameters as follows (Powell and Ryzhov, 2012): the prior mean θ_x^0 is drawn from $\mathcal{N}(0, \sqrt{\sigma})$, the prior covariance matrix Σ^0 is of the form $\sigma \exp(-\beta(x - x'))$ and M is the number of alternatives.

Prior indicates the ways to get a prior, *MLE, Default, Given* and *Uninformative*.

Measurement Budget specifies the ratio between the time horizon of the decision making procedure to the number of alternatives. For example, in the spreadsheet a 5 means that the horizon will be 5 times the number of alternatives (which is 100), producing a total experimental budget of 500.

Belief Model specifies whether we are using independent or correlated beliefs for the policies which use a Bayesian belief model.

Offline/Online controls whether the objective is to maximize the expected terminal reward Eq. (1.1) or the expected cumulative rewards Eq. (1.2).

Number of Policies is the number of policies under comparison. This specifies the number of columns which contain the name of a policy to be tested, each represented in the corresponding `.m` file with the same name. If there are parentheses with a number after the name of the policy, it means setting the tunable parameter to the value specified in the parentheses. If there are parentheses with $*$, it means tuning the parameters and using the tuned value in the comparison; otherwise use the default value (in fact some policies, e.g. KG and Kriging, do not have tunable parameters). All policies are compared against the first policy in the list.

7.1.3 Output

All the data and figures are saved in a separate folder for each problem class. Within the folder of each problem class, each one of the `numP` folders (with the folder name from 1 to `numP`) contains:

`objectiveFunction.mat` saves the value of the online or offline objective function achieved by each policy for each of the `numP` replica.

`choice.mat` saves the decisions made by each policy in a variable named `choices` and the name of all policies in another variable `policies`.

`FinalFit.mat` saves the final estimate of the surface by each policy after the measurement budget exhausted, together with the corresponding truth. This file is only obtained for the first trial.

`alpha.txt` saves the value of tunable parameter for each policy that requires tuning, i.e. with a $(*)$ in the input spreadsheet.

`offline_hist.pdf` is the histogram for each policy describing the distribution over `numP` trials of the expected terminal reward compared to the reward obtained by the reference policy (which is the first policy in the input spreadsheet).

`online_hist.pdf` is the histogram describing the distribution of the expected cumulative reward over `numP` trials. One of the example figure is Fig. 7.1. A distribution centered around a positive value implies the policy underperforms the reference policy, which in this example is UCBV.

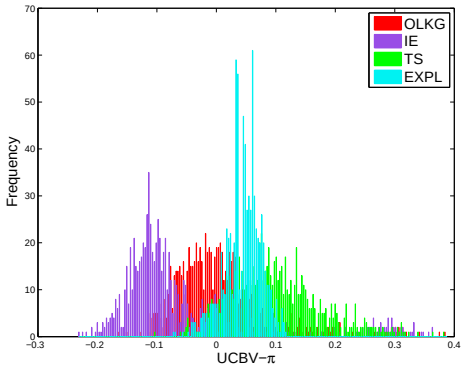


Figure 7.1: Example figure of online_hist.pdf.

It is always useful for researchers to examine the sampling pattern of each policy to gain a better understanding of its behavior. To this end, we provide a function `histChoice.m` that reads in the `choice.mat` and generates the distribution of the frequency of choosing each of the alternatives for each policy. `filedir` specifies which one of the `numP` trials is used to generate the sampling pattern, e.g. `filedir='./1/'`. Since within each trial, `numTruth` different truths are sampled, `numT` is used to indicate the number of truths the user would like to draw the sampling pattern from. Figure

7.2 is an example of a sampling pattern with the x-axis the 100 alternatives and the histogram of the sampling pattern under a measurement budget of 300.

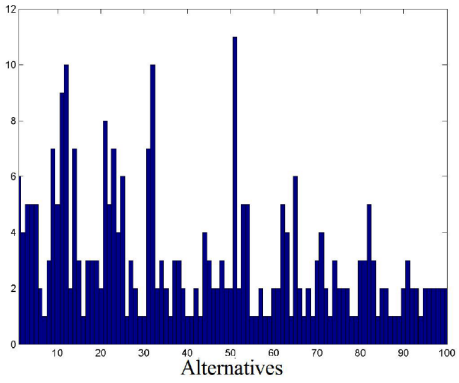
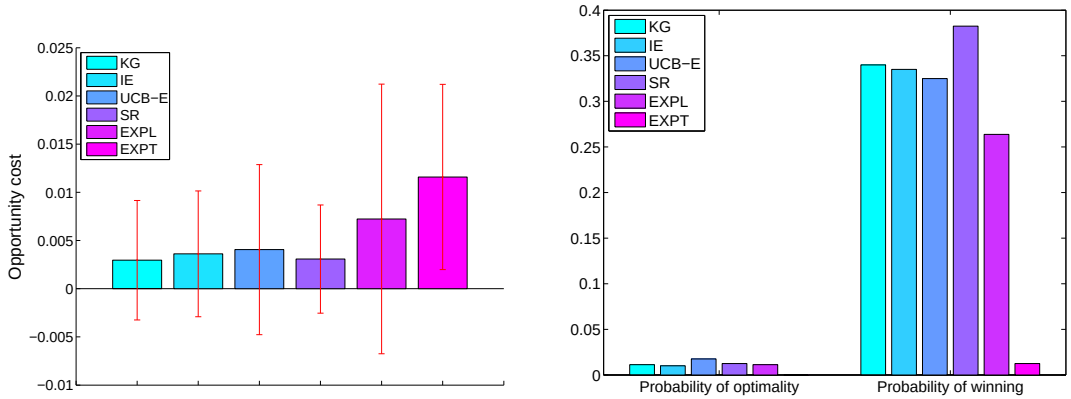


Figure 7.2: Example figure of the histogram of the frequency of choosing each of the alternative under a policy.

We also provide other graphical metrics for comparing the policies. `genProb.m` can read in the `objectiveFunction.mat` and depict the mean opportunity cost with error bars indicating the standard deviation of each policy as shown in Figure 7.3(a), together with the probability of each policy being optimal and being the best in Figure 7.3(b).



(a) Opportunity cost

(b) Probability of optimality/winning

Figure 7.3: (a) depicts the mean opportunity cost with error bars indicating the standard deviation. The first bar group in (b) demonstrates the probability that the final recommendation of each policy is the optimal one. The second bar group in (b) illustrates the probability that the opportunity cost of each policy is the lowest.

The statistics stored in `objectiveFunction.mat`, `choice.mat` and `FinalFit.mat` can easily be used for other illustrations. For example, one can use the truth values stored in `FinalFit.mat` and the number of times each policy samples each alternative in `choice.mat` to generate two dimensional contour plot using Matlab commands `contour(...)`, `plot(...)` and `text(...)`, as well as the corresponding posterior contour using the final estimate of the surface stored in `FinalFit.mat`, as we demonstrate later in Fig. 7.4.

7.1.4 Pre-coded Problem Classes

While a wide range of problem classes and policies are precoded in **MOLTE**, in the next two subsections we only briefly summarize the problem classes and policies mentioned in the following numerical experiments of this paper. As of this writing, **MOLTE** includes 23 pre-coded problem classes, and 20 pre-coded policies.

Bubeck's Experiments: (Audibert and Bubeck, 2010) We consider Bernoulli distributions with the mean of the best arm always $\mu_1 = 0.5$. M is the number of arms.

Bubeck1: $M = 20$, $\mu_{2:20} = 0.4$.

Bubeck2: $M = 20$, $\mu_{2:6} = 0.42$, $\mu_{7:20} = 0.38$.

Bubeck3: $M = 4$, $\mu_i = 0.5 - (0.37)^i$, $i \in \{2, 3, 4\}$.

Bubeck4: $M = 6$, $\mu_2 = 0.42$, $\mu_{3:4} = 0.4$, $\mu_{5:6} = 0.35$.

Bubeck5: $M = 15$, $\mu_i = 0.5 - 0.025i, i \in \{2, \cdots, 15\}$.

Bubeck6: $M = 20$, $\mu_2 = 0.48$, $\mu_{3:20} = 0.37$.

Bubeck7: $M = 30$, $\mu_{2:6} = 0.45$, $\mu_{7:20} = 0.43$, $\mu_{21:30} = 0.38$.

Asymmetric unimodular function (AUF): x is a controllable parameter ranging from 21 to 120. The objective function is $F(x, \xi) = \theta_1 \min(x, \xi) - \theta_2 x$, where θ_1, θ_2 and the distribution of the random variable ξ are all unknown. ξ is taken as a normal distribution with mean 60. Three noise levels are considered by setting different noise

ratios between the standard deviation and the mean of ξ: HNoise–0.5, MNoise–0.4, LNoise–0.3. Unless explicitly pointed out, experiments are taken under LNoise.

Equal-prior: $M = 100$. The true values μ_x are uniformly distributed over $[0, 60]$ and measurement noise $\sigma_W = 100$. $\theta_x^0 = 30$ and $\sigma_x^0 = 10$ for every x.

All the standard optimization test functions are flipped in MOLTE to generate maximization problems instead of minimization in line with R&S and bandit problems. The standard deviation of the additive Gaussian noise is set to 20 percent of the range of the function values.

Rosenbrock functions with additive noise:

$$f(x, y, \phi) = 100(y - x^2)^2 + (1 - x)^2 + \phi,$$

where $-3 \leq x \leq 3$, $-3 \leq y \leq 3$. x and y are uniformly discretized into 13×13 alternatives. This function is unimodel with the global minimum lies in a narrow, parabolic valley.

Pinter's function with additive noise:

$$\begin{aligned} f(x, y, \phi) = {} & \log_{10}\left(1 + (y^2 - 2x + 3y - \cos x + 1)^2\right) + \log_{10}\left(1 + 2(x^2 - 2y + 3x - \cos y + 1)^2\right) \\ & + x^2 + 2y^2 + 20\sin^2(y \sin x - x + \sin y) + 40\sin^2(x \sin y - y + \sin x) + 1 + \phi, \end{aligned}$$

where $-3 \leq x \leq 3$, $-3 \leq y \leq 3$. x and y are uniformly discretized into 13×13 alternatives.

Goldstein-Price's function with additive noise:

$$\begin{aligned} f(x, y, \phi) = {} & [1 + (x + y + 1)^2(19 - 14x + 3x^2 - 14y + 6xy + 3y^2)] \cdot \\ & [30 + (2x - 3y)^2(18 - 32x + 12x^2 + 48y - 36xy + 27y^2)] + \phi, \end{aligned}$$

where $-3 \leq x \leq 3$, $-3 \leq y \leq 3$. x and y are uniformly discretized into 13×13 alternatives. This function has several local minima.

Branins's function with additive noise:

$$f(x, y, \phi) = (y - \frac{5.1}{4\pi^2}x^2 + \frac{5}{\pi}x - 6)^2 + 10(1 - \frac{1}{8\pi})\cos(x) + 10 + \phi,$$

where $-5 \leq x \leq 10$, $0 \leq y \leq 15$. x and y are uniformly discretized into 15×15 alternatives. Branins's function has three global minima.

Ackley's function with additive noise:

$$f(x, y, \phi) = -20\exp\left(-0.2 \cdot \sqrt{\frac{1}{2}(x^2 + y^2)}\right) - \exp\left(\frac{1}{2}(\cos(2\pi x) + \cos(2\pi y))\right) + 20 + \exp(1) + \phi,$$

where $-3 \leq x \leq 3$, $-3 \leq y \leq 3$. x and y are uniformly discretized into 13×13 alternatives. In its two-dimensional form, it is characterized by a nearly flat outer region, and a large hole at the centre. The function poses a risk for optimization algorithms, to be trapped in one of its many local minima.

Hyper Ellipsoid function with additive noise:

$$f(x, y, \phi) = x^2 + 2y^2 + \phi.$$

where $-3 \leq x \leq 3$, $-3 \leq y \leq 3$. x and y are uniformly discretized into 13×13 alternatives. This function is a rotated version of the axis parallel hyper ellipsoid function. It is convex and unimodal.

Rastrigin function with additive noise:

$$f(x, y, \phi) = 20 + \left[x^2 - 10\cos(2\pi x)\right] + \left[x^2 - 10\cos(2\pi y)\right] + \phi,$$

where $-3 \leq x \leq 3$, $-3 \leq y \leq 3$. x and y are uniformly discretized into 11×11 alternatives. This function is highly multimodal, but locations of the minima are regularly distributed.

Six-hump camel back function with additive noise:

$$f(x, y, \phi) = (4 - 2.1x^2 + \frac{x^4}{3})x^2 + xy + (-4 + 4y^2)y^2 + \phi,$$

where $-2 \leq x \leq 2$, $-1 \leq y \leq 1$. x and y are uniformly discretized into 13×13 alternatives. With the selected input domain, the function has six local minima, two of which are global.

7.1.5 Pre-coded Policies

We have pre-coded various state-of-the-art policies π, which differ according to their decision $X^{\pi,n}(S^n)$ of the alternative to measure at time n given state S^n.

Knowledge gradient (KG): (Frazier et al., 2008, 2009) This policy is designed for offline objective (1.1). Define the knowledge gradient as

$$\nu_x^{\mathrm{KG},n} = \mathbb{E}[\max_{x'} \theta_{x'}^{n+1} - \max_{x'} \theta_{x'}^n | x^n = x, S^n].$$

$$X^{\mathrm{KG},n}(S^n) = \arg\max_{x \in \mathcal{X}} \nu_x^{\mathrm{KG},n}.$$

Online knowledge gradient (OLKG): (Ryzhov et al., 2012)

$$X^{\mathrm{OLKG},n}(S^n) = \arg\max_{x \in \mathcal{X}} \theta_x^n + (N - n)\nu_x^{\mathrm{KG},n}.$$

Interval Estimation (IE): (Kaelbling, 1993)

$$X^{\mathrm{IE},n}(S^n) = \arg\max_x \theta_x^n + z_{\alpha/2}\sigma_x^n,$$

where $z_{\alpha/2}$ is a tunable parameter.

Kriging: Huang et al. (2006)

Let $x^* = \arg\max_x(\theta_x^n + \sigma_x^n)$, and then

$$X^{\text{Kriging},n}(S^n) = \arg\max_x(\theta_x^n - \theta_{x^*}^n)\Phi(\frac{\theta_x^n - \theta_{x^*}^n}{\sigma_x^n}) + \sigma_x^n\phi(\frac{\theta_x^n - \theta_{x^*}^n}{\sigma_x^n}),$$

where ϕ and Φ are the standard normal density and cumulative distribution functions.

Thompson sampling (TS): (Thompson, 1933)

$$X^{\text{TS},n}(S^n) = \arg\max_x \tilde{\theta}_x^n,$$

where $\tilde{\theta}_x^n \sim \mathcal{N}(\theta_x^n, \sigma_x^n)$ for independent beliefs or $\tilde{\theta}_x^n \sim \mathcal{N}(\theta^n, \Sigma^n)$ for correlated beliefs.

UCB: (Auer et al., 2002)

$$X^{\text{UCB},n}(S^n) = \arg\max_x \hat{\theta}_x^n + \sqrt{\frac{2V_x^n \log n}{N_x^n}},$$

where $\hat{\theta}_x^n$, V_x^n, N_x^n are the sample mean of μ_x, sample variance of μ_x, and number of times x has been sampled up to time n, respectively. The quantity $\hat{\theta}_x^0$ is initialized by measuring each alternative once. These are similarly defined in the following variants of UCB.

UCB-E: (Audibert and Bubeck, 2010)

$$X^{\text{UCB-E},n}(S^n) = \arg\max_x \hat{\theta}_x^n + \sqrt{\frac{\alpha}{N_x^n}},$$

where α is a tunable parameter.

UCB-V: (Audibert et al., 2009)

$$X^{\text{UCB-V},n}(S^n) = \arg\max_x \hat{\theta}_x^n + \sqrt{\frac{V_x^n \log n}{N_x^n}} + 1.5\frac{\log n}{N_x^n}.$$

SR: (Audibert and Bubeck, 2010) Let $A_1 = \mathcal{X}$, $\overline{\log}(M) = \frac{1}{2} + \sum_{i=2}^{M} \frac{1}{i}$,

$$n_m = \left\lceil \frac{1}{\overline{\log}(M)} \frac{n-M}{M+1-m} \right\rceil.$$

For each phase $m = 1, ..., M-1$:

1. For each $x \in A_m$, select alternative x for $n_m - n_{m-1}$ rounds.

2. Let $A_{m+1} = A_m \setminus \arg\min_{x \in A_m} \hat{\theta}_x$.

KLUCB: (Cappé et al., 2013)

$$X^{\text{KLUCB},n}(S^n) = \arg\max_x \hat{\theta}_x^n + \sqrt{\frac{2V_x^n(\log n + 3\log\log(n))}{N_x^n}}.$$

EXPL: A pure exploration strategy that tests each alternative equally often through random sampling of the set of alternatives.

EXPT: A pure exploitation strategy.

$$X^{\text{EXPT},n}(S^n) = \arg\max_x \hat{\theta}_x^n.$$

7.1.6 Prior Generation

MOLTE features the following strategies for building a prior:

- If an *uninformative* prior is specified by the user for independent beliefs, a uniform prior will be used with $\theta_x^0 = 0$ and $\sigma_x^0 = inf$ for every x. In such case, same as with frequentist approaches (for example, UCBs), Bayesian approaches will measure each alternative once at the very beginning.

- User-defined priors can be achieved either by specifying the parameters of the problem class, e.g. GPR(50, 0.45;100), or by providing a `Prior_problemClass.mat` file containing `mu_0`, `covM` and `beta_W` in the `./Prior` folder, e.g. `Prior_GPR.mat`.

- If maximum likelihood estimation (*MLE*) is chosen to obtain the prior distribution for either independent beliefs or correlated beliefs, we follow Jones et al. (1998) and Huang et al. (2006) to use Latin hypercube designs for initial fit. For independent beliefs, we adopt a uniform prior with the same mean value θ_x^0 and standard deviation σ_x^0 for all alternatives. For correlated beliefs, we use a constant mean value θ_x^0 for all alternatives and a prior covariance matrix of the form

$$\Sigma_{xx'}^0 = \sigma e^{-\sum_{i=1}^{d} \lambda_i (x_i - x_i')^2},$$

where each arm x is a d-dimensional vector and σ, λ_i are constant. We adopt the rule of thumb by Jones et al. (1998) for the default number $(10 \times p)$ of points, where p is the number of parameters to be estimated. In addition, as suggested by Huang et al. (2006), to estimate the random errors, after the first $10 \times p$ points are evaluated, we add one replicate at each of the locations where the best p responses are found. Maximum likelihood estimation is then used to estimate the parameters based on the points in the initial design.

7.2 Experiments for Offline (Terminal Reward) Problems

In this section we report on a series of experiments with the goal of illustrating the use of **MOLTE** and the types of reports that it produces. We do not attempt to demonstrate that any policy is better than another, but our experiments support the hypothesis that different policies work well on different problem classes. This observation supports the claim that more careful empirical work is needed to develop a better understanding of which policies work best, and under what conditions.

We consider correlated beliefs between alternatives in order to strengthen the effect of each measurement so that one measurement of some alternative can provide information for other alternatives.

In order to better understand the behavior of each policy, a useful way is to examine the sampling pattern of each policy. We present an example of the frequency of measuring each alternative for each competing policy for Branin functions with a measurement budget of 100. To take advantage of correlated beliefs, rather than measuring each alternative once to initialize the empirical mean, we use the prior mean as the starting point and use the posterior mean θ^n in place of the empirical mean $\hat{\theta}^n$ for UCB-E. In the left column of Figure 7.4, the sampling pattern of each policy is displayed together with the contour of the Branin objective function which exhibits one global maximum at $(-3, 12)$ and other two local maxima at $(9, 3)$ and $(16, 4)$. The frequency that each alternative is measured is marked in numbers. The right column depicts the final prediction under each policy. All the observations are pre-generated and shared for all policies. We see from the figures that since KGCB and Kriging take correlation into consideration in the decision functions, they need less exploration and rely on the correlation to provide information for less explored alternatives. They quickly begin to focus on the alternatives that have the best values. Yet Kriging wanders around local minima for a while before it heads toward the global maximum. Note that the prediction of KGCB gives a good match in general. The function value at the true maximum alternative is well approximated, while moderate error in the estimate is located away from this region of interest. UCB-E is exploring more than necessary and wasting time on less promising regions. But when the budget is big enough, the exploration will contribute to better prediction of the surface, leading to a potentially larger final outcome in the long run. Pure exploitation gets stuck in a seemingly good alternative and the sampling pattern is not reasonable nor meaningful.

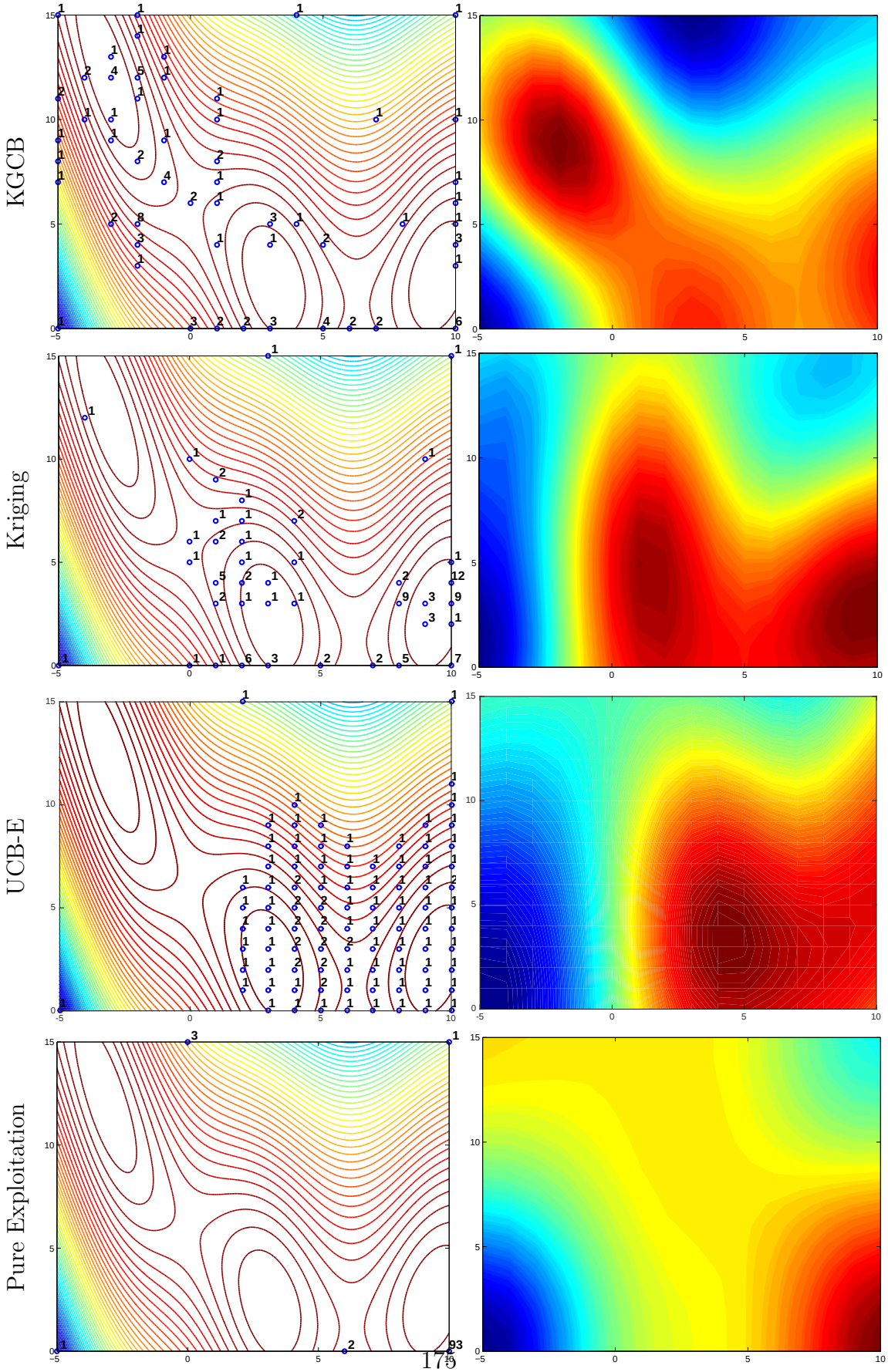


Figure 7.4: Left column: sampling distribution. Right column: posterior distribution.

7.3 Experiments for Online (Cumulative Reward) Problems

In this section, we provide sample comparisons of different policies using the online (cumulative reward) objective function. The performance measure that we use to evaluate a policy π in an online setting is $\frac{\bar{R}_N^\pi}{N}$, where the pseudo-regret $\bar{R}_N^\pi$ is defined as

$$\bar{R}_N^\pi = N \max_{x \in \mathcal{X}} \mu_x - \sum_{n=0}^{N} \mathbb{E}[\mu_{X^{\pi,n}(s^n)}].$$

The opportunity cost (OC) between two policies in an online setting is defined as the difference of their pseudo-regrets.

7.3.1 Experiments with Independent Beliefs

In real world problems, especially in experimental science, frequentist techniques cannot incorporate prior knowledge from domain experts, relying instead on the training from vast pools of data. This may be infeasible to perform in reality since running one experiment might be very expensive. The advantage of a Bayesian approach is unarguable in such cases. However, if we use MLE to fit the prior instead of using domain knowledge, it seems that the comparisons are in favor of Bayesian approaches by using an extra $11 \times p$ measurements. In order to make a seemingly more fair comparison in our synthetic experimental setting, we also experiment with uninformative priors with no additional information provided for Bayesian approach.

Tables 7.2, 7.3 and 7.4 provide comparisons of OLKG, IE with tuning, UCB-E with tuning, UCB, UCB-V, KLUCB, pure exploration (EXPL) using the Bubeck problems with uninformative prior. The measurement budgets are set to 10, 100 and 500 times the number of alternatives of each problem class in Tables 7.2, 7.3 and 7.4, respectively. IE and UCB-E are carefully tuned for each problem class. Under

each problem class, we ran each policy for `numP=1000` times. In each run, all the measurements are pre-generated and shared across all the policies. For each policy we record the normalized opportunity cost between OLKG and other competing policies, where the normalized opportunity cost is defined as the ratio between the opportunity cost $\frac{\bar{R}_N^{\pi}}{N} - \frac{\bar{R}_N^{\text{OLKG}}}{N}$ and the range of the truth μ. Positive values of OC indicate that the corresponding policy underperforms OLKG on average. Other than the interest of average performance measured by pseudo-regret, only one sample path will be realized in real world experiments and it is meaningful to find out which policy is most likely to perform the best in one sample run. Thus we also report the probability that each of the other policy outperforms (obtains a lower regret than) OLKG within 1000 realizations. Any policy can be set as a benchmark by placing it as the first policy in the input spreadsheet.

We see from the three tables that the probability of any other policy that outperforms OLKG is in general much less than 0.5. If this criterion is what an experimenter anticipates, then OLKG is a safe choice in most situations. We then discuss the performance of each policy in terms of OC. At the beginning of each trial, IE and UCB-E are more exploiting than exploring while OLKG tends to explore before it moves toward the best estimates. This contributes to good performance (measured by OC) of IE and UCB-E in Table 7.2 with a small measurement budget. The tuned values of parameters further sharpen this effect by utilizing smaller values compared to those under larger measurement budgets as reported in Table 7.5 which summarizes the optimally tuned values for each parameter. Since UCB policies tend to explore more than necessary (which can be seen from the sampling pattern, for example, Figure 7.4), the performance degenerates with a moderate measurement budget as shown in Table 7.3. In this case, OLKG yields the best performance since after an exploration period, it begins to focus on the alternatives that have the best estimates while looking for alternatives whose estimates are less certain. Yet exploration benefits in

Table 7.2: The difference between each policy and OLKG (OC), and the probability that each policy outperforms OLKG, using uninformative priors with a measurement budget 10 times the number of alternatives.

Problem Class	IE		UCBE		UCBV		UCB		KLUCB		EXPL	
	OC	Prob.	OC	Prob.	OC	Prob.	OC	Prob.	OC	Prob.	OC	Prob.
Bubeck1	-0.031	0.43	-0.032	0.43	0.073	0.51	0.016	0.35	0.054	0.50	0.078	0.50
Bubeck2	-0.032	0.55	-0.031	0.52	0.097	0.30	0.025	0.43	0.070	0.35	0.105	0.29
Bubeck3	-0.000	0.29	0.006	0.30	0.068	0.26	0.021	0.53	0.020	0.34	0.095	0.23
Bubeck4	-0.004	0.39	-0.003	0.57	0.100	0.36	0.029	0.48	0.040	0.40	0.124	0.33
Bubeck5	-0.019	0.71	-0.020	0.71	0.213	0.01	0.018	0.48	0.087	0.11	0.255	0.00
Bubeck6	-0.034	0.49	-0.035	0.48	0.139	0.34	0.034	0.41	0.098	0.37	0.151	0.33
Bubeck7	-0.036	0.70	-0.036	0.71	0.065	0.17	0.009	0.48	0.043	0.22	0.073	0.15

Table 7.3: The difference between each policy and OLKG (OC), and the probability that each policy outperforms OLKG, using uninformative priors with a measurement budget 100 times the number of alternatives.

Problem Class	IE		UCBE		UCBV		UCB		KLUCB		EXPL	
	OC	Prob.	OC	Prob.	OC	Prob.	OC	Prob.	OC	Prob.	OC	Prob.
Bubeck1	0.006	0.34	0.015	0.32	0.387	0.36	0.245	0.14	0.311	0.37	0.431	0.36
Bubeck2	0.006	0.31	0.017	0.35	0.399	0.09	0.226	0.17	0.309	0.22	0.458	0.06
Bubeck3	0.002	0.32	0.007	0.31	0.111	0.18	0.077	0.39	0.052	0.25	0.214	0.07
Bubeck4	-0.014	0.31	-0.005	0.30	0.232	0.27	0.156	0.32	0.114	0.30	0.365	0.17
Bubeck5	-0.003	0.39	0.003	0.34	0.228	0.01	0.064	0.26	0.094	0.15	0.425	0.00
Bubeck6	0.014	0.38	0.025	0.38	0.522	0.10	0.274	0.12	0.380	0.10	0.619	0.09
Bubeck7	0.015	0.52	0.016	0.44	0.260	0.00	0.158	0.21	0.215	0.09	0.303	0.00

the long run. Thus the performance of UCB policies and IE improves if allowed to explore for a sufficiently long time as reported in Table 7.4.

7.3.2 Experiments with Correlated Beliefs

In this section, we summarize numerical experiments on problems with correlated beliefs between different policies, including OLKG, IE with tuning, UCBE, UCB-V, Kriging, UCB, Thompson Sampling (TS) and pure exploration (EXPL). To take advantage of correlated beliefs, we use the prior mean as the starting point and use posterior mean θ^n in place of the empirical mean for UCB-V and UCB policies.

In order to gain a good understanding of the performance of the policies, **MOLTE** produces histograms illustrating the distribution of the difference between the nor-

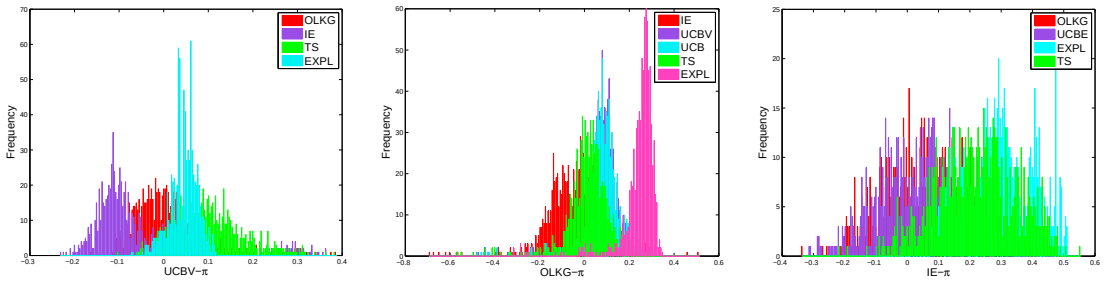
Table 7.4: The difference between each policy and OLKG (OC), and the probability that each policy outperforms OLKG, using uninformative priors with a measurement budget 500 times the number of alternatives.

Problem Class	IE		UCBE		UCBV		UCB		KLUCB		EXPL	
	OC	Prob.	OC	Prob.	OC	Prob.	OC	Prob.	OC	Prob.	OC	Prob.
Bubeck1	-0.105	0.30	-0.098	0.30	0.296	0.26	0.288	0.10	0.175	0.27	0.634	0.26
Bubeck2	-0.089	0.28	-0.080	0.26	0.253	0.31	0.226	0.15	0.139	0.32	0.609	0.02
Bubeck3	-0.009	0.34	-0.006	0.31	0.069	0.18	0.077	0.39	0.035	0.29	0.268	0.03
Bubeck4	-0.075	0.28	-0.069	0.27	0.091	0.26	0.174	0.24	0.014	0.26	0.462	0.12
Bubeck5	-0.030	0.33	-0.026	0.31	0.066	0.28	0.050	0.23	0.012	0.34	0.462	0.00
Bubeck6	-0.024	0.26	-0.022	0.24	0.310	0.05	0.227	0.16	0.190	0.06	0.771	0.05
Bubeck7	-0.045	0.33	-0.045	0.34	0.262	0.11	0.152	0.23	0.200	0.27	0.430	0.00

Table 7.5: Tuned parameters of IE and UCB-E under different problem classes and measurement budgets. The second row indicates the ratio between the measurement budget and the number of alternatives.

Problem Class	IE			UCBE		
	10	100	500	10	100	500
Bubeck1	0.0007079	1.295	2.036	0.0008991	0.3934	1.103
Bubeck2	0.1675	1.295	2.169	0.002359	0.337	0.9063
Bubeck3	0.8991	1.395	1.878	0.1206	0.4562	0.8635
Bubeck4	0.8991	1.571	2.196	0.004392	0.5332	1.197
Bubeck5	0.004566	1.395	2.169	0.0003102	0.3518	1.002
Bubeck6	0.09063	1.197	1.642	0.000505	0.3201	0.7748
Bubeck7	0.002773	0.8991	1.878	0.0005936	0.2169	0.8007

malized OC of a benchmark policy and either of the other policies over 1000 runs. Whichever policy that is listed as the first policy is treated as the benchmark. The measurement budget is set to 0.2 times the number of alternatives of each problem class. Figure 7.5 compares the performance of several policies under various problem classes with different benchmark policies. A distribution centered around a positive value implies the policy underperforms the benchmark policy, while one centered around a negative number means the policy outperforms the benchmark. For example, Figure 7.5(a) compares the performance of UCBV, OLKG, IE, TS and EXPL under Goldstein with UCBV as the benchmark policy. We can see that the tuned IE and OLKG are outperforming UCBV and others are underperforming.



(a) Goldstein (b) HyperEllipsoid (c) Rastrigin

Figure 7.5: Normalized opportunity cost between different policies.

Table 7.6: Comparisons with OLKG for correlated beliefs with the measurement 0.2 times the number of alternatives of each problem class.

Problem Class	IE		UCBE		UCBV		Kriging		TS		EXPL	
	OC	Prob.	OC	Prob.	OC	Prob.	OC	Prob.	OC	Prob.	OC	Prob.
Goldstein	-0.061	0.81	-0.097	0.92	-0.003	0.45	-0.031	0.73	0.100	0.09	0.041	0.16
AUF_HNoise	0.058	0.40	0.022	0.43	0.037	0.54	0.031	0.39	0.073	0.22	0.047	0.48
AUF_MNoise	0.043	0.29	0.027	0.42	0.343	0.21	0.023	0.28	0.173	0.21	-0.057	0.52
AUF_LNoise	-0.043	0.73	-0.013	0.64	0.053	0.51	0.005	0.53	0.038	0.20	0.003	0.62
Branin	-0.027	0.76	0.025	0.24	0.026	0.26	0.004	0.54	0.041	0.07	0.123	0.00
Ackley	0.007	0.42	0.04	0.41	0.106	0.20	0.037	0.42	0.100	0.23	0.344	0.00
HyperEllipsoid	-0.059	0.73	0.064	0.12	0.08	0.07	0.146	0.22	0.011	0.38	0.243	0.03
Pinter	-0.028	0.56	-0.003	0.51	0.029	0.42	-0.055	0.65	0.122	0.19	0.177	0.04
Rastrigin	-0.082	0.70	-0.03	0.56	0.162	0.04	-0.026	0.57	0.136	0.08	0.203	0.01

We close this section by providing more comparisons between other policies with OLKG under various problem classes. The measurement budget is set to 0.2 times the number of alternatives of each problem class. Table 7.6 reports the normalized mean OCs and the probability that each of the other policy outperforms OLKG under 1000 runs. IE and UCB-E are carefully tuned for each problem classes with the optimal value shown in Table 7.7. IE and UCB-E after tuning works generally well. Yet the optimal values of the tuned parameters are quite different for different problems as shown in Table 7.5 and 7.7. In addition, the performance of the policies are sensitive to the value of the tunable parameters. In light of this issue, we can conclude that OLKG and Kriging have one attractive advantage over IE and UCB-E: they require no tuning at all, while yielding comparable performance to a finely tuned IE or UCB-E policy. A detailed study on the issue of tuning is presented in Section 7.4.

Table 7.7: Tuned parameters of IE and UCB-E under different problem classes.

Problem Class	IE	UCBE
Goldstein	0.009939	2571
AUF_HNoise	0.01497	0.319
AUF_MNoise	0.01871	1.591
AUF_LNoise	0.01095	6.835
Branin	0.2694	0.0003664
Ackley	1.197	1.329
HyperEllipsoid	0.8991	21.21
Pinter	0.9989	0.0001636
Rastrigin	0.2086	0.001476

Table 7.6 together with the comparisons shown in previous sections suggests that there is no universal best policy for all problem classes and one could possibly design toy problems for either policy to perform the best. Similar observations have also been reported by (Kuleshov and Precup, 2000) for different bandit problems on different metrics. Besides, there are theoretical guarantees proved for each of the policy mentioned above, but the existence of these bounds does not appear to provide reliable guidance regarding which policy works best. An asymptotic bound does not provide any assurance that an algorithm will work well on a particular problem in finite time. In practice, we believe that more useful guidance could be obtained by abstracting a real world problem, running simulations and using these to indicate which policy works best.

7.4 Discussion: the Issue of Tuning

We close our presentation by discussing the tuning issues (of heuristic parameters) that tend to be overlooked in comparisons of learning algorithms.

Previous experimental results show that tuned version of IE and UCB-E yield good performance in general and yet the optimal value for IE and UCB-E may be highly problem dependent. Our experiments also suggest that the performance of a policy is sensitive to the value of the tuned parameter. For example, Figure 7.8 provides

the comparisons between the performances of IE with different parameter values (provided in the parentheses) with the online objective function under various problem classes. The measurement budget is set to five times the number of alternatives for each problem class experimented with independent beliefs and 0.3 times the number of alternatives for each problem class experimented with correlated beliefs. 'OC' is the mean opportunity cost comparing tuned IE with others $OC^{\text{IE}} - OC^{\pi}$, with a positive value indicating a win for tuned IE. 'Prob.' is the probability that other policies outperform the tuned IE. We see from the table that z_α is highly problem dependent and the performance degrades quickly away from the optimal value. For some experimental applications, tuning can require running physical experiments, which may be very expensive or even entirely infeasible.

Problem Class	B	z_α^*	IE(1)		IE(2)		IE(3)		IE(4)		IE(5)	
			OC	Prob.	OC	Prob.	OC	Prob.	OC	Prob.	OC	Prob.
Bubeck4	I	2.086	0.002	0.40	0.001	0.45	0.002	0.46	0.015	0.47	0.017	0.47
Bubeck6	I	2.01	0.003	0.44	0.001	0.48	0.004	0.43	0.013	0.23	0.028	0.13
AUF_MNoise	I	1.1305	0.004	0.38	0.041	0.04	0.071	0.00	0.095	0.00	0.114	0.09
CamelBack	I	1.295	0.006	0.35	0.069	0.32	0.108	0.03	0.145	0.00	0.172	0.00
AUF_LNoise	C	0.9498	0.043	0.00	0.080	0.00	0.105	0.00	0.123	0.03	0.136	0.00
Branin	C	0.4438	0.001	0.25	0.005	0.32	0.014	0.07	0.023	0.01	0.032	0.01
Goldstein	C	0.079	0.071	0.00	0.090	0.00	0.101	0.00	0.108	0.00	0.113	0.00
Rosenbrock	C	0.9989	0.007	0.18	0.060	0.08	0.093	0.05	0.120	0.04	0.143	0.03

Table 7.8: Comparisons between tuned IE and IEs with fixed parameter values. The second column indicates the belief model, with I for independent belief and C for correlated belief. z_α^* is the tuned value for each problem class. The number included in the parenthesis is the parameter value used by each IE policy.

7.5 Conclusion

We offer MOLTE as a public-domain test environment to facilitate the process of more comprehensive comparisons, on a broader set of test problems and a broader set of policies, so that researchers can more easily draw insights into the behavior of different policies in the context of different problem classes. There has been a long history in the optimal learning literature of proving some sort of bound, supported at times

by relatively thin empirical work by comparing a few policies on a small number of randomly generated problems. When choosing policies from a huge algorithms pool, we hope **MOLTE** can be a starting point for researchers, experimental scientists and students to more easily draw insights into the behavior of different policies in the context of different problem classes. We demonstrate the ability of **MOLTE** through extensive experimental results. We draw the conclusion that there is no universal best policy for all problem classes, and bounds, by themselves, do not provide reliable guidance to the policy that will work the best. We envision **MOLTE** as a modest spur to induce other researchers to come forward to study interesting questions involved in optimal learning, for example, the issue of tuning in this paper. We hope **MOLTE** can help with the current issue of relative paucity of empirical testing of learning algorithms.

Chapter 8

Conclusion and Future Work

8.1 Conclusion

My thesis lies at the intersection of data analytics, statistics and machine learning, focusing on sequential decision-making under uncertainty, exploring the ways where efficient information collection influences and improves decision-making strategies. This area is known as "optimal learning", an interdisciplinary field that spans Bayesian optimization, multi-armed bandit problems, derivative-free stochastic search, statistical learning and many others, with potential applications as diverse as business analytics, healthcare, natural sciences, financial data analysis and revenue management.

In many applications, decisions are made sequentially over time. Despite the previous successes of applying optimal learning techniques to transportation, e-commerce, and material sciences, most approaches are restricted to fully sequential settings with non-parametric Gaussian noise models where exact analytic solutions can be easily obtained. My thesis work centers around a class of value of information policies, known as the knowledge gradient, and provides a comprehensive set of techniques that span from designing appropriate stochastic models to describe the uncertain environment, to proposing novel statistical models and inferences with new information

collected, to finite-time and asymptotic guarantees, with an emphasis on how efficient information collection can expand access, decrease costs and improve quality in health care.

First, we provide the first finite-time bound for the knowledge gradient policy, solving the open problem of long standing. We offer a new perspective of interpreting ranking and selection problems as adaptive stochastic multi-set maximization problems and deriving the first finite-time bound of the knowledge-gradient, which characterizes KG as a near-optimal algorithm with an approximation ratio of $e/(e-1) \approx 1.582$. In addition, we propose the concept of prior-optimality which provides a cleaner relationship between the performance of the policy and the sample taken, making it possible to relate the value of information to the submodularity of the sample.

Since there are many situations where the outcomes are dichotomous, we consider the problem of sequentially making decisions that are rewarded by "successes" and "failures" which can be predicted through an unknown relationship that depends on a partially controllable vector of attributes for each instance. The learner takes an active role in selecting samples from the instance pool. The goal is to maximize the probability of success, either after the offline training phase or minimizing regret in online learning. Unlike prior work with the knowledge gradient which assumed Gaussian noise and/or linear belief models, the non-linearity introduced by the link functions causes additional computational hurdle. With the adaptation of an online Bayesian linear classifier, we propose a stochastic binary feedback (success/failure) model and designed a knowledge gradient (KG) policy to guide the experiment by maximizing the expected value of information of labeling each alternative, in order to reduce the number of expensive physical experiments.

We further study the problem of how sequentially assignment of physicians/facilities to individual patients can reduce the health care costs. This is

an example of the broader area of personalized medicine, which formalizes clinical decision making as a function that maps individual patient information (including measures of disease stage severity, medical history, clinical diagnosis) to a recommended treatment. Each experiment is expensive, forcing us to learn the most from each health episode. By formulating the problem as Bayesian contextual bandits and introducing the concept of post-observation state, we develop an optimal learning policy to guide the treatment assignment. We provide a detailed case study on a real-world knee replacement dataset. The goal is to find the treatments that lead to the best patient responses, on average, over time. Due to the intrinsic sparsity of health datasets, we use network modularity detection and LASSO to perform feature selection.

As in the healthcare example, a patient can have a number of attributes, spanning from the age, weight, to diagnoses and to their medical history. One of the most important elements in sequential decision making problems is stochastic models of the environment and proper statistical models and inferences to represent our changing beliefs about the environment as new information is collected. For this reason, we design an ensemble optimal learning method to respond quickly and robustly to complex data streams. In our ensemble systems, multiple models, such as classifiers or experts, are strategically generated and combined to minimize the unfortunate selection of a particularly poorly performing statistical model. Similar to the idea of online boosting, we use Bayesian learning with expert advice as the belief model, aiming to improve the prediction of the performance of each alternative overall, so as to spend the limited measurement budget more wisely. To the best of our knowledge, this work is the first attempt to use an online boosting framework as the prediction model in Bayesian optimization and multi-armed bandit literature. We use logistic learners as an illustration of the base models and derive an efficient and practical al-

gorithm for ensemble sequential decision making, overcoming massive computational hurdles.

Most previous work in optimal learning assumes a fully sequential setting where at each time step only one decision is made. However, the sequential design fails to account for the ability to run several parallel experiments in batches. Driven by numerous needs among materials science society, we develop a Nested-Batch-KG policy for sequential experiments when experiments can be conducted in parallel and/or there are multiple tunable parameters which are decided at different stages in the process. We demonstrate the effectiveness of our approach on the material design problem of maximizing output current of a photoactive device.

There has been a long history in the optimal learning literature of proving some sort of bound, supported at times by relatively thin empirical work by comparing a few policies on a small number of randomly generated problems. In the last past of this thesis, we present a public-domain test environment **MOLTE**, aiming to facilitate the process of more comprehensive comparisons, on a broader set of test problems and a broader set of policies, so that researchers can more easily draw insights into the behavior of different policies in the context of different problem classes.

8.2 Future Research

By bridging statistical machine learning and stochastic optimization, my goal is to develop information collection strategies that are efficient, flexible and broadly applicable.

My approach to this goal has two folds. First, I plan to develop optimal learning strategies for a wide range of belief models that are arisen from real-world applications of interest within healthcare, revenue management, market research, and elsewhere. Second, I plan to provide theoretical guarantees to characterize the performance of

information collection strategies in general and use these results to motivate better practical strategies. Possible future research directions are listed below:

Optimal learning for expensive multi-objective optimization. Many real-world applications involve multi-objective optimization. For example, in healthcare analytics, one might want to minimize healthcare cost and maximize quality of life at the same time. Multi-objective optimization then involves simultaneously optimizing more than one objective function. My goal is to develop an optimal learning algorithm that iteratively and adaptively selects a sequence of designs so as to effectively identify the Pareto frontier. Besides health care applications, other application includes financial data analysis, environmental data analysis, and transportation.

Optimal learning with ordered feedback. Many clinical procedures are measured on an ordinal scale (e.g. poor, fair, good, very good and excellent), which ideally should not be reduced for analysis to a simple dichotomy. It can be thought of as an extension of the binary generalized linear model that allows for more than two ordered response categories. The challenge lies in the ability to sequentially make decisions under ordered feedback.

Optimal learning under nonlinear constraints. Current optimal learning literature is mainly focused on Bayesian optimization with constraints only on the ranges of the design variable values to be considered. I plan to extend the methodology to general nonlinear constraints. For example, in revenue management, we care about the risk of total revenue falling below a target. One interesting line of research addresses how to compute the knowledge gradient while using a risk measure as the objective.

Active preference learning. Many online recommender systems provide each user a pair of items and record the user's choice. The goal of preference learning is to learn a predictive preference model from observed preference information. I plan to study the problem of how to adaptively select the next measuring pair so as to find the item with the highest user valuation in as few comparisons as possible. One possible way is to use Gaussian processes and design a knowledge gradient policy that depends on the probability of each alternative is preferred to another one.

Sparse optimal learning in high-dimensional settings with Relevance Vector Machine. How to effectively learn in the presence of high-dimensional data is always a challenge. Relevance Vector Machine uses Bayesian inference to obtain parsimonious solutions for regression and probabilistic classification. I plan to develop effective optimal learning methods for high-dimensional data by taking advantage of Relevance Vector Machine and kernel methods. It has the potential to be used in healthcare, image processing, text mining, bioinformatics, and financial data analysis.

Information collection using deep learning. Deep learning attempts to model high-level abstractions in data by using a deep structure with multiple processing layers, which is composed of multiple linear and non-linear transformations. I plan to design efficient information collection methods under proper deep neural net belief models and develop (approximated) updating schemas. With its ability to approximate any continuous function, this class of optimal learning methods will be broadly applicable in, but not restricted to, image processing, healthcare, revenue management, energy and economics.

Appendix A

Proofs

A.1 Proof of Lemma 2.2.6

For any ψ with $|\psi| = n$, we consider the resulting knowledge state $S^n = (\theta_x^n, \beta_x^n)_{x\in\mathcal{X}}$. Since $\sigma^W \neq 0$, there exists such ψ that $\max_x \theta_x^n > \max_{x\neq x'} \theta_x^n$ with positive probability. Now consider another realization ψ' with $\mathrm{dom}(\psi') = \mathrm{dom}(\psi) \cup \{x_2\}$, where x_2 is the second largest alternative of θ_x^n. We denote the observation of x_2 in ψ' as W_2 and the resulting S^{n+1} as $(\theta_x^{n+1}, \beta_x^{n+1})_{x\in\mathcal{X}}$ according to Bayes' rule. With independent normal beliefs, the knowledge gradient $\Delta(x|\psi) = \nu_x^{\mathrm{KG},n}$ can be analytically expressed by

$$\nu_x^{\mathrm{KG},n} = \tilde{\sigma}_x^n f(\zeta_x^n),$$

where $\tilde{\sigma}_x^n = \sqrt{(\beta_x^n)^{-1} - (\beta_x^n + \beta^W)^{-1}}$, $\zeta_x^n = -\left|\frac{\theta_x^n - \max_{x'\neq x}\theta_{x'}^n}{\tilde{\sigma}_x^n}\right|$ and $f(\zeta) = \zeta\Phi(\zeta) + \phi(\zeta)$. $\Phi(\zeta)$ and $\phi(\zeta)$ are, respectively, the cumulative standard normal distribution the standard normal density (Frazier et al., 2008). We first notice that $f'(\zeta) = \Phi(\zeta) \geq 0$ for any $\zeta \in \mathbb{R}$ so that $f(\zeta)$ is non-decreasing. We next compare $\nu_{x_1}^{\mathrm{KG},n}$ and $\nu_{x_1}^{\mathrm{KG},n+1}$ for $x_1 = \arg\max_x \theta_x^n$. According to Bayes' rule, the precision β of x changes only when x is measured. So we have $\tilde{\sigma}_{x_1}^n = \tilde{\sigma}_{x_1}^{n+1}$. Similarly we have all the θ_x^{n+1} unchanged except for alternative x_2.

We will next show that for any W_2 such that $\theta^n_{x_2} < W_2 \leq \frac{\beta^n_{x_2}}{\beta^W}(\theta^n_{x_1} - \theta^n_{x_2}) + \theta^n_{x_1}$, we have $\nu^{\mathrm{KG},n}_{x_1} < \nu^{\mathrm{KG},n+1}_{x_1}$. Recall that $\theta^{n+1}_{x_2} = \frac{\beta^n_{x_2}\theta^n_{x_2} + \beta^W W_2}{\beta^n + \beta^W}$. When $W_2 > \theta^n_{x_2}$, we have $\theta^{n+1}_{x_2} > \theta^n_{x_2}$. Since $\theta^{n+1}_x = \theta^n_x$ for all $x \neq x_2$, and $\theta^n_{x_2} = \max_{x' \neq x_1} \theta^n_{x'}$, we have $\theta^{n+1}_{x_2} = \max_{x' \neq x_1} \theta^{n+1}_{x'}$ and thus $\zeta^{n+1}_{x_1} = -\left|\frac{\theta^{n+1}_{x_1} - \max_{x' \neq x_1} \theta^{n+1}_{x'}}{\tilde{\sigma}^{n+1}_x}\right| = -\left|\frac{\theta^n_{x_1} - \theta^{n+1}_{x_2}}{\tilde{\sigma}^n_x}\right|$.

If $W_2 \leq \frac{\beta^n_{x_2}}{\beta^W}(\theta^n_{x_1} - \theta^n_{x_2}) + \theta^n_{x_1}$ holds, we have $\theta^n_{x_1} - \theta^{n+1}_{x_2} > 0$ and

$$\zeta^{n+1}_{x_1} = -\Big(\frac{\theta^n_{x_1} - \theta^{n+1}_{x_2}}{\tilde{\sigma}^n_x}\Big) > -\Big(\frac{\theta^n_{x_1} - \theta^n_{x_2}}{\tilde{\sigma}^n_x}\Big) = -\left|\frac{\theta^n_{x_1} - \theta^{n+1}_{x_2}}{\tilde{\sigma}^n_x}\right| = \zeta^n_{x_1}.$$

Due to the fact that $f(\zeta)$ is non-decreasing, we have

$$\nu^{\mathrm{KG},n}_{x_1} = \tilde{\sigma}^n_{x_1} f(\zeta^n_{x_1}) < \tilde{\sigma}^n_{x_1} f(\zeta^{n+1}_{x_1}) = \tilde{\sigma}^{n+1}_{x_1} f(\zeta^{n+1}_{x_1}) = \nu^{\mathrm{KG},n+1}_{x_1}.$$

Since $\theta^n_{x_1} > \theta^n_{x_2}$ by construction, such W_2 that satisfies $\theta^n_{x_2} < W_2 \leq \frac{\beta^n_{x_2}}{\beta^W}(\theta^n_{x_1} - \theta^n_{x_2}) + \theta^n_{x_1}$ can be obtained with positive probability.

A.2 Proof of Proposition 2.2.1

Let $z^*(Z, \pi, \Phi)$ be the next adaptive greedy choice that maximizes the expected marginal increment given that policy π has generated Z. We first show that

$$F^{\pi_2 \odot \pi_1} \leq F^{\pi_2} + n_1 \sum_{Z \in \mathcal{Z}^n} \mathbb{P}(\pi_2 \rightsquigarrow Z)\Big(\mathbb{E}\big[\hat{v}(Z \cup \{z^*(Z, \pi_2, \Phi)\}, \Phi)\big] - v(Z)\Big)$$

for all policies π_1 with a measurement budget n_1 and π_2 with a budget n_2 under any prior and probability distribution that describes a measurement.

Proof. Let $\pi^{[j]}$ denote the first j measurement decisions under some policy π. First of all we break $F^{\pi_2 \odot \pi_1} - F^{\pi_2}$ into n_1 consecutive differences,

$$F^{\pi_2 \odot \pi_1} - F^{\pi_2} = \sum_{j=1}^{n_1} \Big(F^{\pi_2 \odot \pi_1^{[j]}} - F^{\pi_2 \odot \pi_1^{[j-1]}}\Big).$$

Similar to what we did in the last lemma, for each difference we have

$$
\begin{aligned}
& F^{\pi_2 \odot \pi_1^{[j]}} - F^{\pi_2 \odot \pi_1^{[j-1]}} \\
= & \sum_{Z_1 \in \mathcal{Z}^{n_2+j}} \mathbb{P}(\pi_2 \odot \pi_1^{[j]} \rightsquigarrow Z_1) v(Z_1) - \sum_{Z_2 \in \mathcal{Z}^{n_2+j-1}} \mathbb{P}(\pi_2 \odot \pi_1^{[j-1]} \rightsquigarrow Z_2) v(Z_2) \\
= & \sum_{Z_1 \in \mathcal{Z}^{n_2+j}} \sum_{Z_2 \in \mathcal{Z}^{n_2+j-1}, Z_2 \cup Z_3 = Z_1} \mathbb{P}(\pi_2 \odot \pi_1^{[j-1]} \rightsquigarrow Z_2) \mathbb{P}(\pi_1^{\{j\}} \rightsquigarrow Z_3 | \pi_2 \odot \pi_1^{[j-1]} \rightsquigarrow Z_2) v(Z_1) \\
- & \sum_{Z_2 \in \mathcal{Z}^{n_2+j-1}} \sum_{Z_3 \in \mathcal{Z}^1} \mathbb{P}(\pi_2 \odot \pi_1^{[j-1]} \rightsquigarrow Z_2) \mathbb{P}(\pi_1^{\{j\}} \rightsquigarrow Z_3 | \pi_2 \odot \pi_1^{[j-1]} \rightsquigarrow Z_2) v(Z_2) \\
= & \sum_{Z_2 \in \mathcal{Z}^{n_2+j-1}} \sum_{Z_3 \in \mathcal{Z}^1} \mathbb{P}(\pi_2 \odot \pi_1^{[j-1]} \rightsquigarrow Z_2) \mathbb{P}(\pi_1^{\{j\}} \rightsquigarrow Z_3 | \pi_2 \odot \pi_1^{[j-1]} \rightsquigarrow Z_2) \big(v(Z_2 \cup Z_3) - v(Z_2) \big).
\end{aligned}
$$

Now we consider all possible pair (Z_4, Z_5) such that $Z_4 \in \mathcal{Z}^{n_2}$, $Z_5 \in \mathcal{Z}^{j-1}$ and $Z_4 \cup Z_5 = Z_2$. Notice that the policy $\pi_2 \odot \pi_1^{[j]}$ employs a fresh start at the time n_2, therefore the events before and after time n_2 are independent. Then we have

$$
\begin{aligned}
& \sum_{Z_2 \in \mathcal{Z}^{n_2+j-1}} \sum_{Z_3 \in \mathcal{Z}^1} \mathbb{P}(\pi_2 \odot \pi_1^{[j-1]} \rightsquigarrow Z_2) \mathbb{P}(\pi_1^{\{j\}} \rightsquigarrow Z_3 | \pi_2 \odot \pi_1^{[j-1]} \rightsquigarrow Z_2) \big(v(Z_2 \cup Z_3) - v(Z_2) \big) \\
= & \sum_{Z_2 \in \mathcal{Z}^{n_2+j-1}} \sum_{Z_4 \cup Z_5 = Z_2} \sum_{Z_3 \in \mathcal{Z}^1} \mathbb{P}(\pi_2 \rightsquigarrow Z_4) \mathbb{P}(\pi_1^{[j-1]} \rightsquigarrow Z_5) \mathbb{P}(\pi_1^{\{j\}} \rightsquigarrow Z_3 | \pi_2 \odot \pi_1^{[j-1]} \rightsquigarrow Z_2) \\
& \times \big(v(Z_2 \cup Z_3) - v(Z_2) \big).
\end{aligned}
$$

Based on the submodular property of function v, we have

$$
v(Z_2 \cup Z_3) - v(Z_2) \leq v(Z_4 \cup Z_3) - v(Z_4).
$$

Then from the definition of z^*, we have

$$
\begin{aligned}
v(Z_4 \cup Z_3) - v(Z_4) &= \mathbb{E}[\hat{v}(Z_4 \cup Z_3, \Phi) - \hat{v}(Z_4, \Phi)] \\
&= \mathbb{E}_\Phi\big[\mathbb{E}[\hat{v}(Z_4 \cup Z_3, \Phi) - \hat{v}(Z_4, \Phi) | Z^{\pi_2}(\Phi) = Z_4]\big] \\
&\leq \mathbb{E}_\Phi\big[\mathbb{E}[\hat{v}(Z_4 \cup \{z^*(Z_4, \pi_2, \Phi)\}, \Phi) - \hat{v}(Z_4, \Phi) | Z^{\pi_2}(\Phi) = Z_4]\big] \\
&= \mathbb{E}_\Phi[\hat{v}(Z_4 \cup \{z^*(Z_4, \pi_2, \Phi)\}, \Phi)] - v(Z_4).
\end{aligned}
$$

Combining the last two inequalities, we have

$$
\begin{aligned}
& \sum_{Z_2 \in \mathcal{Z}^{n_2+j-1}} \sum_{Z_4 \cup Z_5 = Z_2} \sum_{Z_3 \in \mathcal{Z}^1} \mathbb{P}(\pi_2 \rightsquigarrow Z_4)\mathbb{P}(\pi_1^{[j-1]} \rightsquigarrow Z_5)\mathbb{P}(\pi_1^{\{j\}} \rightsquigarrow Z_3 | \pi_2 \odot \pi_1^{[j-1]} \rightsquigarrow Z_2) \\
& \times \big(v(Z_2 \cup Z_3) - v(Z_2)\big) \\
\leq & \sum_{Z_2 \in \mathcal{Z}^{n_2+j-1}} \sum_{Z_4 \cup Z_5 = Z_2} \sum_{Z_3 \in \mathcal{Z}^1} \mathbb{P}(\pi_2 \rightsquigarrow Z_4)\mathbb{P}(\pi_1^{[j-1]} \rightsquigarrow Z_5)\mathbb{P}(\pi_1^{\{j\}} \rightsquigarrow Z_3 | \pi_2 \odot \pi_1^{[j-1]} \rightsquigarrow Z_2) \\
& \times \big(\mathbb{E}\hat{v}(Z_4 \cup \{z^*(Z_4, \pi_2, \Phi)\}, \Phi) - v(Z_4)\big) \\
= & \sum_{Z_2 \in \mathcal{Z}^{n_2+j-1}} \sum_{Z_4 \cup Z_5 = Z_2} \mathbb{P}(\pi_2 \rightsquigarrow Z_4)\mathbb{P}(\pi_1^{[j-1]} \rightsquigarrow Z_5)\big(\mathbb{E}\hat{v}(Z_4 \cup \{z^*(Z_4, \pi_2, \Phi)\}, \Phi) - v(Z_4)\big) \\
= & \sum_{Z_4 \in \mathcal{Z}^{n_2}} \sum_{Z_5 \in \mathcal{Z}^{j-1}} \mathbb{P}(\pi_2 \rightsquigarrow Z_4)\mathbb{P}(\pi_1^{[j-1]} \rightsquigarrow Z_5)\big(\mathbb{E}\hat{v}(Z_4 \cup \{z^*(Z_4, \pi_2, \Phi)\}, \Phi) - v(Z_4)\big) \\
= & \sum_{Z_4 \in \mathcal{Z}^{n_2}} \mathbb{P}(\pi_2 \rightsquigarrow Z_4)\big(\mathbb{E}\hat{v}(Z_4 \cup \{z^*(Z_4, \pi_2, \Phi)\}, \Phi) - v(Z_4)\big),
\end{aligned}
$$

and this ends the proof. □

Set $\pi_1 = \pi^*$ and $\pi_2 = \ \mathrm{KG}^{[n-1]}$ in Lemma 2.2.12 and the above proposition then what left to show is that

$$
F^{\mathrm{KG}^{[n]}} - F^{\mathrm{KG}^{[n-1]}} \geq \sum_{Z \in \mathcal{Z}^n} \mathbb{P}(\pi_2 \rightsquigarrow Z)\Big(\mathbb{E}\hat{v}(Z \cup \{z^*(Z, \mathrm{KG}^{[n-1]}, \Phi)\}, \Phi) - v(Z)\Big).
$$

From the definition, the left hand side of the last equation:

$$\begin{aligned} F^{\mathrm{KG}^{[n]}} - F^{\mathrm{KG}^{[n-1]}} &= \sum_{Z_1 \in \mathcal{Z}^{n+1}} \mathbb{P}(\mathrm{KG} \rightsquigarrow Z_1) v(Z_1) - \sum_{Z_2 \in \mathcal{Z}^n} \mathbb{P}(\mathrm{KG} \rightsquigarrow Z_2) v(Z_2) \\ &= \sum_{Z_2 \in \mathcal{Z}^n} \sum_{Z_3 \in \mathcal{Z}^1} \mathbb{P}(\mathrm{KG} \rightsquigarrow Z_2) \mathbb{P}(\mathrm{KG} \rightsquigarrow Z_3 | \mathrm{KG} \rightsquigarrow Z_2) v(Z_2 \cup Z_3) \\ &\quad - \sum_{Z_2 \in \mathcal{Z}^n} \mathbb{P}(\mathrm{KG} \rightsquigarrow Z_2) v(Z_2). \end{aligned}$$

Now it is enough to show that

$$\begin{aligned} & \sum_{Z_3 \in \mathcal{Z}^1} \mathbb{P}(\mathrm{KG} \rightsquigarrow Z_3 | \mathrm{KG} \rightsquigarrow Z_2) v(Z_2 \cup Z_3) - v(Z_2) \\ \geq \quad & \mathbb{E}\hat{v}(Z_2 \cup \{z^*(Z_2, \mathrm{KG}^{[n-1]}, \Phi)\}, \Phi) - v(Z_2). \end{aligned}$$

We could group together the partial realizations ψ that lead to the same single step optimal decision $z^*(Z_2, \mathrm{KG}^{[n-1]}, \Phi)$, and then the last inequality follows from the adaptive greedy nature of the KG policy.

A.3 Proof of Theorem 2.3.2

First of all, we consider the case when f is a two dimensional function and the four points we pick form a rectangle. Assume $f(x, y)$ is submodular. For any given point (x_0, y_0), we have $f(x_0 + t + s, y_0) - f(x_0 + t, y_0) \leq f(x_0 + s, y_0) - f(x_0, y_0)$ and $f(x_0 + t, y_0) - f(x_0, y_0) \leq f(x_0 + t, y_0 + s) - f(x_0, y_0 + s)$ for any $s, t > 0$. From the first inequality we get $f_{xx}(x_0, y_0) \leq 0$ directly. From the second inequality, we have $f_x(x_0, y_0) \leq f_x(x_0, y_0 + s)$, and finally $f_{x,y}(x_0, y_0) \leq 0$. On the other hand, if we have $f_{xy} \leq 0$, $f_{xx} \leq 0$, for any (x, y), then due to the fact that $f(x_0 + t, y_0 + s) - f(x_0 + t, y_0) - \big(f(x_0, y_0 + s) - f(x_0, y_0)\big) = \int_{x_0}^{x_0+t} \int_{y_0}^{y_0+s} f_{xy}(u, v) \mathrm{d}u \mathrm{d}v \leq 0$, $f(x_0 + t + s, y_0) -$

$f(x_0+t,y_0)-\big(f(x_0+s,y_0)-f(x_0,y_0)\big)=stf_{xx}(x_0+\xi,y_0)\leq 0$, for some $0<\xi<t+s$, we obtain the submodularity.

We next consider the general case when f is n dimensional and the four points only form a parallelogram. Since the difference between the two marginal values can be decomposed into summation of several marginal value differences whose reference points form rectangles that parallel to coordinate planes, the result for the general case is straightforward from the two dimensional case.

A.4 Proofs of Asymptotic Optimality

In this section, we provide detailed proofs of all the asymptotic optimal results in Section 3.5.3.

A.4.1 Proof of Proposition 3.5.1

We use $q(\boldsymbol{w})$ to denote the predictive distribution under the state s and use $p(\boldsymbol{w}|s,\boldsymbol{x},y_{\boldsymbol{x}})$ to denote the posterior distribution after we observe the outcome of $\boldsymbol{x}$ to be y. By Jensen's inequality, we have

$$\begin{aligned}\mu^{\mathrm{KG}}_{\boldsymbol{x}}(s) &= \mathbb{E}\Big[\max_{\boldsymbol{x}'} p(y=+1|\boldsymbol{x}',T(s,\boldsymbol{x},y_{\boldsymbol{x}}))\Big]-\max_{\boldsymbol{x}'} p(y=+1|\boldsymbol{x}',s)\\ &\geq \max_{\boldsymbol{x}'}\mathbb{E}\big[p(y=+1|\boldsymbol{x}',T(s,\boldsymbol{x},y_{\boldsymbol{x}}))\big]-\max_{\boldsymbol{x}'} p(y=+1|\boldsymbol{x}',s).\end{aligned}$$

We then show that $\mathbb{E}\big[p(y=+1|\boldsymbol{x}',T(s,\boldsymbol{x},y_{\boldsymbol{x}}))\big]=p(y=+1|\boldsymbol{x}',s)$ for any $\boldsymbol{x},\boldsymbol{x}'$ and s, which leads to $\mu^{\mathrm{KG}}_{\boldsymbol{x}}(s)\geq 0$. Since $y_{\boldsymbol{x}}$ is binomial distributed with mean $p(y_{\boldsymbol{x}}=+1|\boldsymbol{x},s)$, we have

$$\begin{aligned}&\mathbb{E}\big[p(y=+1|\boldsymbol{x}',T(s,\boldsymbol{x},y_{\boldsymbol{x}}))\big]\\ =\;& p(y_{\boldsymbol{x}}=+1|\boldsymbol{x},s)p(y=+1|\boldsymbol{x}',T(s,\boldsymbol{x},+1))+(1-p(y_{\boldsymbol{x}}=+1|\boldsymbol{x},s))p(y=+1|\boldsymbol{x}',T(s,\boldsymbol{x},-1)).\end{aligned}$$

Recall that $p(y=+1|\boldsymbol{x},s)=\int\sigma(\boldsymbol{w}^T\boldsymbol{x})p(\boldsymbol{w}|s)\mathrm{d}\boldsymbol{w}$. By Bayes' Theorem, the posterior distribution in the updated state $T(s,\boldsymbol{x},y_x)$ becomes

$$p(\boldsymbol{w}'|T(s,\boldsymbol{x},+1))=\frac{\sigma((\boldsymbol{w}')^T\boldsymbol{x})p(\boldsymbol{w}'|s)}{\int\sigma(\boldsymbol{w}^T\boldsymbol{x})p(\boldsymbol{w}|s)\mathrm{d}\boldsymbol{w}},$$

and

$$p(\boldsymbol{w}'|T(s,\boldsymbol{x},-1))=\frac{\left(1-\sigma((\boldsymbol{w}')^T\boldsymbol{x})\right)p(\boldsymbol{w}'|s)}{\int\left(1-\sigma(\boldsymbol{w}^T\boldsymbol{x})\right)p(\boldsymbol{w}|s)\mathrm{d}\boldsymbol{w}}.$$

Notice that

$$\begin{aligned}
p(y=+1|\boldsymbol{x}',T(s,\boldsymbol{x},+1)) &= \int\sigma((\boldsymbol{w}')^T\boldsymbol{x}')p(\boldsymbol{w}'|T(s,\boldsymbol{x},+1))\mathrm{d}\boldsymbol{w}' \\
&= \int\sigma((\boldsymbol{w}')^T\boldsymbol{x}')\frac{\sigma((\boldsymbol{w}')^T\boldsymbol{x})p(\boldsymbol{w}'|s)}{\int\sigma(\boldsymbol{w}^T\boldsymbol{x})p(\boldsymbol{w}|s)\mathrm{d}\boldsymbol{w}}\mathrm{d}\boldsymbol{w}' \\
&= \frac{\int\sigma((\boldsymbol{w}')^T\boldsymbol{x}')\sigma((\boldsymbol{w}')^T\boldsymbol{x})p(\boldsymbol{w}'|s)\mathrm{d}\boldsymbol{w}'}{\int\sigma(\boldsymbol{w}^T\boldsymbol{x})p(\boldsymbol{w}|s)\mathrm{d}\boldsymbol{w}},
\end{aligned}$$

and similarly, we have

$$p(y=+1|\boldsymbol{x}',T(s,\boldsymbol{x},-1))=\frac{\int\sigma((\boldsymbol{w}')^T\boldsymbol{x}')(1-\sigma((\boldsymbol{w}')^T\boldsymbol{x}))p(\boldsymbol{w}'|s)\mathrm{d}\boldsymbol{w}'}{\int(1-\sigma(\boldsymbol{w}^T\boldsymbol{x}))p(\boldsymbol{w}|s)\mathrm{d}\boldsymbol{w}}.$$

Therefore,

$$\begin{aligned}
&\mathbb{E}\big[p(y=+1|\boldsymbol{x}',T(s,\boldsymbol{x},y_{\boldsymbol{x}}))\big] \\
=\;& p(y=+1|s,\boldsymbol{x})p(y=+1|\boldsymbol{x}',T(s,\boldsymbol{x},+1))+p(y=-1|s,\boldsymbol{x})p(y=+1|\boldsymbol{x}',T(s,\boldsymbol{x},-1)) \\
=\;& \int\sigma((\boldsymbol{w}')^T\boldsymbol{x}')\sigma((\boldsymbol{w}')^T\boldsymbol{x})p(\boldsymbol{w}'|s)\mathrm{d}\boldsymbol{w}'+\int\sigma((\boldsymbol{w}')^T\boldsymbol{x}')(1-\sigma((\boldsymbol{w}')^T\boldsymbol{x}))p(\boldsymbol{w}'|s)\mathrm{d}\boldsymbol{w}' \\
=\;& \int\sigma((\boldsymbol{w}')^T\boldsymbol{x}')p(\boldsymbol{w}'|s)\mathrm{d}\boldsymbol{w}' \\
=\;& p(y=+1|s,\boldsymbol{x}'),
\end{aligned}$$

and thus we obtain

$$\mathbb{E}\big[p(y=+1|\boldsymbol{x}', T(s, \boldsymbol{x}, y_{\boldsymbol{x}}))\big] = p(y=+1|s, \boldsymbol{x}').$$

A.4.2 Proof of Proposition 3.5.2

The proof is similar to that by Frazier et al. (2009) with additional tricks for Bernoulli distributed random variables. Let $\mathcal{G}$ be the sigma-algebra by the collection $\{\hat{y}^{n+1}\mathbf{1}_{\{\boldsymbol{x}^n=\boldsymbol{x}\}}\}$. Since if the policy π measures alternative $\boldsymbol{x}$ infinitely often, this collection is an infinite sequence of independent random variables with common Bernoulli distribution with mean $\sigma(\boldsymbol{w}^T\boldsymbol{x})$, the strong law of large numbers implies $\sigma(\boldsymbol{w}^T\boldsymbol{x}) \in \mathcal{G}$. Since $\mathcal{G} \in \mathcal{F}^\infty$, we have $\sigma(\boldsymbol{w}^T\boldsymbol{x}) \in \mathcal{F}^\infty$. Let U be a uniform random variable in $[0, 1]$. Then the Bernoulli random variable $y_{\boldsymbol{x}}$ can be rewritten as $\mathbf{1}_{U\leq\sigma(\boldsymbol{w}^T\boldsymbol{x})}$. Since U is independent with $\mathcal{F}^\infty$ and the σ-algebra generated by $\sigma(\boldsymbol{w}^T\boldsymbol{x})$, it can be shown that by properties of conditional expectations,

$$\mathbb{E}\big[\sigma(\boldsymbol{w}^T\boldsymbol{x}')|\mathcal{F}^\infty, \mathbf{1}_{U\leq\sigma(\boldsymbol{w}^T\boldsymbol{x})}\big] = \mathbb{E}\big[\sigma(\boldsymbol{w}^T\boldsymbol{x}')|\mathcal{F}^\infty\big].$$

We next show that the knowledge gradient value of measuring alternative $\boldsymbol{x}$ is zero by substituting this relation into the definition of the knowledge gradient. We have

$$\begin{aligned}
\nu_{\boldsymbol{x}}(\mathcal{F}^\infty) &= \mathbb{E}\bigg[\max_{\boldsymbol{x}'}\mathbb{E}\big[\sigma(\boldsymbol{w}^T\boldsymbol{x}')|\mathcal{F}^\infty, \mathbf{1}_{U\leq\sigma(\boldsymbol{w}^T\boldsymbol{x})}\big]|\mathcal{F}^\infty\bigg] - \max_{\boldsymbol{x}'}\mathbb{E}\big[\sigma(\boldsymbol{w}^T\boldsymbol{x}')|\mathcal{F}^\infty\big] \\
&= \mathbb{E}\bigg[\max_{\boldsymbol{x}'}\mathbb{E}\big[\sigma(\boldsymbol{w}^T\boldsymbol{x}')|\mathcal{F}^\infty\big]|\mathcal{F}^\infty\bigg] - \max_{\boldsymbol{x}'}\mathbb{E}\big[\sigma(\boldsymbol{w}^T\boldsymbol{x}')|\mathcal{F}^\infty\big] \\
&= \max_{\boldsymbol{x}'}\mathbb{E}\big[\sigma(\boldsymbol{w}^T\boldsymbol{x}')|\mathcal{F}^\infty\big] - \max_{\boldsymbol{x}'}\mathbb{E}\big[\sigma(\boldsymbol{w}^T\boldsymbol{x}')|\mathcal{F}^\infty\big] \\
&= 0.
\end{aligned}$$

A.4.3 Proof of the Theorem 3.5.6: Consistency of the KG Policy

It has been established that almost surely the knowledge gradient policy will achieve a state (at time T) that the KG values for all the alternatives are smaller than ϵ, after which the probability of selecting each alternative is $1/M$ where M is the number of alternatives. Notice that in each round, KG policy first picks one out of M alternatives, then a feedback of either 1 or -1 is observed. Equivalently, we can interpret the two procedures as one of the $2M$ possible outcomes from the set $\tilde{\mathcal{Y}} := \{(0, \ldots, +1/-1, \ldots 0)\}$, where each $\tilde{y} \in \tilde{\mathcal{Y}}$ is a M-dimensional vector with only element being +1 or -1. It should be noted that by changing the feedback schema in this way will not affect the Bayesian update equations because the likelihood function and the normalization factor in the posterior will both multiply by a factor of $1/M$. On the other hand, this combined feedback schema makes it possible to treat each measurement $(\boldsymbol{x}^n, y^n)$ as i.i.d. samples in $\tilde{\mathcal{Y}}$.

Define the K-L neiborhood as $K_\epsilon(u) = \{v : KL(u, v) < \epsilon\}$, where the K-L divergence is defined as $KL(u, v) := \int v \log(v/u)$. Since the prior distribution is Gaussian with positive definite covariance matrix, and the likelihood function is the sigmoid function which only takes positive values, then after time T, the posterior probability in the K-L neighborhood of w^* is positive. Based on standard results on the consistency of Bayes' estimates (Ghosal and Roy, 2006; Ghosal et al., 1999; Tokdar and Ghosh, 2007; Freedman, 1963), the posterior is weakly consistent at w^* in the sense that for any neighborhood U of w^*, the probability that $\mu(w)$ lies in U converges to 1.

$$\mathbb{P}[U|\tilde{y}^1, \tilde{y}^2, \ldots, \tilde{y}^n] \to 1.$$

Without loss of generality, assume that the alternative $\boldsymbol{x}^*$ with the largest probability of +1 is unique, which means $\sigma((\boldsymbol{x}^*)^T \boldsymbol{w}^*) > \sigma(\boldsymbol{x}^T \boldsymbol{w}^*)$ for any alternative

$\boldsymbol{x}$ other than $\boldsymbol{x}^*$. Then we can pick U to be the neighborhood of $\boldsymbol{w}^*$ such that $\sigma((\boldsymbol{x}^*)^T\boldsymbol{w}) > \sigma(\boldsymbol{x}^T\boldsymbol{w})$ holds for any $\boldsymbol{w} \in U$. The neighborhood U exists because we only have finite number of altervatives. From the consistency results, the probability that the best arm under posterior estimation is the true best alternative goes to 1 as the measurement budget goes to infinity.

Bibliography

Agrawal, R. (1995). Sample mean based index policies with $o(logn)$ regret for the multi-armed bandit problem. *Advances in Applied Probability*, pages 1054–1078.

Agrawal, R., Hegde, M., and Teneketzis, D. (1990). Multi-armed bandit problems with multiple plays and switching cost. *Stochastics and Stochastic Reports*, 29(4):437–459.

Agrawal, S. and Goyal, N. (2012). Thompson sampling for contextual bandits with linear payoffs. *arXiv preprint arXiv:1209.3352*.

Almirall, D., Compton, S. N., Gunlicks-Stoessel, M., Duan, N., and Murphy, S. A. (2012). Designing a pilot sequential multiple assignment randomized trial for developing an adaptive treatment strategy. *Statistics in medicine*, 31(17):1887–1902.

Ashby, D. (2006). Bayesian statistics in medicine: a 25 year review. *Statistics in medicine*, 25(21):3589–3631.

Audibert, J., Bubeck, S., and Lugosi, G. (2013). Regret in online combinatorial optimization. *Math. Oper. Res.*

Audibert, J.-Y. and Bubeck, S. (2010). Best arm identification in multi-armed bandits. In *COLT-23th Conference on Learning Theory-2010*, pages 13–p.

Audibert, J.-Y., Munos, R., and Szepesvári, C. (2009). Exploration-exploitation tradeoff using variance estimates in multi-armed bandits. *Theor. Comput. Sci.*, 410(19).

Auer, P., Cesa-Bianchi, N., and Fischer, P. (2002). Finite-time analysis of the multiarmed bandit problem. *Mach. Learn.*, 47(2-3):235–256.

Babenko, B., Yang, M.-H., and Belongie, S. (2009). A family of online boosting algorithms. In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pages 1346–1353. IEEE.

Barber, D. and Bishop, C. M. (1998). Ensemble learning for multi-layer networks. *Advances in neural information processing systems*, pages 395–401.

Barut, E. and Powell, W. B. (2013). Optimal learning for sequential sampling with non-parametric beliefs. *Journal of Global Optimization*, pages 1–27.

Bather, J. (1985). On the allocation of treatments in sequential medical trials. *International Statistical Review/Revue Internationale de Statistique*, pages 1–13.

Berry, D. A. and Eick, S. G. (1995). Adaptive assignment versus balanced randomization in clinical trials: a decision analysis. *Statistics in medicine*, 14(3):231–246.

Berry, S. M., Carlin, B. P., Lee, J. J., and Muller, P. (2010). *Bayesian adaptive methods for clinical trials*. CRC press.

Beygelzimer, A., Kale, S., and Luo, H. (2015). Optimal and adaptive algorithms for online boosting. In *ICML*, pages 2323–2331.

Bishop, C. M. et al. (2006). *Pattern recognition and machine learning*, volume 4. springer New York.

Bodenheimer, T. (2005). High and rising health care costs. part 1: seeking an explanation. *Annals of internal medicine*, 142(10):847–854.

Boyen, X. and Koller, D. (1998). Tractable inference for complex stochastic processes. In *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, pages 33–42. Morgan Kaufmann Publishers Inc.

Branke, J., Chick, S. E., and Schmidt, C. (2007). Selecting a selection procedure. *Management Science*, 53(12):1916–1932.

Brinkley, J. (2014). A doubly robust estimator for the attributable benefit of a treatment regime. *Statistics in medicine*, 33(29):5057–5073.

Brochu, E., Cora, V. M., and De Freitas, N. (2010). A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *arXiv preprint arXiv:1012.2599*.

Bubeck, S. and Cesa-Bianchi, N. (2012). *Regret analysis of stochastic and nonstochastic multi-armed bandit problems*. Foundations and Trends in Machine Learning.

Bubeck, S., Cesa-Bianchi, N., and Lugosi, G. (2012). Bandits with heavy tail. *arXiv preprint arXiv:1209.1727*.

Buckwalter, J. A. and Lohmander, S. (1994). Operative treatment of osteoarthrosis. current practice and future development. *J Bone Joint Surg Am*, 76(9):1405–1418.

Callahan, C. M., Drake, B. G., Heck, D. A., and Dittus, R. S. (1994). Patient outcomes following tricompartmental total knee replacement: a meta-analysis. *Jama*, 271(17):1349–1357.

Cappé, O., Garivier, A., Maillard, O.-A., Munos, R., Stoltz, G., et al. (2013). Kullback–leibler upper confidence bounds for optimal sequential allocation. *Ann. Statist.*, 41(3):1516–1541.

Cesa-Bianchi, N. and Lugosi, G. (2012). Combinatorial bandits. *J. Comput. System Sci.*, 78(5):1404–1422.

Chapelle, O. and Li, L. (2011). An empirical evaluation of thompson sampling. In *Advances in neural information processing systems*, pages 2249–2257.

Chen, C.-H., Chen, H.-C., and Dai, L. (1996). A gradient approach for smartly allocating computing budget for discrete event simulation. In *Proceedings of the 28th conference on Winter simulation*, pages 398–405. IEEE Computer Society.

Chen, C.-H., He, D., and Fu, M. (2006). Efficient dynamic simulation allocation in ordinal optimization. *Automatic Control, IEEE Transactions on*, 51(12):2005–2009.

Chen, C.-H., Lin, J., Yücesan, E., and Chick, S. E. (2000). Simulation budget allocation for further enhancing the efficiency of ordinal optimization. *Discrete Event Dynamic Systems*, 10(3):251–270.

Chen, S.-T., Lin, H.-T., and Lu, C.-J. (2012). An online boosting algorithm with theoretical justifications. In *ICML*.

Chen, Y. and Krause, A. (2013). Near-optimal batch mode active learning and adaptive submodular optimization. In *Proceedings of The 30th International Conference on Machine Learning*, pages 160–168.

Chiang, B. N., Perlman, L. V., and Epstein, F. H. (1969). Overweight and hypertension a review. *Circulation*, 39(3):403–421.

Chick, S. E. (2001). New two-stage and sequential procedures for selecting the best simulated system. *Operations Research*, 49(5):732–743.

Chow, S.-C. (2014). Adaptive clinical trial design. *Annual review of medicine*, 65:405–415.

Chu, W., Li, L., Reyzin, L., and Schapire, R. E. (2011a). Contextual bandits with linear payoff functions. In *International Conference on Artificial Intelligence and Statistics*, pages 208–214.

Chu, W., Zinkevich, M., Li, L., Thomas, A., and Tseng, B. (2011b). Unbiased online active learning in data streams. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 195–203. ACM.

Cox, D. R. and Hinkley, D. V. (1979). *Theoretical statistics*. CRC Press.

DeGroot, M. H. (1970). *Optimal Statistical Decisions*. McGraw-Hill.

Dietterich, T. G. (2000). Ensemble methods in machine learning. In *International workshop on multiple classifier systems*, pages 1–15. Springer.

Djurii, A. B. and Leung, Y. (2006). Optical properties of zno nanostructures. *Small*, 2(8-9):944–961.

Dudík, M., Erhan, D., Langford, J., Li, L., et al. (2014). Doubly robust policy evaluation and optimization. *Statistical Science*, 29(4):485–511.

Eichelsdoerfer, D. J., Liao, X., Cabezas, M. D., Morris, W., Radha, B., Brown, K. A., Giam, L. R., Braunschweig, A. B., and Mirkin, C. A. (2013). Large-area molecular patterning with polymer pen lithography. *Nat. Protoc.*, 8(12):2548–2560.

Fahrmeir, L. and Kaufmann, H. (1985). Consistency and asymptotic normality of the maximum likelihood estimator in generalized linear models. *The Annals of Statistics*, pages 342–368.

Ferreira, A. J. and Figueiredo, M. A. (2012). Boosting algorithms: A review of methods, theory, and applications. In *Ensemble Machine Learning*, pages 35–85. Springer.

Filippi, S., Cappe, O., Garivier, A., and Szepesvári, C. (2010). Parametric bandits: The generalized linear case. In *Advances in Neural Information Processing Systems*, pages 586–594.

Flory, F.and Escoubas, L. and Berginc, G. (2011). Optical properties of nanostructured materials: a review. *J. of Nanophotonics*, 5(1):052502–052502.

Frazier, P., Powell, W., and Dayanik, S. (2009). The knowledge-gradient policy for correlated normal beliefs. *INFORMS journal on Computing*, 21(4):599–613.

Frazier, P. I. and Powell, W. B. (2010). Paradoxes in learning and the marginal value of information. *Decision Analysis*, 7(4):378–403.

Frazier, P. I. and Powell, W. B. (2011). Consistency of sequential bayesian sampling policies. *SIAM J. Control Optim.*, 49(2):712–731.

Frazier, P. I., Powell, W. B., and Dayanik, S. (2008). A knowledge-gradient policy for sequential information collection. *SIAM Journal on Control and Optimization*, 47(5):2410–2439.

Freedman, D. A. (1963). On the asymptotic behavior of bayes' estimates in the discrete case. *The Annals of Mathematical Statistics*, pages 1386–1403.

Freund, Y., Seung, H. S., Shamir, E., and Tishby, N. (1997). Selective sampling using the query by committee algorithm. *Machine learning*, 28(2-3):133–168.

Gehr, R. J. and Boyd, R. W. (1996). Optical properties of nanostructured optical materials. *Chemistry of Materials*, 8(8):1807–1819.

Gelman, A., Carlin, J. B., Stern, H. S., and Rubin, D. B. (2014). *Bayesian data analysis*, volume 2. Chapman & Hall/CRC Boca Raton, FL, USA.

Ghosal, S., Ghosh, J. K., Ramamoorthi, R., et al. (1999). Posterior consistency of dirichlet mixtures in density estimation. *Ann. Statist*, 27(1):143–158.

Ghosal, S. and Roy, A. (2006). Posterior consistency of gaussian process prior for nonparametric binary regression. *The Annals of Statistics*, pages 2413–2429.

Ginsburg, P. B. (2008). *High and rising health care costs: Demystifying US health care spending*. Robert Wood Johnson Foundation.

Gittins, J. C. (1979). Bandit processes and dynamic allocation indices. *Journal of the Royal Statistical Society. Series B (Methodoogical)*.

Goldengorin, B., Sierksma, G., Tijssen, G. A., and Tso, M. (1999). The data-correcting algorithm for the minimization of supermodular functions. *Management Science*, 45(11):1539–1551.

Golovin, D. and Krause, A. (2010). Adaptive submodularity: A new approach to active learning and stochastic optimization. In *COLT*, pages 333–345.

Golub, G. H. and Van Loan, C. F. (2012). *Matrix computations*, volume 3. JHU Press.

Gopalan, A., Mannor, S., and Mansour, Y. (2014). Thompson sampling for complex online problems. In *Proceedings of The 31st International Conference on Machine Learning*, pages 100–108.

Gourieroux, C. and Monfort, A. (1981). Asymptotic properties of the maximum likelihood estimator in dichotomous logit models. *Journal of Econometrics*, 17(1):83–97.

Graepel, T., Candela, J. Q., Borchert, T., and Herbrich, R. (2010). Web-scale bayesian click-through rate prediction for sponsored search advertising in microsoft's bing search engine. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 13–20.

Guo, Y. and Schuurmans, D. (2008). Discriminative batch mode active learning. In *Advances in neural information processing systems*, pages 593–600.

Gupta, S. S. and Miescke, K. J. (1996). Bayesian look ahead one-stage sampling allocations for selection of the best population. *Journal of statistical planning and inference*, 54(2):229–244.

Gutmann, H.-M. (2001). A radial basis function method for global optimization. *Journal of Global Optimization*, 19(3):201–227.

Guttman, I., Tiao, G. C., et al. (1964). A bayesian approach to some best population problems. *Ann. Math. Statist.*, 35(2):825–835.

Haberman, S. J. and Haberman, S. J. (1974). *The analysis of frequency data*, volume 194. University of Chicago Press Chicago.

He, D., Chick, S. E., and Chen, C.-H. (2007). Opportunity cost and OCBA selection procedures in ordinal optimization for a fixed number of alternative systems. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 37(5):951–961.

He, J. and Thiesson, B. (2007). Asymmetric gradient boosting with application to spam filtering. In *CEAS*.

Hennig, P. and Schuler, C. J. (2012). Entropy search for information-efficient global optimization. *The Journal of Machine Learning Research*, 13(1):1809–1837.

HHS (2015). https://innovation.cms.gov/initiatives/ccjr/index.html.

Hoffman, M. D., Shahriari, B., and de Freitas, N. (2014). On correlation and budget constraints in model-based bandit optimization with application to automatic machine learning. In *AISTATS*, pages 365–374.

Hosmer Jr, D. W. and Lemeshow, S. (2004). *Applied logistic regression*. John Wiley & Sons.

Howard, R. (1966). Information value theory. *IEEE Trans Syst. Sci. and Cybern.*, 2(1):22–26.

Hu, F. and Rosenberger, W. F. (2006). *The theory of response-adaptive randomization in clinical trials*, volume 525. John Wiley & Sons.

Huang, D., Allen, T. T., Notz, W. I., and Zeng, N. (2006). Global optimization of stochastic black-box systems via sequential kriging meta-models. *Journal of global optimization*, 34(3):441–466.

Huo, F., Zheng, Z., Zheng, G., Giam, L. R., Zhang, H., and Mirkin, C. A. (2008). Polymer pen lithography. *Science*, 321(5896):1658–1660.

Jack Lee, J. and Chu, C. T. (2012). Bayesian clinical trials in action. *Statistics in medicine*, 31(25):2955–2972.

Jones, D. R. (2001). A taxonomy of global optimization methods based on response surfaces. *Journal of global optimization*, 21(4):345–383.

Jones, D. R., Schonlau, M., and Welch, W. J. (1998). Efficient global optimization of expensive black-box functions. *Journal of Global optimization*, 13(4):455–492.

Kaelbling, L. P. (1993). *Learning in embedded systems*.

Kale, S., Reyzin, L., and Schapire, R. E. (2010). Non-stochastic bandit slate problems. In *NIPS*, pages 1054–1062.

Kapoor, A. and Greiner, R. (2005). *Learning and classifying under hard budgets*. Springer.

Kleinberg, R., Niculescu-Mizil, A., and Sharma, Y. (2010). Regret bounds for sleeping experts and bandits. *Machine learning*, 80(2-3):245–272.

Krause, A. and Ong, C. S. (2011). Contextual gaussian process bandit optimization. In *Advances in Neural Information Processing Systems*, pages 2447–2455.

Kuleshov, V. and Precup, D. (2000). Algorithms for multi-armed bandit problems. *Journal of Machine Learning Research*.

Kushner, H. J. (1964). A new method of locating the maximum point of an arbitrary multipeak curve in the presence of noise. *Journal of Basic Engineering*, 86(1):97–106.

Laber, E. B., Zhao, Y.-Q., Regh, T., Davidian, M., Tsiatis, A., Stanford, J. B., Zeng, D., Song, R., and Kosorok, M. R. (2015). Using pilot data to size a two-arm randomized trial to find a nearly optimal personalized treatment strategy. *Statistics in medicine*.

Lai, T. L. (1987). Adaptive treatment allocation and the multi-armed bandit problem. *The Annals of Statistics*, pages 1091–1114.

Lai, T. L. and Liao, O. Y.-W. (2012). Efficient adaptive randomization and stopping rules in multi-arm clinical trials for testing a new treatment. *Sequential analysis*, 31(4):441–457.

Lai, T. L. and Robbins, H. (1985). Asymptotically efficient adaptive allocation rules. *Advances in applied mathematics*, 6(1):4–22.

Langford, J. and Zhang, T. (2008). The epoch-greedy algorithm for contextual multi-armed bandits. In *Advances in Neural Information Processing Systems*.

Langille, M. R., Personick, M. L., Zhang, J., and Mirkin, C. A. (2012). Defining rules for the shape evolution of gold nanoparticles. *J. Am. Chem. Soc.*, 134(35):14542–14554.

Lauritzen, S. L. (1992). Propagation of probabilities, means, and variances in mixed graphical association models. *Journal of the American Statistical Association*, 87(420):1098–1108.

Lavori, P. W. and Dawson, R. (2000). A design for testing clinical strategies: biased adaptive within-subject randomization. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 163(1):29–38.

Le Cessie, S. and Van Houwelingen, J. C. (1992). Ridge estimators in logistic regression. *Applied statistics*, pages 191–201.

Lehnert, T., Heider, D., Leicht, H., Heinrich, S., Corrieri, S., Luppa, M., Riedel-Heller, S., and König, H.-H. (2011). Review: health care utilization and costs of elderly persons with multiple chronic conditions. *Medical Care Research and Review*, 68(4):387–420.

Li, L., Chu, W., Langford, J., and Schapire, R. E. (2010). A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 661–670. ACM.

Li, L., Chu, W., Langford, J., and Wang, X. (2011). Unbiased offline evaluation of contextual-bandit-based news article recommendation algorithms. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 297–306. ACM.

Liao, X., Brown, K. A., Schmucker, A. L., Liu, G., He, S., Shim, W., and Mirkin, C. A. (2013). Desktop nanofabrication with massively multiplexed beam pen lithography. *Nat. Commun.*, 4.

Lichman, M. (2013). UCI machine learning repository.

Lin, H.-T., EDU, N., Lu, C.-J., and EDU, S. (2014). Boosting with online binary learners for the multiclass bandit problem. In *ICML*.

Mahajan, D. K., Rastogi, R., Tiwari, C., and Mitra, A. (2012). Logucb: an explore-exploit algorithm for comments recommendation. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 6–15. ACM.

Martinez-Cantin, R. (2014). Bayesopt: a bayesian optimization library for nonlinear optimization, experimental design and bandits. *The Journal of Machine Learning Research*, 15(1):3735–3739.

Maybeck, P. S. (1982). *Stochastic models, estimation, and control*, volume 3. Academic press.

Mes, M. R., Powell, W. B., and Frazier, P. I. (2011). Hierarchical knowledge gradient for sequential sampling. *The Journal of Machine Learning Research*, 12:2931–2974.

Millstone, J. E., Hurst, S. J., Mtraux, G. S., Cutler, J. I., and Mirkin, C. A. (2009). Colloidal gold and silver triangular nanoprisms. *Small*, 5(6):646–664.

Mockus, J. (1994). Application of bayesian approach to numerical methods of global and stochastic optimization. *Journal of Global Optimization*, 4(4):347–365.

Montgomery, D. C. (2008). *Design and Analysis of Experiments*. John Wiley and Sons.

Murphy, S. A. (2003). Optimal dynamic treatment regimes. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 65(2):331–355.

Murphy, S. A. (2005). An experimental design for the development of adaptive treatment strategies. *Statistics in medicine*, 24(10):1455–1481.

Murphy, S. A. and Collins, L. M. (2007). Customizing treatment to the patient: Adaptive treatment strategies. *Drug and alcohol dependence*, 88(Suppl 2):S1.

Murphy, S. A., Van Der Laan, M., and Robins, J. M. (2001). Marginal mean models for dynamic regimes. *Journal of the American Statistical Association*, 96(456):1410–1423.

Negoescu, D. M., Frazier, P. I., and Powell, W. B. (2011). The knowledge-gradient algorithm for sequencing experiments in drug discovery. *INFORMS Journal on Computing*, 23(3):346–363.

Nemhauser, G. L., Wolsey, L. A., and Fisher, M. L. (1978). An analysis of approximations for maximizing submodular set functions. *Mathematical Programming*, 14(1):265–294.

Newman, M. E. (2006). Modularity and community structure in networks. *Proceedings of the national academy of sciences*, 103(23):8577–8582.

Nickisch, H. and Rasmussen, C. E. (2008). Approximations for binary gaussian process classification. *Journal of Machine Learning Research*, 9(Oct):2035–2078.

O'Hagan, A. (1978). On curve fitting and optimal design for regression. *J. Royal Stat. Soc. B*, 40:1–32.

Osborne, M. A., Garnett, R., and Roberts, S. J. (2009). Gaussian processes for global optimization. In *3rd international conference on learning and intelligent optimization (LION3)*, pages 1–15. Citeseer.

Oza, N. C. (2005). Online bagging and boosting. In *Systems, man and cybernetics, 2005 IEEE international conference on*, volume 3, pages 2340–2345. IEEE.

Park, S. Y., Lytton-Jean, A. K., Lee, B., Weigand, S., Schatz, G. C., and Mirkin, C. A. (2008). DNA-programmable nanoparticle crystallization. *Nat.*, 451(7178):553–556.

Picheny, V., Wagner, T., and Ginsbourger, D. (2013). A benchmark of kriging-based infill criteria for noisy optimization. *Structural and Multidisciplinary Optimization*, 48(3):607–626.

Powell, W. B. (2016). A unified framework for optimization under uncertainty. In *Optimization Challenges in Complex, Networked and Risky Systems*, pages 45–83. INFORMS.

Powell, W. B. and Ryzhov, I. O. (2012). *Optimal learning*. John Wiley & Sons.

Qian, M. and Murphy, S. A. (2011). Performance guarantees for individualized treatment rules. *Annals of statistics*, 39(2):1180.

Radlinski, F., Kleinberg, R., and Joachims, T. (2008). Learning diverse rankings with multi-armed bandits. In *Proceedings of the 25th international conference on Machine learning*, pages 784–791. ACM.

Raiffa, H. and Schlaifer, R. (1961). Applied statistical decision theory. *Harvard Business School Publications*.

Regis, R. G. and Shoemaker, C. A. (2005). Constrained global optimization of expensive black box functions using radial basis functions. *Journal of Global Optimization*, 31(1):153–171.

Robins, J. M. (1993). Information recovery and bias adjustment in proportional hazards regression analysis of randomized trials using surrogate markers. In *Proceedings of the Biopharmaceutical Section, American Statistical Association*, volume 24, page 3. American Statistical Association.

Robins, J. M. (2004). Optimal structural nested models for optimal sequential decisions. In *Proceedings of the second seattle Symposium in Biostatistics*, pages 189–326. Springer.

Ryzhov, I. O. and Powell, W. (2009a). A Monte Carlo knowledge gradient method for learning abatement potential of emissions reduction technologies. *Proceedings of the Winter Simulation Conference*, pages 1492–1502.

Ryzhov, I. O. and Powell, W. B. (2009b). The knowledge gradient algorithm for online subset selection. In *IEEE SSCI ADPRL, Nashville, TN*, pages 137–144.

Ryzhov, I. O., Powell, W. B., and Frazier, P. I. (2012). The knowledge gradient algorithm for a general class of online learning problems. *Operations Research*, 60(1):180–195.

Sahami, M., Dumais, S., Heckerman, D., and Horvitz, E. (1998). A bayesian approach to filtering junk e-mail. In *Learning for Text Categorization: Papers from the 1998 workshop*, volume 62, pages 98–105.

Schapire, R. E. and Freund, Y. (2012). *Boosting: Foundations and algorithms*. MIT press.

Schein, A. I. and Ungar, L. H. (2007). Active learning for logistic regression: an evaluation. *Machine Learning*, 68(3):235–265.

Senesi, A. J., Eichelsdoerfer, D. J., Macfarlane, R. J., Jones, M. R., Auyeung, E., Lee, B., and Mirkin, C. A. (2013). Stepwise evolution of DNA-programmable nanoparticle superlattices. *Angew. Chem. Int. Ed.*, 52(26):6624–6628.

Settles, B. (2010). Active learning literature survey. *University of Wisconsin, Madison*, 52(55-66):11.

Snoek, J., Larochelle, H., and Adams, R. P. (2012). Practical bayesian optimization of machine learning algorithms. In *Advances in neural information processing systems*, pages 2951–2959.

Spiegelhalter, D. J. and Lauritzen, S. L. (1990). Sequential updating of conditional probabilities on directed graphical structures. *Networks*, 20(5):579–605.

Srinivas, N., Krause, A., Kakade, S. M., and Seeger, M. (2009). Gaussian process optimization in the bandit setting: No regret and experimental design. *arXiv preprint arXiv:0912.3995*.

Stigler, G. J. (1961). The economics of information. *J. political Econ.*, pages 213–225.

Tesch, M., Schneider, J., and Choset, H. (2013). Expensive function optimization with stochastic binary outcomes. In *Proceedings of The 30th International Conference on Machine Learning*, pages 1283–1291.

Thall, P. F. and Simon, R. (1994). Practical bayesian guidelines for phase iib clinical trials. *Biometrics*, pages 337–349.

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

Tokdar, S. T. and Ghosh, J. K. (2007). Posterior consistency of logistic gaussian process priors in density estimation. *Journal of Statistical Planning and Inference*, 137(1):34–42.

Tong, S. and Koller, D. (2002). Support vector machine active learning with applications to text classification. *The Journal of Machine Learning Research*, 2:45–66.

Uchiya, T., Nakamura, A., and Kudo, M. (2010). Algorithms for adversarial bandit problems with multiple plays. In *Algorithmic Learning Theory*, pages 375–389. Springer.

Wang, Y. and Powell, W. (2016a). Finite-time analysis for the knowledge-gradient policy. *arXiv preprint arXiv:1606.04624*.

Wang, Y. and Powell, W. (2016b). Molte: a modular optimal learning testing environment.

Wang, Y. and Powell, W. (2016c). An optimal learning method for developing personalized treatment regimes. *arXiv preprint arXiv:1607.01462*.

Wang, Y., Reyes, K. G., Brown, K. A., Mirkin, C. A., and Powell, W. B. (2015). Nested-batch-mode learning and stochastic optimization with an application to sequential multistage testing in materials science. *SIAM Journal on Scientific Computing*, 37(3):B361–B381.

Wang, Y., Wang, C., and Powell, W. (2016). The knowledge gradient for sequential decision making with stochastic binary feedbacks. In *Proceedings of the 33rd International Conference on Machine Learning*, pages 1138–1147.

Wennberg, J. E., Fisher, E. S., Goodman, D. C., and Skinner, J. S. (2008). Tracking the care of patients with severe chronic illness-the dartmouth atlas of health care 2008.

Wetherill, G. B. and Glazebrook, K. D. (1986). *Sequential Methods in Statistics*. Chapman and Hall.

Wright, S. J. and Nocedal, J. (1999). *Numerical optimization*, volume 2. Springer New York.

Yin, G., Chen, N., and Jack Lee, J. (2012). Phase ii trial design with bayesian adaptive randomization and predictive probability. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 61(2):219–235.

Yokota, F. and Thompson, K. M. (2004). Value of information literature analysis: a review of applications in health risk management. *Med. Decis. Mak.*, 24(3):287–298.

Zhang, B., Tsiatis, A. A., Laber, E. B., and Davidian, M. (2012). A robust method for estimating optimal treatment regimes. *Biometrics*, 68(4):1010–1018.

Zhang, T. and Oles, F. (2000). The value of unlabeled data for classification problems. In *Proceedings of the Seventeenth International Conference on Machine Learning,(Langley, P., ed.)*, pages 1191–1198. Citeseer.